Enhancing Cloud Forensic Investigation System in Distributed Cloud Computing Using DK-CP-ECC Algorithm and EK-ANFIS

Shaiqa Nasreen[1,2,*] and Ajaz Hussain Mir[1]

[1]*Department of Electronics and Communication Engineering, National Institute of Technology, Srinagar, J & K, India*
[2]*Islamic University of Science and Technology, Awantipora, J&K, India*
E-mail: shakks10@outlook.com; ahmir@rediffmail.com; ahmir@nitsri.net
[*]*Corresponding Author*

Received 29 March 2022; Accepted 25 August 2022;
Publication 15 February 2023

Abstract

The investigation as well as recovery of data gathered as of digital devices associated with computer crime is involved in Digital Forensics (DF). In a distributed Cloud Server (CS), DF investigation is more complicated (during collecting, preserving, and reporting the evidence) as well as insecure during gathering evidence as of the cloud sources. Centered on the DF investigation system, numerous works were performed. However, lots of challenges still remain that bring about cybercrime. The work developed a robust cloud forensic investigation system centered upon distributed Cloud Computing (CC) for conquering the challenges. It is framed into the '3' phase (i.e.) originally, Group Key Generation (GKG) phase that enables the authorized user to upload or download the evidence for maintaining the evidence's trustworthiness. Distributed Key Cipher Policy Elliptic Curve Cryptography (DK-CP-ECC) algorithm performed the Secure Data Transfer (SDT) phase. It aids in maintaining the evidence's privacy together with confidentiality. Exponential Membership Function Adaptive Neuro-Fuzzy Interference System (EK-ANFIS) carries out the CS selection with the aid of a deer hunting

Journal of Mobile Multimedia, Vol. 19_3, 679–706.
doi: 10.13052/jmm1550-4646.1933

genetic algorithm that evades the reporting issues and renders secure evidence storage. 97% Security Level (SL) is obtained by the proposed work that is better analogized to the prevailing frameworks.

Keywords: Digital forensic investigation, group key generation, L-caesar cipher, elliptic curve cryptography, feature extraction, deer hunting genetic algorithm, and exponential membership function adaptive neuro-fuzzy interference system (EK-ANFIS).

1 Introduction

In current days, to detect real-time data as of the surroundings, loads of intelligent objects and devices are now equipped with sensors. But with the development of distributed computing, this concept has been shifted because all the intelligent objects are linked via a network of networks and endowed with data analytic capacities. At a swift speed, these distributed devices generate a large quantity of data streams [1]. Consequently, there is a greater chance of cloud-based attacks. The competence to check the cloud environments is made possible by distributed cloud-based Forensic investigation (FI). In an epoch wherein the linked devices are expected to be approximately 30 billion, DF was diffused to cover an extensive gamut of technologies. DFs are a reputable skill field in the cyber security sector and an important as well as an unavoidable aspect of an Incident Response Plan addressing electronic data. DFs are mostly used to conduct technical analysis within the limits of the legal system in retaliation to criminal actions that engage electronic devices [2]. Amid the data integrity protection, a FI includes gathering, auditing, and analyzing the data [3].

But, forensic analysis has become harder as more people move to greatly distributed cloud architecture. Managing forensic Big Data in relation to hardware as well as software logs that are produced and kept on a cloud platform is one amongst the primary issues [4]. Owing to the assorted character of the distributed infrastructure, the present DF methods, tools, and communication standards can't be handled. The risk of crime was raised with the aggressive expansion of the distributed device [5]; this could result in unintended consequences for lots of cyber-criminals. As the sensor-sensitive data passes across different networks, user devices, and communication passageways, the data might be weak or unprotected in intricate physical devices [6]. To strengthen the creditability of the entire stakeholder's communication data, there is no method in cloud forensics. In cloud forensics, there

are yet no ways for strengthening the creditability of the entire stakeholders' communication data. Society has turned out to be increasingly digitalized, devices have turned into more generalized, and more effectual electronic devices were developed as the Internet's penetration rate has augmented and devices have turned ever-smaller [7].

The Anti-forensic System (AFS) can well be utilized by the users [8, 9] together with the tools that are employed to modify, eliminate, interfere, or disturb the proof of the illegal activities on the devices. The AFS is primarily used for evidence obfuscation [10, 11], concealment, or damage. Hackers or harmful users use it to conceal their actions, and the sensitive secret organizations would like nothing about them to be traced back to [12]. The persistent solution for the sweltering challenge that could influence the turmoil of the fast-emerging digital storage technologies is safety, conservation of confidentiality, and cost-effective cloud depository [13]. A public auditing protocol is a serious solution that assists the critical roles, say protection of privacy [14, 15], validation of data [16], auditing, verification [17], reliability and auditing of vibrant data [18], along with cloud storage's privacy [19]. Hence, it ought to be adaptable, efficient, and produce the advantages of asymmetric verification [20]. It includes an expanded safety, dispensable updates by the host with a covert in the field, short key along with less calculation as a proof to create association even with a new individual who should be reliable for sharing the distributed cloud's data.

1.1 Problem Definition

Though lots of techniques were introduced recently to manage and protect data on the distributed cloud, these have not rendered any desirable security due to the following drawbacks enlisted as,

- Many techniques were posited for securely amassing the data in the cloud; however, during massive transmission, none of them maintain end-to-end security.
- Data security issues related to data storage, integrity, access, and breaches are identified in cloud computing.
- Though some level of control is given over the virtual machines, a loss of control exists for the data storage.
- Various user authentication schemes developed to protect a user's personal information in a cloud environment has revealed that the level of security is still insufficient to authenticate and manage users in the current cloud computing environment.

- As the data owners and data are at different locations and platforms, it is challenging to manage unauthorized access to services and stored data and identity controls in cloud computing.

This paper has created a new structure for enriching a cloud FI in distributed CS to render a result to manage the prevailing challenge. The key contribution and findings of the presented approach are,

- In order to maintain data confidentiality and to restrict access to the information, user authentication is performed with the group key generation process.
- For ensuring the secure uploading and downloading of data secure data transfer has been performed using DK-CP-ECC techniques.
- To assure correct and safe distribution of data, the selection of cloud server is performed by feature extraction and feature selection and for accurate classification of CS, the EK-ANFIS technique is introduced.

Thus, this paper is arranged as, several methods created for cloud forensic devices is reviewed in Section 2, on the base of the cloud FI system, the proposed work is exhibited in Section 3, the outcome obtained by the proposed work with the aid of diverse performance metrics is exhibited in Section 4, at last, this paper is concluded in Section 5 along with its future scope.

2 Literature Survey

Mehran Pourvahab et al. [21] illustrated to safeguard the device from illegal users by recommending a technique called Secure Ring Verification based Authentication (SRVA). Secret keys were created through the Harmony Search Optimization (HSO) technique to reinforce the cloud environment. Relying on the sensitivity level, entire data were encrypted and accumulated in the CS. Sensitivity Aware Deep Elliptic Curve Cryptography was used for encryption. A block was established on the SDN controller for each piece of cloud data, and then the data's history was logged as metadata. Secure Hashing Algorithm-3 was utilized to create the Merkle hash tree in every block. At last, the construction of a Logical Graph of Evidences as of the blockchain facilitated evidence analysis. SDN's network forensics was not determined.

