Hindawi
Scientific Programming
Volume 2021, Article ID 7294206, 19 pages
https://doi.org/10.1155/2021/7294206

Research Article

An Intelligent Blockchain and Software-Defined Networking-Based Evidence Collection Architecture for Cloud Environment

Yunus Khan and Sunita Verma

Department of Computer Engineering, Shri G. S. Institute of Technology and Sciences Indore (RGPV), Bhopal, India

Correspondence should be addressed to Yunus Khan; yunuskhansgsits@gmail.com

Received 18 July 2021; Accepted 2 September 2021; Published 29 September 2021

Academic Editor: Punit Gupta

Cloud forensics is an extension of contemporary forensic science that guards against cybercriminals. However, consolidated data assortment and storage compromise the legitimacy of digital indication. This essay proposes an evolving modern algorithm automated forensic platform based on the blockchain idea. This proposes forensic structure design, evidence gathering, and storage on a blockchain that are peer to peer. Secure Block Verification Mechanism (SBVM) will protect unauthorised users. Secret keys are optimally produced using the cuckoo search optimization method. All data are saved and encrypted at the cloud authentication server for secrecy. Confidentiality-Based Algebraically Homomorphism, a new encryption method, is given to cryptosystem learning. Every data is assigned a block in the SDN controller, and the history is kept as metadata about data. Each block has a Secure Hash Algorithm version 3 of 512-bit hash-based tree. Our approach uses graph theory-based graph neural networks in Smart Contracts to track users' data (GNNSC). Finally, a blockchain-based evidence graph allows for evidence analysis. The experiments were run in a cloud environment with Python and network simulator-3.30 (for software-defined network). We achieved good results in terms of evidence response time, cloud evidence insertion time, cloud evidence verification time, computational overhead, hash calculation time, key generation times, and entire overall change rate of indication using our newly deliberated forensic construction using blockchain (FAuB).

1. Introduction

Cloud computing is an emerging technological concept that, through virtualization technology, provides users with physical resources. The cloud computing industry is growing with the benefit of allowing network accessing to a scalable and elastic combination of shared physical or virtual resources [1] with self-owned service provisioning and on-demand available services. There is also an enhancement in the number of cloud users using cloud computing because of these features. Security risks have begun to develop, however, with the rising cloud computing industry. Several security strategies for the cloud environment are being investigated with virtualization technologies, making it difficult to implement current digital forensic methods [2]. Access to certain system layers is restricted in Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) [3] environments when the cloud environment is categorized according to the service model access to that layer which are regulated by Cloud Service Provider (CSP). It is therefore appropriate to supply the log data generated in the inaccessible layer to the CSP through agreement [4]. Investigators have complete control over the evidence in conventional digital forensics. In a cloud environment, however, data centers are geographically distributed; cloud service customers (CSCs) exchange physical infrastructure, unreliable data that disappear when the instance is shut down, virtual network, load balancing, and auto scaling to provide a smooth service environment [5]. Therefore, prior to a security incident for investigation, it is important not only to record data for cloud forensics but also to guarantee the truthfulness of the log data, while it is impossible for the

investigator to directly capture the data and collect the data from the remote server. Forensic architecture is suggested for software-defined networking (SDN) using IoT [6] and blockchain. Blockchain's algebraic homomorphic encryption scheme is adapted here. Evidence data collection is performed in the presence of the SDN policy [7]. Digital evidence or stored on cloud using the data flow switches during the forensic examination. A (PDMS) data management system of provenance aware has been invented and built on the existing provenance monitoring framework [8]. Mchain [9] proposed an integrity management framework based on blockchain. Therefore, many analyses make an attempt that are distributed exploitation blockchain technology within the SDN cloud atmosphere [10]. During this analysis, within the cloud atmosphere, we tend to use the blockchain concept for cloud digital computer forensics. Forensic in cloud computing is an advancement of modern forensic science that protects against cyber criminals. Single centralize point compilation and storage of data, however, overcome the authenticity of digital evidence. In order to address this serious issue, this article suggests a modern automated forensic platform leveraging infrastructure as a cloud service (IaaS) based on blockchain concept. This proposed forensic architecture uses the blockchain technology to store the digital evidence and data are distributed among multiple peers. Secure Block Verification Mechanism (SBVM) is proposed to safeguarding the device from unauthorised users. Using the cuckoo search optimization algorithm for strengthening of the cloud environment, secret keys are optimally generated. On the bases of level of confidentiality, all data are stored and encrypted at cloud authentication server. Confidentiality-Based Algebraically Homomorphic Cryptosystems learning is presented with a fast-forwarding algorithm for encryption. A block in the SDN controller is created for every data, and information is stored in the cloud service provider, and the history is recorded as metadata about data. A hash-based tree is constructed in each block by the Secure Hash Algorithm version-3 of 512 bits. By implementing graph theory-based graph neural networks in Smart Contracts, our framework enables users to track their data (GNNSC). Finally, the construction of a Logical Graph of Evidence from blockchain data enables evidence analysis. Experiments were carried out in a Python for cloud and blockchain-integrated environment with network simulator-3.30 (for software-defined network). The proposed forensic architecture (FAuB) shows promising results in response time, evidence insertion time, evidence verification time, communication overhead, hash computation time, key generation time, encryption time, decryption time, and total change rate according to a comprehensive comparative study.

1.1. Research Contribution. In this article, the following contributions have been made to provide additional digital forensics research:

(1) In the case of cloud environment like infrastructure as a cloud service (IaaS), the digital forensics mechanism [11] design is constructed to collect, analyze, and release evidence. Blockchain technology is used to collect evidence.

(2) Evidence and information are secured against malicious users by using the Secure Block Verification Mechanism (SBVM) [12] driven by a cloud authentication server (CAS). The SBVM involves users who have completed successfully secure verification process by means of a globular logic and secret key (SK).

(3) Based on confidentiality level or the generation of digital signature [13] and encryption, the EL GAMAL algorithm is proposed. Key generation is done by the cuckoo search optimization algorithm in CB-EL GAMAL to generate strong secret keys. The main contribution of the Algebraically Homographic Cryptosystem algorithm based on confidentiality is that the proposed algorithm is based on the data level of sensitivity and adaptive in nature.

(4) Block was generated by control plane SDN and distributed across the blockchain network for all facts and statistics being deposited in the cloud-based server. For added security, a Secure Hashing-3 (SHA-3-512) algorithm has been proposed for blockchain accounts. By using neural network-based smart contracts (GNNSCs) on graph to track data activities throughout its life cycle, the data source is preserved.

2. Background

Siva Rama Krishna Tummalapalli [14] developed Bayesian fuzzy clustering and cluster search laid on support vector neural network-based intrusion detection mechanism simulator for clustering and two-level classifier working on cloud environment [15]. Saad Said Alkahtny developed a novel architecture to support forensic evidence collection and analysis of infrastructure as a service (IaaS) in cloud environment formally known as cloud forensic acquisition and analysis system without depending on cloud service provider and third party. This approach also provides the access of deleted data and overwritten data files which is not provided in existing forensic investigation techniques [5]. Zareefa and Mustafa found information obtained from the Zen Cloud Platform utilizing usable resources in the inquiry. Essentially the work focused on the three fields, such as adapting current techniques in the cloud world, gathering objects and data from the cloud, and assessing the interest of the information collected. In the near future, we will integrate existing tools of Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) (or all service type frameworks in one framework) as part of the future path. Finally, this work centered on and retrieved XCP with file system- dependent storage repositories (SRS) dependent on LVM [16]. Throughout their research, Philip and Clark applied mostly exif metadata found in JPEG image files. In the near future, all research studies will be carried out in specific other file formats such as pdf, text, excel, ppt, and others [17]. Ramakrishnan addressed the big emerging developments in

cloud computing protection and privacy concerns and often categorized security and privacy problems in security issues mainly, privacy issues mainly, and security issues intertwined [18]. In their work, Mhlupheki George and Sibiya explained the specifications for a cloud forensics framework and what standard procedures followed during the cloud forensic phase and how to build a cloud forensics system, as well as cloud forensics as a CFAAS architecture service [19].

In case of denial of service (DDoS), Alex and Kishore created a program that targets if the forensic management plane (FMP) gathers data regarding illegal forensic investigation activities. Throughout the immediate future, we should be able to execute the whole attack scenario throughout cloud platform [20, 21]. In their work, Ameer Pichan, Mihai Lazarescu, and Sie Teng Soh offered a systemic approach for examining cloud forensic problems, a potential answer for any process, and a description of forensic as a business model [22]. In their investigation, Vassil, Irfan, Andres, and Shane applied analysis and acquisition on SaaS and tested the results in their case studies. Kumodd: it is a tool used for the acquisition of cloud drives; Kumoocs: it is a tool for the acquisition and analysis of Google Docs; and Kumofs: it is a tool for remote previewing and cloud drive data screening [23].

Victor R Kebane built a cloud forensic preparation model as a test of the application software [7]. Grobler et al. suggested a six-dimensional virtual forensic approach to include the theory-based modern forensics solution [8]. Valjarevic and Ventor created a model consisting of three preparation phase assessments in the deployment and planning model. In ISO/IEC270 43 : 2015 [9], Valjarevic and Ventor built a model consisting of 3 preparation phase tests in the deployment and planning model [9]. Saad Said Alkahtny proposed a novel framework to assist IAAS cloud-based system (CFAAS) forensic discovery and analytics [10]. Alex and Kishore presented a forensic paradigm of denial of service (DDoS) assault for cloud storage and data processing utilizing forensic security plane (FMP) and FTK analyzer [11]. Emi Morioko, Mehdard S, and Sharbaf presented a method and algorithm for the procurement of Amazon Web Services (AWS) technical evidence [12]. Zareefa and Mustafa proposed a solution for accessing the recorded evidence value from the cloud and found an experimental result on Xen cloud platform [13]. Zachary, Katrina, and Kenji used snapshot submit Google Rapid Response (GRR) to plan and build automated forensic data acquisition system for forensic evidence collection [14]. In the cloud environment, Nhien An Le Khac, Michel Mollema, Robert Craig, and Steven Ryder are developing an innovative solution to data acquisition. We explain the legal context and address how to find the data center and deal with the actual job scenario of AWS [15]. Peng Xu, Yadong Zhang, and Kai Shuang deployed a modern streamlined data collection approach with hybrid data management review across the cloud logging (LOC) web service [24].