Yuan Zhang et al. [22] published a data provenance scheme, that is efficient and safe, and it was executed in the ESP system. Using a block chain-based provenance record chain, ESP was distinguished. It could offer safety along with effective data outsourcing whilst assuring the accuracy, reliability,

and promptness of provenance records. Additionally, to appraise the viability of a secure provenance plan, the Window of Latching (WoL) theory had been established. They examined ESP's safety and assessed its performance through implementation, establishing that ESP's WoL was short and that it was safe and realistic. Through implementation, they examined ESP's safety and assessed its performance, establishing that ESP's WoL was short and it was safe as well as realistic.

Alenezi et al. [23] examined the factors that aid organizations' forensic preparation by presenting a system. After a detailed assessment of prior studies in the literature and an appraisal of the appropriate industry principle, this system was discovered. The factors were grouped and combined to construct the structure by thoroughly studying the components, gathering them as of the literature, investigating them, and eliminating the duplicates. The study strategy included two steps: reviewing of literature after that expert assessment to acquire credible data. To construct a broad image of the investigation concept together with the authentication and confirmation of the findings these methods were utilized.

Xuanyu Liu et al. [24] advocated Decentralization along with Multi-participation in Distributed Cloud Forensics systems (DCFS). With the assumption that users and staff of the cloud or forensic inspectors might be deceitful, the DCFS was established in a disloyal along with multi-tenancy cloud environment. The DCFS did not depend on a single node or any '3rd' party to achieve reliable proof as of the cloud, because it was a distributed and decentralized structure. All DCFS participants were given equal amounts of trust, and they were assigned with supervising each other. The DCFS kept a distributed public ledger, which recorded all the forensic evidentiary proofs together with other pertinent information. To a certain extent, the forensic evidence's reliability in addition to the integrity was improved by the ledger, and it also finished the FI's chain of supervision. The cloud workers gave the forensic evidence to the court of law with the aid of more trustful DCFS. The DCFS still has trust issues in the cloud, and it is one of the constraints.

Reviewing the strengths and weaknesses of the different cloud forensic investigation systems and some of the similarities, differences, and gaps are identified that can be researched further. The digital forensic architecture developed in [21] collects and preserves reliable evidence from the IaaS cloud environment. However, SDN's network forensics was not determined. The data provenance scheme [22] is secure against provenance record forgery, removal, and modification attacks. The DCFS [23] makes the cloud more

accountable and robust. But, the methods have key management and trust issues in the cloud. Although, ProvChain architecture [24] embedded provides security features, the protection for the cloud storage application was not guaranteed.

So far, there were sufficient articles about cloud-based FI systems. However, some difficulties like security cause problems. This investigation paper exhibits an improved cloud FI in a distributed cloud environment, utilizing new methodologies to deal with the above-said issues.

3 Proposed Cloud Forensic Investigation System for Distributed Cloud Server

An infrastructure is possessed by a network of elements or servers called the cloud for providing processing as well as storage facilities to clients all over the globe over the internet. Nevertheless, an added risk is present due to the frauds and cybercrimes that happened to steal vital data. Cloud FI system is required for analyzing the processes, procedures, methodologies, tools as well as techniques together with its utilization in CC for tracking digital evidence, fraud, in addition to cybercrime. This technology is evolving as well as spreading all through the globe and utilizing the capability of the internet for greater achievements. However, a challenge of safe evidence storage still remains. The work has proposed a distributed cloud FI system for coping with the prevailing challenge, which is exhibited in Figure 1. The proposed distributed cloud FI System shown in Figure 1 efficiently handles the challenge of safe evidence storage with three significant steps such as,

- GKG
- SDT
- Selection of distributed CS

The group key generation scheme distinguishes malicious entities from legitimate entities by maintaining data confidentiality. Next, the SDT phase ensures the secure storage of data with the steps called attribute extraction, cipher policy creation, and encryption using the DK-CP-ECC algorithm. Finally, the Feature Extraction (FE) along with the DHGA is done for choosing the CS for securely storing the split files in a distributed mode on the cloud.

In the forthcoming sections, the proposed work is briefly discussed.

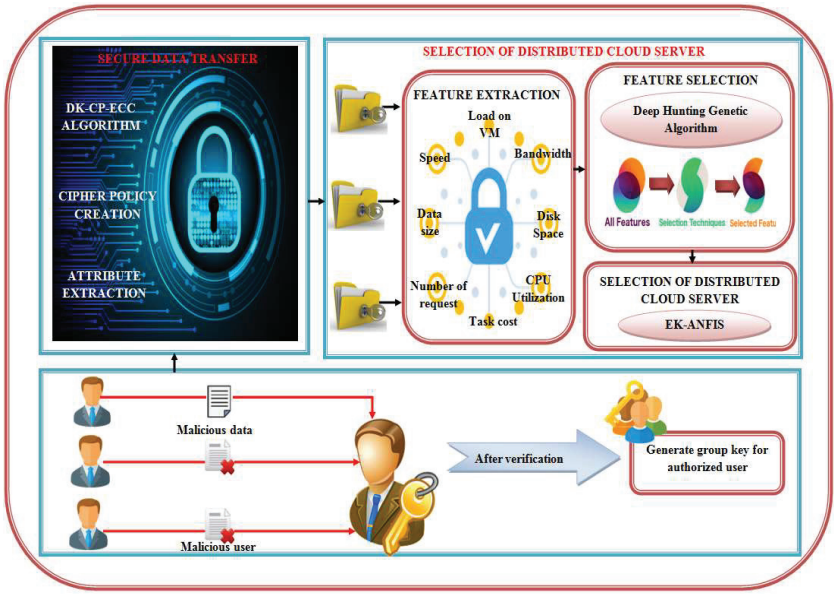


Figure 1 Proposed distributed cloud FI system.

3.1 Group Key Generation

For the user authentication as of the distributed CS, the work is initiated with GKG. Primarily, every user registered to the system to upload or download data to or from the cloud. The user types the name, phone number, birth date, email, etc in the registration time. The group key is generated via the authentication server for the specific registering person (U_K). For the respective user, GKG is rendered by:

$$U_K = \gamma_{GK}[U_1, U_2, U_3, \ldots U_n] \tag{1}$$

Wherein, γ_{GK} signifies the GKG that is a mix of alpha numeric value.

For ameliorating the key security, Group Key (GK) has been split and distributed to CS, key generation server, together with the user.

$$\gamma_{GK} = ([\gamma_{GK1}][\gamma_{GK2}][\gamma_{GK3}] \ldots [\gamma_{GKn}]) \tag{2}$$

The GK is split grounded on the total user in a group. Then, the GK is shared betwixt groups of users and its combination, which is mathematically expressed as,

$$\gamma_{GK} = S_{K_n} + C_S + U_{K_n} \tag{3}$$

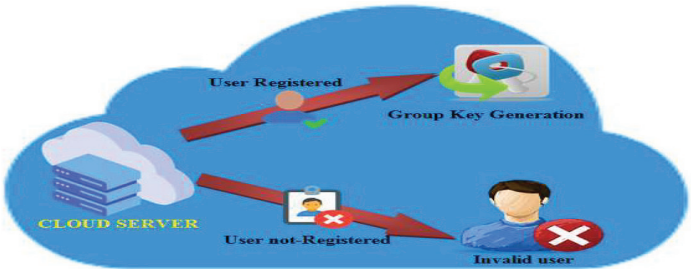


Figure 2 User authentication.

Wherein, S_{K_n} signifies sender's key, C_S cloud server U_{K_n} receiver's key.