A cloud forensics tamperproof framework for cloud forensics is developed by the author that is available in a cloud environment that is untrusted and multitenancy. This framework relies on a forensic system based on the compressed multilayer counting filter [24], independent of daily cloud activity. No standard forensics preparedness model for cloud environments can be applied properly. A model for improving security [16] can be used in a cloud environment. Forensic preparedness is a way of maximizing the potential of an organization to respond to violations [17]. Figure 1 and Table 1 show that the number of papers published in various digital libraries like ACM, IEEE, ScienceDirect, Springer [16–23, 25–55], and Elsevier indicates that the lots of work have been done in the field of cloud forensics, and it is an active research area for the current cloud market.

Cloud logs will include useful data and information for the computer forensic investigation [18, 49], which is essential. Earlier designed logging systems have a few inconveniences to provide the cloud user with security. The existing system gives protection and security for user files that are either saved or uploaded by the user or authenticated [19] by the user. This paper secures logging by encrypting cloud logs using encryption techniques and identifying assaults on the cloud framework from DDoS (distributed denial of service) [25].

3. Evidence Collection

To classify and access forensic data from different parts and sources in the cloud world, the processing of evidence plays a critical role. Evidences are stored in one physical host, and data are split into another geographical region. Therefore, after an incident occurs, the evidence is very hard to find [26]. Proofs are obtained from different forensic origins such as switches, routers, servers, virtual machines, hosts, and browsers and from in-house storage content media such as hard disk drives, ram image files, and physical memory. The information is retrieved from multiple sources. Data collection from cloud servers, web browser objects [27], and physical memory analysis collects evidence.

3.1. Blockchain in Cloud Forensics. Blockchain is one of the overestimated breaking fields and has acquired significant consequences as an invention commonly used in numerous fields [20, 36]. The blockchain is known mostly as a billing book or digital distributed database [21]. The way blockchain interface, render device costs, monitor, and document transactions began to emerge as a revolutionary advance since its introduction in 2008. Blockchain [22] can be inexpensive, removing the do with to supervise and normalize transactions and communications [23] between various members of the central authorities. Other miners who have a record of the entire transaction history in a blockchain mark each move cryptographically [28, 50]. This renders time records that cannot be altered one by one safely, synchronized, and collective. Moreover, blockchain technology is considered IT and can be used in applications, industry, and industrial industries [29]. Figure 2 displays the blockchain design. The concept of blockchain consists of blocks like i to n numbers, current hash, and previous hash of the block; if hash value of any block is changed in blockchain network, it goes to invalid block and data tempering is detected.

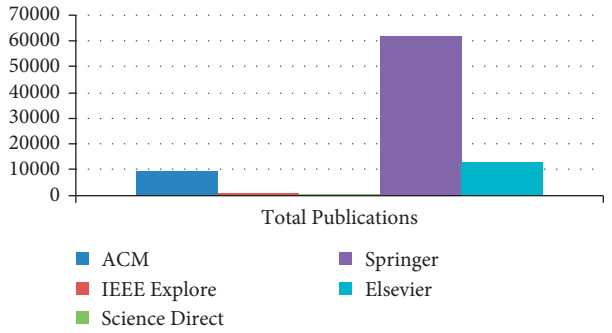

Figure 1: Year-wise analysis of research papers was published in digital libraries.

Table 1: Records of article types in various libraries on this topic.

	ACM	IEEE Explore	Science Direct	Springer	Elsevier
Journals	8994	209	197	683	506
Book chapters	469	3	17	60909	12551
Conference	70	698	3	509	80
Total publications	9533	910	217	62101	13137

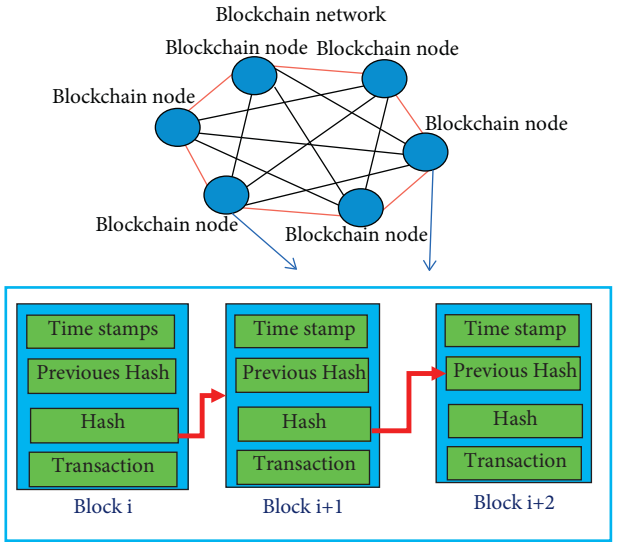

Figure 2: Architecture of blockchain technology.

4. Proposed Blockchain-Based Cloud Forensic (BCF)

The proposed forensic architecture, called blockchain-based forensics, is developed with the necessary algorithms in this section. The proposed forensic cloud uses software-defined network and blockchain concept collection of evidence and investigation.

4.1. Entities of the Architecture. The main objective of our experimental study is to acquire reliable proof or evidence in the cloud environment and to maintain the cloud provenance of data. The following entities comprise the overall forensic system:

(1) *Cloud Users (CU).* Cloud users (CU1, CU2, . . ., CU*n*) are included in our system "*n*" number. It is permissible for cloud users to save and access evidence at the server cloud.

(2) *Cloud Authentication Server (CAS).* At the start, the cloud clients are registered with CAS to deter unwanted access by users. Key generation and authentication are the major responsibilities of CAS.

(3) *Cloud Service Provider (CSP).* Cloud users store up all data in outer surface of their cloud on CSP hosted cloud servers. For every piece of data stored in CSP, a blockchain was developed.

(4) *Dataflow Open Switches (DFSs).* During this practice, a software-defined network is used to gather CSP data. We have therefore used many DFSs to relay CSP data to consumers. For data, the owned database flow regulations applied by the control plane to user DFSs may be mainly responsible. DFSs [R] only deploy and modify flow rules in the software-defined network control plane.

(5) *Software-Defined Networking Control (SDNC) Plane.* The software-defined networking control plane is responsible for applying network status data flow rules and for gathering all CSP evidence. The software-defined networking control plane manages blockchain for proof collection, and a block is generated for any CSP data. The complete machine architecture is seen in Figure 3.

Our forensic architecture's principal objective is to capture and conserve appropriate CSP data. We initially developed an efficient verification design to secure the device beginning unlicensed users. Data saved to the CSP are encrypted to ensure secrecy within the cloud setting. Decentralized data processing was planned based on blockchain technology.

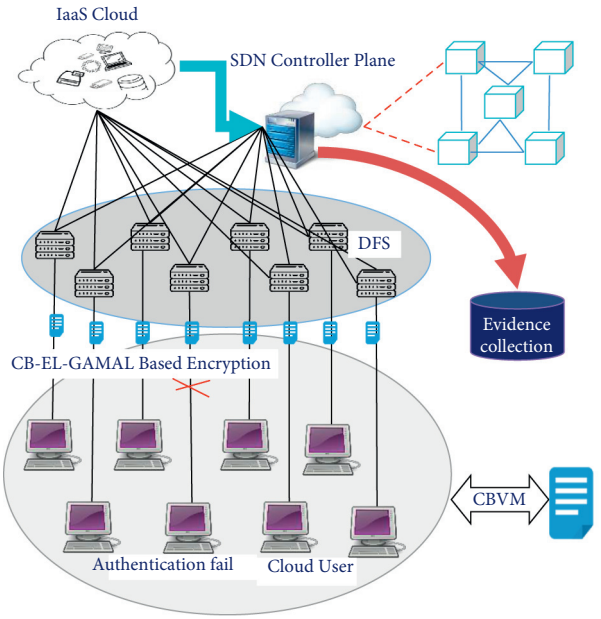

Figure 3: Blockchain-based cloud forensic (BCF) architecture.

Smart contracts can be used for the motto of recording and storing data history. For successful proof analyses, the graph-based research approach is recommended.

4.2. Cloud User Authentication. CAS is first registered with all cloud customers. User ID and password are user credentials that are taken into account when logging (PW). CAS produces a secret key (SK) for each documented CU by means of the cuckoo algorithm. Both users are valid at anywhere using the circular theorem's secret code (SC), SK, ID, and P.

4.2.1. Key Creation and Generation with the Help of Cuckoo Algorithm. The cuckoo search algorithm is a newly invented metaheuristic search optimization algorithm used to solve problems of optimization. This is a metaheuristic nature inspired algorithm focused on the brood parasitism of certain cuckoo birds, as well as spontaneous Levy flight walking. It has been carried out in a number of areas. The cuckoo algorithm is used in this research meant for the main generation of cryptography process.

The EL GAMAL equation is usually defined as follows.

Alice:

$$\begin{aligned} &\text{Choose the secret} 1 \le a \le p-11 \le a \le p-1, \\ &\text{Computer } A = \text{gamod} pA = \text{gamoda}. \end{aligned} \tag{1}$$

Alice sends the public key pk = (p, g, A) pk = (p, g, A) to Bob.

ElGamal is a public key cryptosystem dependent on the discrete logarithm issue for a gathering GG; for example, each individual has a key pair (sk, pk) (sk, pk), where sksk is the mysterious key and pkpk is the public key, and given just the public key, one needs to track down the discrete logarithm (take care of the discrete logarithm issue) to get the mysterious key. The cryptosystem is both an encryption plot (this part) which assists Alice and Bob with the issue of trading delicate data over an uncertain channel listened in by their enemy Eve and a computerized signature conspire (the following segment) which assists them with making advanced marks. The mark conspire is somewhat unique in relation to the encryption plot and different advanced mark plans; for example, the Schnorr signature plot and the digital signature algorithm (DSA) depend on ElGamal's unmistakable plan however with more limited keys. The public key created is as follows:

$$\text{Pu}(\text{SK}) = \text{Pr}(\text{SK}) \times P. \tag{2}$$

We may be capable of making out here the random generation of the private key (Pr(SK)) that attackers can crack quickly. The cuckoo algorithm is used to enhance the key generation process.

4.3. Cuckoo Search Explanations. Each egg is a solution in a nest, and a new solution is the cuckoo egg. The aim is to substitute not so nice nesting solutions with new and hopefully better solutions (cuckoos). The simplest shape of each nest is an egg [19]. The algorithm can be applied to more complex cases in which several eggs are present in each nest representing a set of solutions.