User information is saved by the CSP. When users want to upload or download the data, they have to log in utilizing their user name along with password. The CS tests if a user has already been on the CSP list or not. The user will be called a non-authorized user if that individual doesn't exist on the registry, and the particular information that person is uploading is considered malicious data. Nevertheless, if the user is in the CSP list, then the generated GK is given to that user by the CS during the authentication server phase, which is depicted in Figure 2.

Subsequent to verification, the user, utilizing an SDT algorithm, can securely upload (or) download the data.

3.2 Secure Data Transfer

With the aid of encryption algorithms, the data is stored securely in cloud storage by the SDT. This phase encompasses '2' vital steps:

- Attribute Extraction (AE)
- Encryption of data using DK-CP-ECC Algorithm

3.2.1 Attribute extraction

Filename, file size, file type, etc, are the attributes of forensic data that will be extracted here. The AE helps in the conservation of the data more confidentially. The AE is rendered by:

$$\wp_{AE} = \{\wp_{AE}(\varsigma_{FILE})\} \tag{4}$$

$$\wp_{AE} = [\wp_{A1}, \wp_{A2}, \wp_{A3} \ldots \wp_{An}] \tag{5}$$

Wherein, ς_{FILE} signifies the file that to be extracted and equation '5' implies the disparate attributes of the file.

3.2.2 Cipher policy creation

Here, the L-Caesar Cipher algorithm converted the extracted attributes into cipher policy. Utilizing modular arithmetic (i.e.) by initially transforming the letters into numbers, the L-Caesar Cipher algorithm encryption can well be represented. Mathematically, encryption of a letter can well be illustrated as,

$$\mathrm{E}_{CP}(\wp_{Ai}) = (\wp_{Ai} + n)\mathrm{mod}\ 26 \tag{6}$$

$$D_{CP}(\wp_{Ai}) = (\wp_{Ai} - n)\mathrm{mod}\ 26 \tag{7}$$

$$n = len(\wp_{Ai}), \quad i = 1, 2, 3 \ldots n \tag{8}$$

Wherein E_{CP} signifies the encryption of attributes $\wp_{Ai}$ implies the input attributes n signifies the shifting value that relies upon the length of every attribute, D_{CP} signifies the decryption of attribute. The data is secured by means of the L-Caesar Cipher algorithm encryption via changing the shifting value centered on the attribute length. Distributed Key Elliptic Curves Cryptography (ECC) carries out the encryption of files subsequent to the attributes extraction.

3.2.3 Encryption of data using DK-CP-ECC algorithm

For evading any cybercrime, the files (ς_{FILE}) are encrypted. As the prevailing encryption algorithm wasn't that much confidential in securing the data, the data was opposed to anti-forensic attacks. A DK-CP-ECC is formed to conquer the challenge. The Distributed Key ECC is fundamentally a sort of mechanism utilized in public-key cryptography execution. This technique focuses on a curve with some basis points using a prime number function. The maximum limit is employed for this function. The Distributed Key ECC algorithm is mathematically expressed as:

$$\Phi^2 = \Omega^3 + a\Omega + b \tag{9}$$

(a) Key generation

Key generation is an imperative factor. The public together with the private key must be generated. With the aid of the receiver's public key, the message will be encrypted by the sender; in addition, its private key will be decrypted by the recipient. Initially, a number $\Re$ is randomly selected within a range n.

The public key is created utilizing the equation below,

$$\lambda_{PUK} = \lambda_{PRI} \times \mathrm{P} \tag{10}$$

Wherein, $\Re$ signifies the arbitrary number ranging betwixt (1 to $n-1$), P implies the point on the curve, λ_{PUK} implies the public key and λ_{PRI} signifies the private key.

(b) Encryption

Via selecting Γ as of [1 to $n-1$] arbitrarily, the encryption of the ς_{FILE} is performed. Under '2' ciphertexts that are $\hbar_1$ and $\hbar_2$, the encryption are done.

$$\hbar_1 = \Gamma \times \mathrm{P} \tag{11}$$

$$\hbar_2 = \varsigma_{FILE} + \Gamma \times \lambda_{PUK} \times \gamma_{GK} \times \mathrm{E}_{CP} \tag{12}$$

Then, $\hbar_1$ and $\hbar_2$ will be sent for decryption.

(c) Decryption

The message that was sent should be decrypted.

$$\varsigma_{FILE} = \hbar_2 - \frac{\lambda_{PRI} \times \lambda_{PUK}}{\gamma_{GK} \times \mathrm{E}_{CP}} \tag{13}$$

Thus, ς_{FILE} is the actual message decrypted utilizing the distributed key. Therefore, the SDT pseudo-code is depicted in Figure 3,

Input: Cloud User and Files
Output: Registered users in CSP and upload file

Begin
  Initialize the users $U_k = [U_1, U_2, U_3, \ldots\ldots U_n]$, files $F_k = [F_1, F_2, F_3, \ldots\ldots F_n]$
  For each user from k=1 to n
    Generate group key γ_{GK}
    Split the group key among the cloud server, key generation server and user,
      $\gamma_{GK} = ([\gamma_{GK1}][\gamma_{GK2}][\gamma_{GK3}]\ldots\ldots[\gamma_{GKn}])$
    Share the key among the users $\gamma_{GK} = S_{K_*} + C_S + U_{K_*}$
  End for
  If $(U_k == CS)$
    Upload the file ζ_{file}
    Extract attributes from the file, $\wp_{AE} = \{\wp_{AE}(\varsigma_{FILE})\}$
    Create cipher policy for extracted attribute using,
      $\mathrm{E}_{CP}(\wp_{Ai}) = (\wp_{Ai} + n) \bmod 26$
    Encrypt the file using,
      $\hbar_1 = \Gamma \times \mathrm{P}$
      $\hbar_2 = \varsigma_{FILE} + \Gamma \times \lambda_{PUK} \times \gamma_{GK} \times \mathrm{E}_{CP}$
  Else
    Invalid user
  End if
End begin

Figure 3 Pseudocode for SDT.

3.2.4 Selection of distributed cloud server

The main system element that guarantees correct as well as safe distribution of data is the Selection of Distributed CS. The phase comprises of subsequent steps:

- Feature extraction
- Feature Selection (FS)
- Selection of Distributed CS

3.2.4.1 *Feature extraction*

Features like Speed, Task Cost, data size, etc are extracted grounded on the encrypted data. To analyze the features characteristics and to evade computational intricacy, the features like processing speed as well as bandwidth number of requests, disk space, as well as CPU utilization is extracted as of the CS.

(a) Task cost

The amount that must be paid before the request can well be accomplished is classified as the Task cost. The demand cost is determined using the equation below:

$$\varsigma_1 = \chi * \delta_d \tag{14}$$

Wherein, ς_1 implies the task cost, χ signifies the data rate of task demands, and δ_d implies the holding up time of task demands.

(b) Data size

Data size as of a request can be estimated using the expression below,

$$\varsigma_2 = \frac{\Gamma_s}{1 + \Gamma_s e^2} \tag{15}$$

Wherein, ς_2 signifies the users' data size, Γ_s implies the total size of the users' request, as well as e signifies the allowed likelihood of committing an error in choosing a smaller representative of the user's request.

(c) Speed

By taking the turnaround time ratio of the request and the time of the job request, the speed of a task can well be ascertained. It is mathematically exhibited as

$$\varsigma_3 = \frac{\delta_a}{\omega_t} \tag{16}$$

Wherein, ς_3 implies the task speed, δ_a signifies the turnaround time of undertaking the demand, together with ω_t signifies the holding up time of task request.