Three idealized rules are based on CS:

(1) Per cuckoo lays one egg on a single basis and dumps the egg into a randomly chosen nest.

The better nests with good egg content will hold the next generation.

(2) The number of available host nests is set, and host birds will possibly find the egg laid by a cuckoo.

(3) In this scenario, the host bird will throw away the egg/give up the nest and make a whole new nest.

For continuous nonlinear optimization, the cuckoo optimization algorithm is used. The lifestyle of the cuckoo family of birds is influenced by COA. This development optimization algorithm is based on the life style of these birds, their egg laying, and their breeding features. As other emerging approaches, a cuckoo optimization algorithm is introduced by an initial population. Here are two categories of cucumbers in various societies: mature cucumbers and larvae. The algorithm is based on the attempted survival. Any are discarded as they fight for life. The remaining cuckoos migrate to well again seats and begin raising and laying their eggs. Finally, the surviving cuckoos converge in such a way that there is a society of cuckoos with the same rate of profit.

To address the optimization issue, the variable values of the problem should take shape of an array. The "habitat" is called this array.

In an optimization problem, the next Nvar of a habitat will be a $1 \times \text{Nvar}$ array that shows the current living location of cuckoos. This array is described as follows:

$$\text{Habit} = [X_1, X_2 \ldots, X_n]. \tag{3}$$

The suitability (profit) of the current habitat is obtained by computing the function profit (p-f) in the habitat. Thus,

$$\text{profit} f \cdot b \cdot (\text{habit}) f.b[X_1, X_2 \ldots, X_n]. \quad (4)$$

To establish an optimization search algorithm, a habitat matrix of Npop ∗ Nvar size will be prepared and a random number of eggs will be allocated for each habitat.

Allowing for the number of eggs that every one cuckoo lays and as well as the space between the cuckoos and the current optimized zone, the laying radius will be calculated. After that, in that zone, the cuckoos start to lie. The laying radius is calculated as follows:

$$\text{ELR} = a \times \frac{\text{number of current cuckoos ggs}}{\text{total number of eggs}} \times (\text{Var}(hi) - \text{Var}(\text{low})). \quad (5)$$

Then, each cuckoo begins to lay her eggs in the nest within her ELR.

Thus, after each laying round, the less profitable percent of eggs (p%) (usually 10 percent) (their profit function is at the lower level) is destroyed. In the host nest, other chicks power up and develop.

4.3.1. The Cuckoo's Migration. While growing up and getting older, cuckoos live in their environments, but when the laying time comes, they migrate to superior habitats where the eggs have more chances to survive. The group with the best location will be targeted after composing the groups in different living locations in general (justified area or problem search space), and other cuckoos will migrate there.

When the cuckoos that are grown live all around the environment, it is not easy to determine which group belongs to each cuckoo. The cuckoos will be grouped by "K means" to solve this issue.

This method is actually a traditional method of grouping (finding a K between 3 and 5 is usually acceptable).

They do not travel the direct way when the cuckoos migrate to the target. With the deflection of (φ), they just travel (ƛ %, almost a percent) of the way.

These two parameters (φ) help cuckoos to explore a larger area. ƛ is a random number between 0 and 1, and φ is a number between (Algorithm 1):

$$-\frac{\mu}{6} + \frac{\mu}{6}. \quad (6)$$

In the method, the cuckoo algorithm selects an enhanced vector f(x) and is allotted to Pr (SK). Determining the secret key generated is difficult for cyber criminals because the cuckoo algorithm selects the random number more optimally.

4.3.2. Authentication Using Secure Block Verification Mechanism (SBVM). For those logged-in users, CAS produces secret keys and beginning points. For each operator of a particular circle, the root points are (Ox, Oy) co-ordinates. For each user in CAS, the respective credentials (ID, PW, and SC) are saved. In all stages of verification, all passwords are checked. The CAS key is a random code that makes it impossible for an attacker to invent the code for each user. By the following equation, a circle is defined as follows:

$$(Ax - Ox)^2 + (By - Oy)^2 = R^2. \quad (7)$$

Each user builds an SC consisting of origin points by using origin points (Ax, By). The user chooses an SC that follows the circle equation to effectively complete the authentication. While a client has to use the cloud, the client shall have each one ID and password along with the time stamp (TS).

Algorithms illustrate the method of SBVM-based authentication. A user who has legitimate passwords will complete the validation effectively. By making an allowance for SC next to TS, the protection level of the SBVM is increased. Although the SC differs over time, the attacker cannot split the SC. The attacker cannot use SC for the next authentication without being aware of the source points despite the SC being cracked at a time by the attacker.

4.4. Confidential Data Encryption. Users who have successfully completed the authentication process will enter the cloud computing environment in the planned forensic system. Within the cloud storage, users store their information in the form of ciphertext with extra security of digital signature. When mentioned in the prior paragraph, secret keys are produced by means of the cuckoo search algorithm. Data are translated into ciphertext by using the created strong secret key in the confidentiality encryption (CB- EL GAMAL) algorithm (Algorithm 2).

The EL GAMAL algorithm is paired through the CB-EL GAMAL algorithm probability and algebra. Algebraically homogenous crypto systems are a quick-release solution that is embedded in the decryption and encryption process across many unseen layers. The input layer of the homomorphic cryptosystem algorithm is used to encrypt, and Pu(SK) is initialized, and encryption is done on the secret layer. CB-EL GAMAL, however, is confidential and carries out the following data encryption procedures.

Algorithm 3 demonstrates the overall technique with an efficient hidden key for the CB-EL GAMAL algorithm [56]. By implementing graph theory-based graph neural networks in Smart Contracts, our framework enables users to track their data (GNNSC). The CB-EL GAMAL algorithm being proposed is shown in Figure 4. The neural network is used for the encryption process and calculating ciphertext in hidden layer for secret key generation, in which cryptosystems learning is a fast-forwarding method that is incorporated for the encryption and decryption process through multiple hidden layers [45].

Similarly, the input layer begins the ciphertext, and the output layer gets the original text when the data are decrypted. The participation in encryption of the Homographic Cryptosystems Algebraically algorithm [27] strengthens data security. To retain the documentation of possession, the data will be signed by the customer sooner than outsourcing to the cloud computing surroundings.

```
Start Function objective f (x), x = (x1), x2, . . ., XD) T;
Initial host nest population xi, I = 1, 2, . . ., n)
Duration or stop criterium (t < max generation)
   Get a cuckoo to Levy Flights by random means;
   Analyze Fi fitness
   Select a nest randomly between n (say j)
      If (Fi > Fj)
      Substitute j for the current result
      Finish If
   A fraction of the worst nests is deserted and new nests are created
   Maintain the right options (or quality solutions nests)
   Grading the solution and finding the right solution
   End for
   Posting and visualizing outcomes of processes
End Start
```

Algorithm 1: The Pseudo-code of Cuckoo Optimization Algorithm.

```
Input: password for users
Output: Status of authentication
(1) Begin
(2) For CU//Registration of Cloud User
(3)    Register ID, Password ⟶ CAS
(4)    CAS uses cuckoo algorithm to produce Secret Key (SK)
(5)    CAS provides SKs; Origin Points ⟶ CU
(6) End for//Registration completed by Cloud User
(7)    If Ui requires on right to use cloud//Require validation
(8)       Calculate secret code (SC) via equation (7)
(9)       CUi submits IDi, Password, SC, TS ⟶ CAS
(10)      CAS verifies User credentials
(11)         If (User Credentials are correct match)
(12)            Ui = Authorized user
(13)            Else
(14)            Ui = Unauthorized user
(15)         End if
(16)      Else
(17)         End process
(18)      End if
(19) End
```

Algorithm 2: SBVM authorization mechanism (Pseudocode).

Digital signature using the EL GAMAL algorithm generates the same as mentioned, and the hash value is first created to sign the data as

$$HV = \text{Hash}(D). \tag{8}$$

The digital signature is then created:

$$\text{signature} = \frac{HV + \Pr(SK).K2}{K1}, \tag{9}$$

where the random numbers are $k1$ and $k2$. The data have to be registered by the same data proprietor if data are updated or ownership

4.5. Efficient Collection of Evidence Using Blockchain Technology. In cybercrime, digital data are important source for analysis. The offenders will conceal their details and erase the evidence in a variety of parts of the infrastructure as a service cloud system. The key issue with the infrastructure as a service cloud infrastructure can be with the intention of data collection being spread on a wide scale. In comparison, cloud consumers monitor more than scholars, making it a difficult challenge to gather and preserve data. SDN and blockchain technologies are utilized in the proposed digital forensic infrastructure to gather and maintain cloud forensic data to combat all this issue. The evidence will be stored within the blockchain ledger under the control of the SDN control. In cloud forensics, some relevant meanings are as follows.

Evidence Integrity. Integrity of the evidence guarantees that the certificate reflects correctly the information contained in

```
Input: Public key and input data
Outputs: Ciphertext
 (1) Initialize public key (Pu(SK)) and Input data (d)
 (2)    If (d = Confidential)
 (3)    Split data d ⟶ d1 and d2
 (4)    For data d1
 (5)    Calculate ciphertext 1(c1) as,
 (6)    c1 = d1 φ d2
 (7)    End for
 (8)    For data d2
 (9)       Initiate Pu(SK), d2 at input layer
(10)       Calculate ciphertext 2 (c2) in hidden layer,
(11)       ca = k × P//k is a random number
(12)       cb = d2 + k × Pu(SK)
(13)       c2 = (ca, cb)
(14)       End for
(15)       Get ciphertext (c) as,
(16)       c = (c1, c2)
(17)    Else
(18)       For d
(19)          Repeat step number (8 to 13)
(20)       End for
(21)    End if
(21) End
```

Algorithm 3: CB- EL GAMAL (Pseudocode).

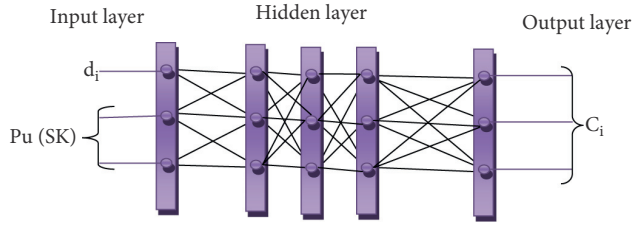

Figure 4: Neural network for encryption.

the PC. Several areas of the cloud influence knowledge respectfulness, but preserving integrity is a core component of the cloud crime scene investigation. The recognized technique to encrypt trustful information uses validated hash techniques, for example, MD5, SHA1, and SHA-256.