(d) Disk space

The CS's disk space is the total utilized space and the quantity of free space available that is expressed mathematically as,

$$\varsigma_4 = \mathrm{O}_s + \Theta_s \tag{17}$$

Wherein, ς_4 signifies the CS's disk space, O_s and Θ_s implies the CS's free space as well as the used space.

(e) CPU Utilization

The quantity of work handled via a Central Processing Unit is called CPU Utilization. It is employed for device output estimation. It is rendered by:

$$\varsigma_5 = N_R * C \tag{18}$$

Wherein, ς_5 signifies the CPU utilization of the CS and N_R implies a busy time, together with C implies the capacity.

(f) Bandwidth

Bandwidth is termed the transferring rate of data, which is gauged via taking the product of numerous tasks with the applications' weight.

$$\varsigma_6 = \mathrm{T}_n * W_u \tag{19}$$

Wherein, ς_6 implies the CS's bandwidth, T_n signifies a number of tasks, and W_u signifies the usage weight.

(g) Processing speed

The total speed wherein a CPU can quickly retrieve and interpret instructions is labelled the processing speed, which is rendered as,

$$\varsigma_7 = \mathrm{I} * \mathrm{I}_c/\vartheta \tag{20}$$

Wherein, ς_7 signifies the CPU's Processing speed and I and I_c implies total tasks and average cycles per task, ϑ implies the clock rate.

(h) Number of requests

The total requests to the CS is implied as,

$$\varsigma_8 = \Re_1 + \Re_2, \ldots \Re_n \tag{21}$$

Wherein, ς_8 signifies the total requests set, and $\Re_1 + \Re_2, \ldots \Re_n$ implies the requests as of the user.

The extracted feature is built into an array as exhibited in the equation below:

$$\varsigma_{ext} = \left[\varsigma_1, \quad \varsigma_2, \quad \varsigma_3, \ldots \quad \varsigma_n\right] \tag{22}$$

3.2.4.2 *Feature selection*

Here, the utmost relevant feature needed for CS selection is done for eliminating the unwanted feature. This FS process lessens the time intricacy as well as renders vital features for training the classification model. A Deer Hunting Genetic Algorithm is rendered, which copes up with better FS for rendering an effectual FS.

3.2.4.2.1 Deer hunting genetic algorithm

A hybrid optimization algorithm is rendered by the FS algorithm for rendering a better outcome. More time is consumed by the prevailing algorithm, and it tends to attain a lower accuracy in terms of choosing the feature. To conquer the prevailing challenge, the work enables the working principle of crossover and mutation of the genetic algorithm into the deer hunting optimization algorithm, which ameliorates the accuracy together with computational time. The step by step execution of the FS algorithm is discussed below:

Step 1: Population Initialization

Initially, the population (count of features) of hunters (features) is initialized by the equation,

$$\breve{\mathrm{Y}}_{pop} = \left\{\varsigma_1, \quad \varsigma_2, \quad \varsigma_3, \cdots \quad \varsigma_n\right\} \tag{23}$$

Wherein, n signifies the total hunters, which is the solutions on the population $\breve{\mathrm{Y}}_{pop}$.

Step 2: Parametric Initialization

Wind angle (θ_w) as well as deer's position angle (θ_{DP}) are the fundamental parameters, which are initialized to ascertain the best solution. The wind angle follows the circle's circumference since the search space (working space) is regarded to be a circle.

$$\beta_i = 2\,\pi\,r \tag{24}$$

Wherein r signifies an arbitrary number with a value on the gamut [0, 1] as well as i signifies the current iteration. In the interim, the position angle of the deer (θ_{DP}) is rendered by,

$$\theta_{DP} = \theta_w + \pi \tag{25}$$

Step 3: Crossover and Mutation

The hunters are randomly chosen to generate a new populace or offsprings for obtaining the best solution. A crossover point is fixed in the cross-over phase and the healthy solution is found grounded on the crossover point. For obtaining better accuracy, the mutation process is carried out. Primarily, the preceding populace of hunters is chosen; centered on that, a new best solution-based population is created. The crossover of the population ($\hat{\zeta}_{crossover}pop$) is rendered by:

$$\hat{\zeta}_{crossover}pop = \hat{\zeta}_{crossover}pop\{\breve{\mathrm{Y}}_{pop}\} \tag{26}$$

A crossover population is obtained as of the initial population, which is rendered by the equation stated below:

$$\hat{\zeta}_{crossover}pop = \{\hat{\zeta}_1, \hat{\zeta}_2, \hat{\zeta}_3, \ldots \hat{\zeta}_n\} \tag{27}$$

Next, to attain the best solution over the crossover populace, mutation ($\hat{\zeta}^{new}_{crossover}pop$) is done, and it is rendered by:

$$\hat{\zeta}^{new}_{crossover}pop = \{\hat{\zeta}^{new}_1, \hat{\zeta}^{new}_2, \hat{\zeta}^{new}_3, \ldots, \hat{\zeta}^{new}_n\} \tag{28}$$

Thus, the best solution populace is attained and preceded further for attaining a global best solution.

Step 4: Position Propagation

The hunter with the aid of position propagation finds out the global best solution as of the optimal space stated as crossover populace. '2' solutions are deemed by the work, explicitly leader position, implied as (ζ_{lead}), which is the initial best position of the hunter, and successor position, ($\zeta_{successor}$), which is the successive hunter's position.

(i) Propagation through a leader's position

The best position is started to be attained by the hunter and position updation starts subsequent to defining an optimal space. The updating of the position is rendered as:

$$\hat{\zeta}_{i+1} = \zeta_{lead} - \gamma R|\Im \times \zeta_{lead} - \hat{\zeta}_i| \tag{29}$$

Wherein, $\hat{\zeta}_i$ signifies the position at the current iteration, $\hat{\zeta}_{i+1}$ implies the position at subsequent iteration, γ and $\Im$ signify coefficient vectors and R implies a random number developed regarding the wind speed, whose value gamut as of 0 to 2.

(ii) Propagation through the position of the successor

This phase is an exploration one. In the encircling population, the successor hunter value is also evaluated here. The global best solution can be searched through this, which is rendered below,

$$\hat{\zeta}_{i+1} = \zeta_{successor} - \gamma R|\Im \times \zeta_{successor} - \hat{\zeta}_i| \tag{30}$$

Wherein, $\zeta_{successor}$ signifies the successor position of the search agent as of the current populace. In every iteration, the search agents' location will be updated grounded on the best solution attained as of the arbitrary initialization of solutions. A search agent is arbitrarily selected when $|\Im| < 1$, and when $|\Im|^3 < 1$, the best solution is chosen for updating the agents' position. Therefore, the proposed work switches betwixt exploration and exploitation phases by means of the adaptive variation of the vector L. Therefore, the proposed work tends to achieve the best features, which are exhibited in the equation below:

$$\hat{\zeta}_{new}^{old} = [\hat{\zeta}_1^0, \hat{\zeta}_2^2, \hat{\zeta}_3^4, \hat{\zeta}_4^6, \hat{\zeta}_5^8, \hat{\zeta}_6^{10}] \tag{31}$$

Wherein, $\hat{\zeta}_{new}^{old}$ signifies the best feature chosen centered on old features.

Step 5: Termination

The position is updated at every iteration, till the best position is ascertained, which is the stopping criteria, centered upon the objective function.