Data Origin. It is a form of authentication that corroborates a party as the (original) source of specified data generated in the past at some (typically unspecified) time.

Data Volatility. Unpredictability, after the power is switched off, memory or power failure of the material occurs. This is an important problem from a measurable standpoint since both memory and CPU procedures would vanish if the server crashes. If virtual computers are involved, these difficulties increase (VM). For ec IaaS, VM does not have permanent storage in this way; if the VM crashes, the volatile data may be lost.

Custody in Chain. The method of retaining and recording the chronological past of treating data as digital information can be represented. Data may be moved from the first responder, prosecutors (one or more), and judges to various layers of hierarchy of the automated forensics system. These provisional owners treat the proof during this lifetime. Because any evidence-based measure is held in the blockchain, our proposed work holds the custody chain.

Digital Evidence Ownership Proof. Digital evidence of ownership is defined here as the proof of existing digital proof of ownership. Multiple owners can manage the data during its lifespan. If the status of the data has shifted, the original owner must sign the data to retain the proof of cloud-based ownership. The patented evidence is retained in the framework as the transition in ownership is still preserved in the blockchain data history.

Graph Neural Network (GNN)-Based Smart Contracts. It is a computer program that tracks data history automatically. When the necessary conditions are met, the smart contract is activated and executed. To optimize smart contracts, graph theory algorithm rules are deployed in this work [56].

Data Lineage. It documents the history of possession and paper process throughout its entire life cycle. In other words, the record sequence showing the behavior taken from the data is known as a lineage or origin. With the aid of blockchain, we retain the data root; that is, in our work, any alteration made to the data is saved and traced by GNNSC in the blockchain.

The evidence has the hash value of the public ledger in the blockchain. We give a SHA-3-512 algorithm better in terms of security for hash value generation. The hash value in SHA-3 is determined accordingly for each block:

$$\text{hash} = \text{sponge}[g, pa\ d, q](T, L). \tag{10}$$

The hash unique value can be calculated here designed for input, that is, transaction (T) padding q, permutation g function, and output length L. The hash value is often created by the "sponge building" mechanism in SHA-3-512 as in EQATERY (10) rather than by the "sponge building" procedure. Accepting SHA-3-512 for hash calculations may bring various benefits over the current system with respect to time consumption and protection. Let us look at the $U1$ user's data $d1$ at time $t1$ in the cloud. After that, the block is formed by $d1$ and the hash value is created by SHA-3-512. Each transaction, i.e., the shift kept on $d1$, is based on the time the GNNSC block was installed in the system. Every update is processed and circulated as evidence in the blockchain network between the peers. The log contains the user name, IP address, time, and all other hardware information of the proof. The proof log, information history, is kept as the proof for each change found in detail in the blockchain. Past of data can involve lines that describe changes, ownership transition, and other behaviors on cloud-specific data. Algorithm 4 explains the method of collecting evidence. In favor of each single data residing within the cloud, the evidence can be gathered and preserved within the blockchain here. Furthermore, in the cloud environment, GNNSC tracks and wheels the ease of access of data stored by users.

During our initiative, we use smart contracts to alert cloud server when a graph theory law, which is often integrated as a proof record within the blockchain, is met. Many registered users will be able to the right of entry information contained in the cloud atmosphere. This thesis draws intelligent contracts from the graphology that functions on a secret stage of data. The smart contract is executed by means of the graph theory principles used in the framework. Figure 5 demonstrates GNNSC's pictorial representation. FSC presence tracks all big activities conducted under the data contained in the cloud server machine. Thus, any accurate evidence of the cloud server machine is gathered, and the correctness of evidence is conserved using blockchain technologies in our proposed forensic architecture.

Table 2 displays the laws of graph theory in GNNSC [57]. Because of these sets of laws, the statement is generated and saved like an evidence log. A modification of the data made after previous access is the previous danger. If the earlier hazard is restricted and information is nonconfidential, the log right of entry evidence will be overlooked and the report will not be produced. The produced statement is well thought out otherwise noteworthy and stored in the blockchain.

5. Cloud Forensic Investigation

If a cybercrime has been detected, the designated investigator (police and lawyers) must examine the digital evidence. CAS also authenticates the investigator prior to the inquiry. If a criminal enters an election voting room, his basic details, such as his Aadhar number and voter id, are kept in the election commission's database. If he tries to update or erase the evidence history by hacking the database, deleting, or modifying his entry into the voting space, he is attempting to upgrade or remove the evidence history.

Given that every one of the evidence record logs stored within a blockchain, we know that it is a distributed ledger and our suggested forensic architecture will be useful in this situation. It also passes the strong authentication before gaining access to the device. According to the investigator, the following steps should be taken when analyzing data.

5.1. Evidence Identification. The first step in a digital forensic investigation is to locate a possible evidence source of reliable evidence. As a result, the investigator must obtain legal consent from the relevant authority as shown in Table 3.