3.2.4.3 *Selection of distributed cloud server*

The CS is selected centered upon the selective features since the existent method was complex to classify the CS. It led to misclassification of CS and lessened the accuracy rate. EK-ANFIS is developed for coping up with this challenge, which is depicted in Figure 4. Accurate classification of the CS is rendered by the EK-ANFIS grounded on the input features.

To classify the input feature, the EK-ANFIS is built layer by layer. Centered on certain rules, the model is executed, which comprises '2' fuzzy if-then rules. It is given as:

Rule 1: ***if*** $\hat{\zeta}_i$ is λ_i *and* $\hat{\zeta}_{i+1}$ is $\hat{\lambda}_i$, ***then*** $\Phi_i = s_i\hat{\zeta}_i + t_i\hat{\zeta}_{i+1} + \Im_i$

Rule 2: ***if*** $\hat{\zeta}_i$ is λ_{i+1} *and* $\hat{\zeta}_{i+1}$ is $\hat{\lambda}_{i+1}$, ***then*** $\Phi_{i+1} = s_{i+1}\hat{\zeta}_i + t_{i+1}\hat{\zeta}_{i+1} + \Im_{i+1}$

Wherein, $\lambda_i, \hat{\lambda}_i, \lambda_{i+1}$ and $\hat{\lambda}_{i+1}$ signifies the fuzzy sets, $\hat{\zeta}_i$ and $\hat{\zeta}_{i+1}$ denotes the input set (encrypted features and CS details). $\Im_i$, s_i, t_i, $\Im_{i+1}$, s_{i+1} and t_{i+1} values signify the parameter set as well as Φ is a '1st'-order polynomial and signifies the outputs of the '1st'-order Sugeno fuzzy inference system. '5' functional layers are present in I-ANFIS. Each layer's function is given as,

Layer 1: Here, each node is a square node with a node function, which is stated as,

$$\Theta_i^1 = \mathrm{M}_i(\Phi_i), \quad i = 1, 2, \ldots \tag{32}$$

Wherein, Θ_i^1 signifies the output of layer 1, Φ_i implies the compilation of input node, M_i signifies the Exponential Kernel Membership Function (EKMF). Exponential kernel membership (EKMF) function is performed to ameliorate the ANFIS's performance, which is computed as,

$$\mathrm{M}_i = \begin{cases} \exp\left\{\left(-\left(\dfrac{\tau - \hat{\zeta}_i}{\kappa}\right)\right\}\right., & \textit{for } \hat{\zeta}_i \leq \tau \\ \exp\left\{\left(-\left(\dfrac{\hat{\zeta}_i - \tau}{\nu}\right)\right\}\right., & \textit{for } \hat{\zeta}_i > \tau \end{cases} \tag{33}$$

Where, κ, ν, τ is the scalar value.

Layer 2: The firing strength for the rules is generated by every node in layer 2. The product of all incoming signals represents every node's outcome.

$$\Theta_i^2 = F_i = \mathrm{M}_i\hat{\zeta}_i \times \mathrm{M}_i\hat{\zeta}_{i+1} \tag{34}$$

Here, $F_i \rightarrow$ firing strength.

Layer 3: The normalization position to the firing strengths as of the preceding layer is represented by the output of every node.

$$\Theta_i^3 = (\bar{F}_i) = \frac{(F_i)}{\sum_{i=1}^{n}(F_i)}, \quad i = 1, 2, \ldots \tag{35}$$

Here, $(\bar{F}_i)$ implies normalized firing strength entropy value.

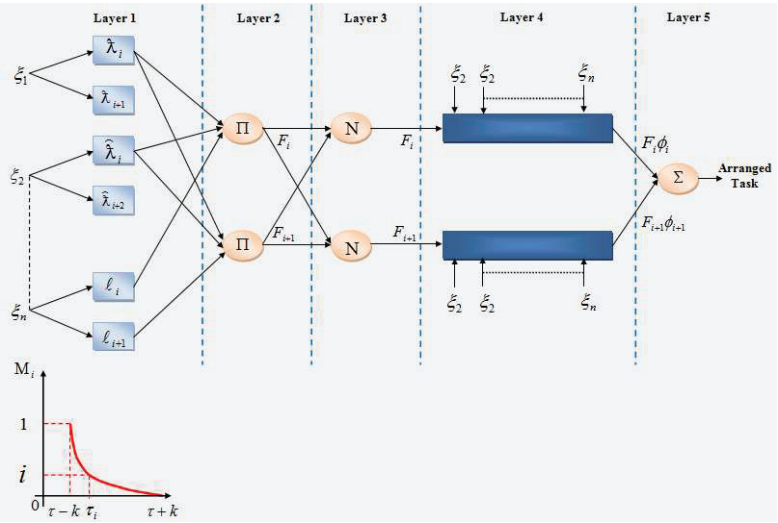


Figure 4 Structure of EK-ANFIS.

Layer 4: A firing strength output of the '1st' order Sugeno-type fuzzy if-then rule is offered by every node of this layer, which is given as,

$$\Theta_i^4 = (\bar{F}_i) \cdot \Phi_i \tag{36}$$

Layer 5: Only one node is possessed by layer 5 and it computes the sum of every output coming as of the nodes of the preceding layer for producing the overall EK-ANFIS output as in Equation (29).

$$\Theta_i^5 = \sum_{i=1}^{n} (\bar{F}_i) \cdot \Phi_i = \frac{\sum_{i=1}^{n} (\bar{F}_i) \cdot \Phi_i}{\sum_{i=1}^{n} (\bar{F}_i)} \tag{37}$$

Wherein, Θ_i^5 implies the overall output of EK-ANFIS. Explicitly, the output Θ_i^5 implies the best CS that is deemed as a chosen VM for executing the encrypted data.

Therefore, the proposed EK-ANFIS pseudo Code is exhibited in Figure 5.

Input: Encrypted File and CS features
Output: Cloud Server selection

Begin
Initialize the parameters $\kappa, \nu, \tau, \Im_i, s_i, t_i$
For i=1 to n
Generate fuzzy output from crisp input using,
$\Theta_i^1 \leftarrow$ Evaluate exponential membership function
If $(\hat{\varsigma}_i \le \tau)$
Evaluate Exponential kernel membership function using,
$$M_i = \exp\left\{\left(-\left(\frac{\tau - \hat{\varsigma}_i}{\kappa}\right)\right)\right\}$$
Else
Evaluate Exponential kernel membership function using,
$$M_i = \exp\left\{\left(-\left(\frac{\hat{\varsigma}_i - \tau}{\nu}\right)\right)\right\}$$
End if
Calculate firing strength for each rule using,
$$\Theta_i^2 \leftarrow M_i(\hat{\varsigma}_i) \times M_{i+1}(\hat{\varsigma}_{i+1}) = F_i$$
Normalize the firing strength using,
$$\Theta_i^3 \leftarrow \frac{(F_i)}{\sum(F_i)} = \overline{F_i}$$
Calculate Defuzzify the output of 3rd layer using,
$$\Theta_i^4 \leftarrow (\overline{F_i})(s_i\hat{\varsigma}_i + t_{i+1}\hat{\varsigma}_{i+1} + \overline{F}) = (\overline{F_i}) \times \Phi_i$$
Evaluate the output by converting fuzzy result into crisp output using,
$$\Theta_T^5 \leftarrow \sum(\overline{F_i}) \times \Phi_i$$
End for
End Begin

Figure 5 Pseudo code for EK-ANFIS.

4 Results and Discussion

Here, the performance estimation of the DKCPECC, the existing ECC, Modified Elliptic Curve Cryptography (MECC), Diffie-Hellman (DH) together with Rivest–Shamir–Adleman (RSA) is assessed. Via varying the file sizes as of 10 MB – 50 MB, the outcomes are estimated. The system is executed in JAVA with the system configuration as Intel i5/core i7 processor, 3.20 GHz CPU speed, along with 4GB RAM. The performance estimation was evaluated for statistical measures.

4.1 Performance Analysis

Centered on encryption time (ms), decryption time (ms), Memory Usage (MU) (kb) on encryption as well as decryption, upload time (ms), Download Time (DT) (ms), and SL (%) analysis phase, the performance examination of the DKCPECC cryptography techniques are exhibited. The file sizes are varied as of 10 MB to 50 MB for every statistical measures analysis. The comparison is done (a) Encryption and Decryption time analysis, (b) MU (kb) analysis, and (c) Upload, DT, and SL analysis.

Table 1 Demonstrate the performance of the proposed DKCPECC with the existing methods in terms of encryption and decryption time (ms)

	File Size (MB)	Proposed DKCPECC	ECC	MECC	RSA	Diffie–Hellman
Encryption	**10**	486	834	563	1054	1257
	20	1024	1532	1374	1668	1874
	30	2067	2510	2237	2754	2965
	40	2489	3056	2812	3275	3485
	50	4023	4533	4238	4765	4967
Decryption	**10**	486	834	612	1024	1245
	20	1035	1532	1243	1764	1975
	30	2074	2510	2268	2784	2987
	40	2512	3056	2786	3254	3457
	50	4035	4533	4268	4875	5047

(a) Evaluation of Encryption time (ms) and Decryption time (ms)

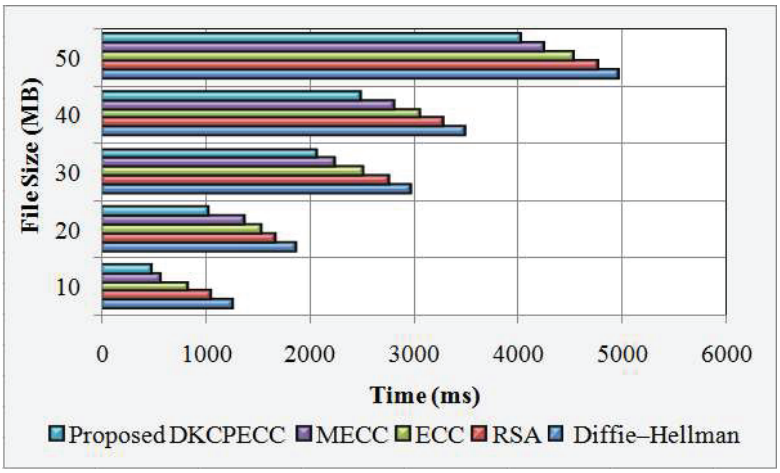
The proposed DKCPECC is analogized to the ECC, MECC, Diffie-Hellman, and RSA with regard to the time taken for the files' encryption together with decryption, which is tabularized in Table 1.

The proposed DKCPECC system's encryption together with decryption time (ms) with the ECC, MECC, RSA, together with DH is exhibited in Table 1. Centered on varying the file size (10 MB to 50 MB), the measured score is achieved. The proposed work takes 486 ms time for the 10 MB file size and the ECC, MECC, RSA, and DH took 834 ms, 563 ms, 1054 ms, and 1257 ms correspondingly for the encryption process. For the same process, the proposed work takes 4023 ms for processing the highest file size of around 50MB, while the prevailing works take 4533 ms, 4238 ms, 4765 ms, and 4967 ms. When considering the decryption process, the proposed DKCPECC, existing ECC, MECC, RSA, and DH algorithms take 4035 ms, 4533 ms, 4268 ms, 4875 ms, and 5047 ms correspondingly for the highest of 50 MB files. These are visualized in Figure 6(a) and (b).

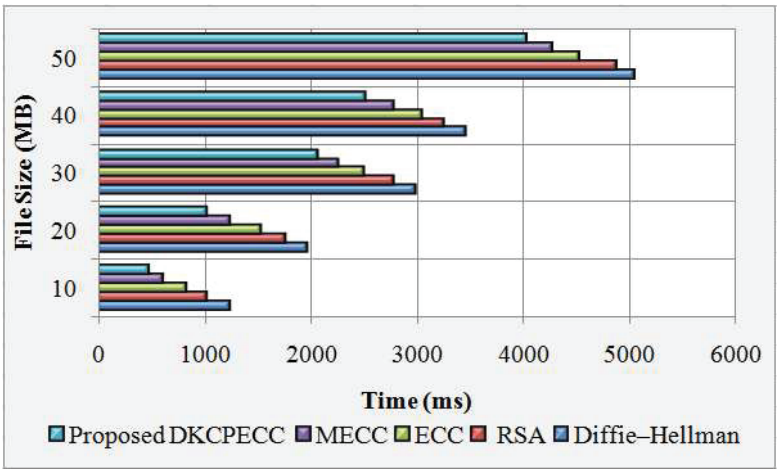
The time taken for encrypting as well as decrypting the varied file sizes is exhibited in Figures 6(a) and (b). The proposed work takes very little time aimed at the encryption in addition to decryption of any larger files size when analogized to the ECC, MECC, RSA, together with DH methods.

(b) Evaluation of Memory usage (kb) on encryption and decryption

The proposed and prevailing methods' performance for the MU is analyzed, which are pictured below.



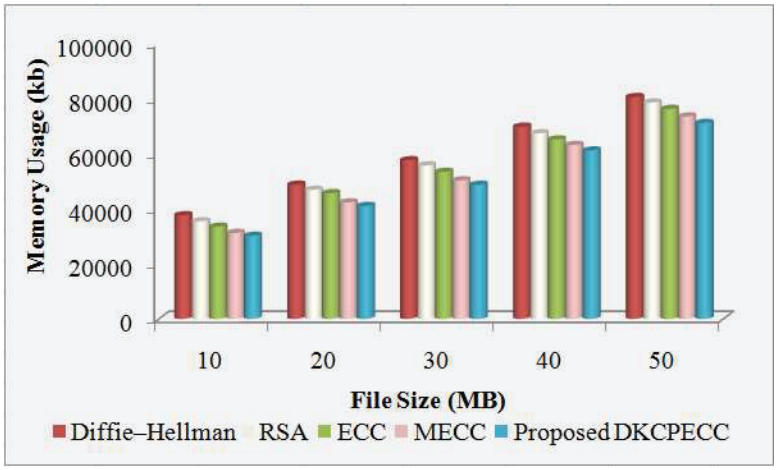
(a)



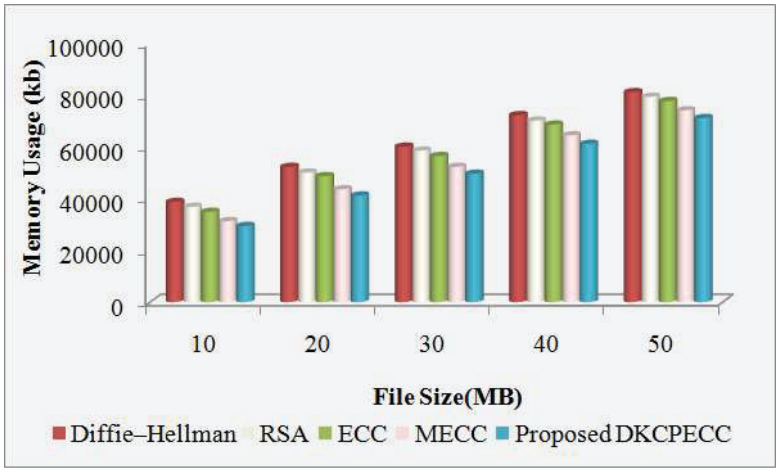
(b)

Figure 6 Graphical representation of (a) encryption time (ms), (b) decryption time (ms).

The MU (kb) on the encryption as well as decryption process for the proposed DKCPECC and the ECC, MECC, RSA together with DH is exhibited in Figures 7(a) and (b). Centered on varying the file size (10 MB to 50 MB), the measured score is achieved. The proposed system used 30175 kb memory for the 10 MB file size and the ECC, MECC, RSA as well as DH algorithms resulted in 33475 kb, 31224 kb, 35478 kb together with 37684 kb



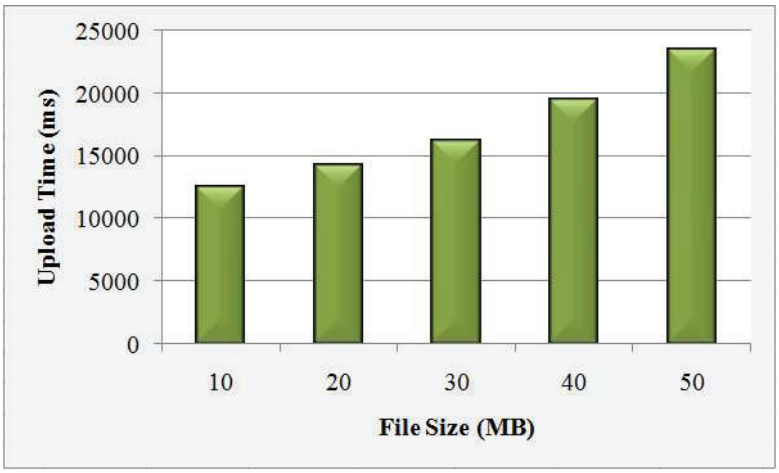
(a)



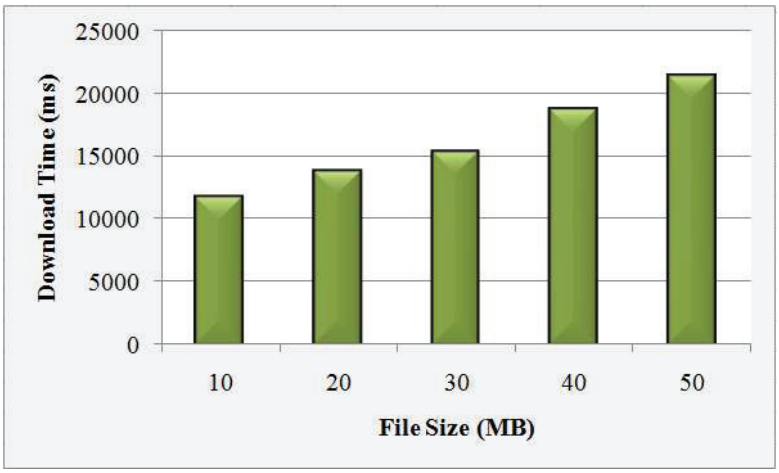
(b)

Figure 7 Visualization of MU (kb) for the (a) encryption process and (b) decryption process.

correspondingly for the encryption. For the same procedure, the proposed work consumes 71243 kb memory for the highest file size of 50 MB, while the prevailing methods take 76335 kb, 73553 kb, 78658 kb, together with 80658 kb. For decryption, the proposed DKCPECC, the existing ECC, MECC, RSA, and DH algorithms take 71245 kb, 77812 kb, 74223 kb, 79451 kb, and 81247 kb correspondingly for the 50 MB files.



(a)



(b)

Figure 8 Graphical representation of (a) upload time (ms), (b) DT (ms).

(c) Evaluation of Upload time, download time, and security level

The time consumed by means of the proposed work to upload and download the files in the cloud as well as their security analysis is visualized here.

The upload time and DT taken for the DKCPECC are exhibited in Figures 8(a) and (b). The upload as well as DT varies for every varied file size. Aimed at 10 MB file size, upload time takes 12447 ms and the DT

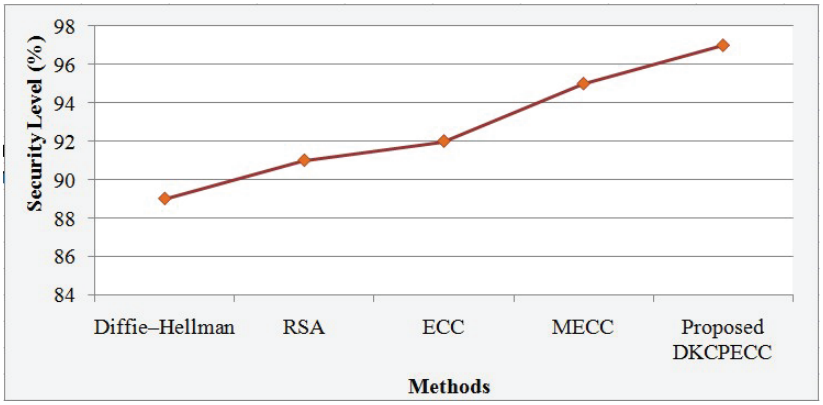


Figure 9 Visualization of SL of the proposed and existing techniques.

takes 11813 ms. Next, the upload and DT are 23441 ms and 21446 ms correspondingly for 50 MB file size. This exhibits the variation that occurred for every files size and also the proposed work's effectiveness.

The way the system is secured is exhibited by the SL analysis to shield the data as of any malicious or unauthorized access. The system's efficiency will be higher with the higher SL. The SL for the proposed together with existing techniques is exhibited in Figure 9. 97% security is attained by the DKCPECC wherein the ECC and MECC attained 92% and 95% correspondingly. While analogized to the prevailing methods, the system is much secured in the FI.

5 Conclusion

A robust cloud FI system intended for distributed CS is proposed, which copes with rendering secure evidence tracking as well as storing over the cloud devoid of any provocations. The proposed work maintains the evidence's trustworthiness by rendering a GK user authentication process. Whilst sharing over the distributed CS, the privacy of the client's data is maintained by the work, and also it also selects the best server grounded on the country utilizing the EK-ANFIS wherein the crime was committed. Stronger reporting of evidence, preservation of evidence together with a proper compilation of evidence in a distributed CS is built up by the proposed FI system. Efficient outcomes are attained by means of the proposed work with an SL of 97%, an average encryption time of 2244 ms, together with decryption time of 2235 ms. Analogized to the prevailing method, a better

outcome is attained via the proposed work. An effectual request handler for manifold requests in Secure Forensic can be added as well as the cloud interoperability betwixt the forensic clouds might be done in the future.

References

[1] Yong Zhang, Songyang Wu, Bo Jin, and Jiaying Du, 'A blockchain-based process provenance for cloud forensics', In IEEE 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 2470–2473, 2017, doi: 10.1109/CompComm.2017.8322979.

[2] Haider Al-Khateeb, Gregory Epiphaniou, and Herbert Daly, 'Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger', In Blockchain and Clinical Trial, Springer, Cham, pp. 149–168, 2019, doi: 10.1007/978-3-030-11289-9_7.

[3] Saurav Nanda, and Raymond A. Hansen, 'Forensics as a service: Three-tier architecture for cloud based forensic analysis', In IEEE15th International Symposium on Parallel and Distributed Computing (ISPDC), pp. 178–183, 2016, doi: 10.1109/ISPDC.2016.31.

[4] Mahmud Hossain Md, Ragib Hasan, and Shams Zawoad, 'Probe-IoT: A public digital ledger based forensic investigation framework for IoT', In INFOCOM workshops, pp. 1–2, 2018, doi: 10.1109/INFCOMW.2018.8406875.

[5] Victor R. Kebande, and Hein S. Venter, 'Novel digital forensic readiness technique in the cloud environment', Australian Journal of Forensic Sciences, vol. 50, no. 5, pp. 552–591, 2018.

[6] Jung Hyun Ryu, Pradip Kumar Sharma, JeongHoon Jo, and Jong Hyuk Park, 'A blockchain-based decentralized efficient investigation framework for IoT digital forensics', The Journal of Supercomputing, vol. 75, no. 8, pp. 4372–4387, 2019.

[7] Keke Gai, Jinnan Guo, Liehuang Zhu, and Shui Yu, 'Blockchain Meets Cloud Computing: A Survey', IEEE Communications Surveys & Tutorials, 2020, doi: 10.1109/COMST.2020.2989392.

[8] Deevi Radha Rani and G. Geethakumari, 'A framework for detecting anti-forensics in cloud environment', In IEEE International Conference on Computing, Communication and Automation (ICCCA), pp. 1277–1280, 2016, doi: 10.1109/CCAA.2016.7813913.

[9] Deevi Radha Rani, and G. Geethakumari, 'A framework for the identification of suspicious packets to detect anti-forensic attacks in the cloud

environment', Peer-to-Peer Networking and Applications, pp. 1–14, 2020, doi: 10.1007/s12083-020-00975-6.

[10] Shohreh Hosseinzadeh, Sampsa Rauti, Samuel Laurén, Jari-Matti Mäkelä, Johannes Holvitie, Sami Hyrynsalmi, and Ville Leppänen, 'Diversification and obfuscation techniques for software security: A systematic literature review', Information and Software Technology, vol. 104, pp. 72–93, 2018, doi: 10.1016/j.infsof.2018.07.007.

[11] Roberto Battistoni, Roberto Di Pietro, and Flavio Lombardi, 'CURE—Towards enforcing a reliable timeline for cloud forensics: Model, architecture, and experiments', Computer Communications, vol. 91, pp. 29-43, 2016, doi: 10.1016/j.comcom.2016.03.024.

[12] Deevi Radha Rani, and G. Geethakumari, 'Secure data transmission and detection of anti-forensic attacks in cloud environment using MECC and DLMNN', Computer Communications, vol. 150, pp. 799–810, 2019, doi: 10.1016/j.comcom.2019.11.048.

[13] Indumathi Jayaraman, and Amala Stanislaus Panneerselvam, 'A novel privacy preserving digital forensic readiness provable data possession technique for health care data in cloud', Journal of Ambient Intelligence and Humanized Computing, pp. 1–14, 2020, doi: 10.1007/s12652-020-01931-1.

[14] Nesrine Kaaniche, and Maryline Laurent, 'Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms', Computer Communications, vol. 111, pp. 120–141, 2017, doi: 10.1016/j.comcom.2017.07.006.

[15] Santosh Kumar, Sanjay Kumar Singh, Amit Kumar Singh, Shrikant Tiwari, and Ravi Shankar Singh, 'Privacy preserving security using biometrics in cloud computing', Multimedia Tools and Applications, vol. 77, no. 9, pp. 11017–11039, 2018.

[16] Deebak B. D., and AL-Turjman Fadi, 'Lightweight authentication for IoT/Cloud-based forensics in intelligent data computing', Future Generation Computer Systems, 2020, doi: 10.1016/j.future.2020.11.010.

[17] Jia Wang, Fang Peng, Hui Tian, Wenqi Chen, and Jing Lu, 'Public auditing of log integrity for cloud storage systems via blockchain', In International Conference on Security and Privacy in New Computing Environments, Springer, Cham, pp. 378–387, 2019, doi: 10.1007/978-3-030-21373-2_29.

[18] Anyi Liu, Huirong Fu, Yuan Hong, Jigang Liu, and Yingjiu Li, '$LiveForen$: Ensuring Live Forensic Integrity in the Cloud', IEEE

Transactions on Information Forensics and Security, vol. 14, no. 10, pp. 2749–2764, 2019.

[19] Kirti Dhwaj Singh, Ayushi Sharma, Shivali Singh, Vikram Singh, and SagarRane, 'Integrity and confidentiality preservation in cloud', In IEEE International conference of Electronics, Communication and Aerospace Technology (ICECA), vol. 2, pp. 419–424, 2017, doi: 10.1109/ICECA.2017.8212848.

[20] Jiamin Zheng, Qikun Zhang, Xiaosong Zhang, Yuanzhang Li, and Quanxin Zhang, 'A specific-targeting asymmetric group key agreement for cloud computing', Chinese Journal of Electronics, vol. 27, no. 4, pp. 866–872, 2018.

[21] Mehran Pourvahab, and Gholamhossein Ekbatanifard, 'Digital forensics architecture for evidence collection and provenance preservation in iaas cloud environment using sdn and blockchain technology', IEEE Access, vol. 7, pp. 153349–153364, 2019, doi: 10.1109/ACCESS.2017.

[22] Yuan Zhang, Xiaodong Lin, and Chunxiang Xu, 'Blockchain-based secure data provenance for cloud storage', In International conference on information and communications security, Springer, Cham, pp. 3–19, 2018, doi: 10.1007/978-3-030-01950-1_1.

[23] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla, 'Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability', In 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 468-477, 2017, doi: 10.1109/CCGRID.2017.8.

[24] Xuanyu Liu, Xiao Fu, Bin Luo, and Xiaojiang Du, 'Distributed Cloud Forensic System with Decentralization and Multi-participation', In International Wireless Internet Conference, Springer, Cham, pp. 181–196, 2017, doi: 10.1007/978-3-319-90802-1_16.

Biographies

Shaiqa Nasreen received her B.Tech in Electronics and Communication Engineering from Al-falah School of Engineering & Technology, Faridabad, Haryana, India, affiliated to Maharishi Dayanand University, Haryana, India in 2002 and M.Tech. Degree from CR State College of Engineering, Murthal, Haryana, India with specialization in Instrumentation and Control, in 2009. She is currently an Assistant Professor at Islamic University of Science and Technology, Awantipora, J & K, India. She is pursuing her Ph.D degree in Electronics and Communication Engineering from National Institute of Technology, Srinagar, J & K, India. Her current research interest includes Network Forensics and Security

Ajaz Hussain Mir received the B.E. degree in Electrical Engineering from Regional Engineering College, Srinagar, India, in 1982 and the M.Tech. degree in Computer Technology and the Ph.D. degree from Indian Institute of Technology Delhi, Delhi, India, in 1989 and 1996, respectively. He is currently a Professor with Electronics and Communication Engineering Department, National Institute of Technology, Srinagar, J & K, India. He is the Chief Investigator of Ministry of Communication and Information Technology, Govt. of India project: "Information Security Education and Awareness". He has published more than 90 research and review papers in reputed national and international journals.