5.2. Evidence Acquisition. The investigator possibly will gather round all evidence log records of the blockchain by way of the consent of officially authorized authorities. The evidence log recorded inside the study contains mutual credentials of the user and evidence based on hardware. During this time, the investigators will have to adhere to court restrictions while also abiding by SLA agreements.

5.3. Evidence Analysis. The investigator then goes through all data logs and compiles a report on digital evidence. Logical graph with evidence for better research, this paper proposes a graph of proof. The evidence is used to build a graph of data with matching log attributes. If the perpetrator checks in at a polling site, submit the history of persons visited in the voting center, i.e., original details, just before the cloud to the administration of the election commission, i.e., a registered person. The evidence is currently being developed on blockchain for each one log record attribute (source_IP, timestamp, actions made, transaction hash, server of virtual machine, DFS_ID, and the like).

Think about the case where the suspect's check-in history was changed at $t2$. Then, in a subsequent block of log attributes, the next log is modified. Similarly, as soon as the hacker tries to access the information or erase it from the cloud, this should be treated as evidence and recorded in the subsequent block. The investigator must complete the following steps to create a graph of evidence:

(1) Sequentially arrange the evidence according to the timestamp

(2) Store each evidence through its attributes of log record

Input: cloud, user, data
Outcome: collected digital evidence
(1) Start
(2) For every CUi ϵ CU
(3) Creates Cloud users with GNNSC
(4) End for
(5) For every data
(6) $U1$ stores $d1$ in Infrastructure as a Cloud Service
(7) Create the block for $d1$
(8) Calculate Hash value ($d1$) with the help of Equation (10)
(9) Track $d1$ and modernize the evidence
(10) End for
(11) For every transaction on $d1$
(12) Store Log timestamp, source or origin IP, Visual machine disk filetransaction hashdetails, Virtual Machine server, actions made, etc.
(13) If (Graphtheoryrulesarenottrue)//GNNSC
(14) Report Generation
(15) Else
(16) Do not produce the report
(17) End of if
(18) End of for
(19) End

Algorithm 4: Efficient Evidence Collection Method (Pseudocode).

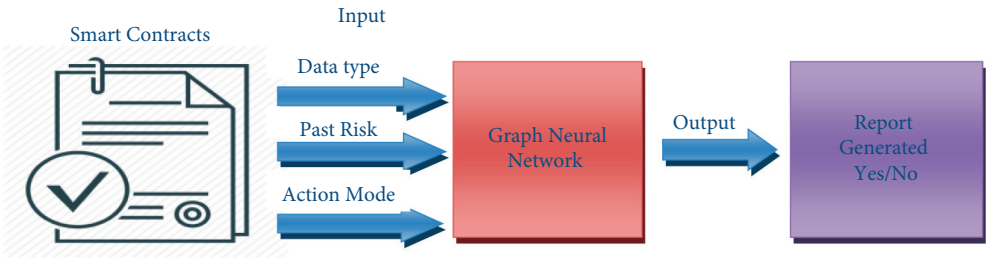

Figure 5: Pictorial representation of GNNSC.

Table 2: Attribute rules for GNNSC.

Data type	Past risk	Action performed	Report generation by GNNSC
Nonconfidential	Low	Read	No
Confidential	Low	Read	No
Nonconfidential	Low	Edit	No
Confidential	Low	Edit	Yes
Nonconfidential	Low	Delete	Yes
Confidential	Low	Delete	Yes
Nonconfidential	High	Read	No
Confidential	High	Read	Yes
Nonconfidential	High	Edit	No
Confidential	High	Edit	Yes
Nonconfidential	High	Delete	Yes
Confidential	High	Delete	Yes

Table 3: Evidence sample along with attributes.

Evidence identity (ID)	Different timestamps	IP_Source	Upload_User	Accessed user	Hash_Tn	Performed actions	Block hash	Location_Attribute	Virtual machine server	DFS
001	Ts1	192.168.10.xx	User A	User A	m-bits	Upload	n-bits	ZZZ	Pqrst	1
002	Ts2	192.168.10.xx	User A	User A	m-bits	Read	n-bits	ZZZ	Pqrst	2

Table 3: Continued.

Evidence identity (ID)	Different timestamps	IP_Source	Upload_User	Accessed user	Hash_Tn	Performed actions	Block hash	Location_Attribute	Virtual machine server	DFS
003	Ts3	192.168.10.xy	User A	User X	m-bits	Read	n-bits	ZZZ	Pqrst	3
004	Ts4	192.168.10.xx	User A	User X	m-bits	Edit	n-bits	zzz	pqrstklj	3
005	Ts5	192.168.10.xx	User A	User X	m-bits	Edit	n-bits	ZZZ	pqrstbvf	1
006	Ts6	192.168.10.xy	User A	User A	m-bits	Upload	n-bits	ZZZ	Pqrst	2
007	Ts7	192.168.10.xx	User A	User B	m-bits	Upload	n-bits	ZZZ	Pqrst	1
008	Ts8	192.168.10.xx	User A	User B	m-bits	Delete	n-bits	zzz	Pqrst	1

(3) Build an evidence graph according the evidence order and log record attributes

Table 2 shows properties of the survey evidence collection. A graph of evidence can be constructed using these data, as seen in Figure 6. The investigator can see from the graph of evidence that the suspect has edited (modified) the evidence (User *X*). However, the authorized user's location and IP addresses are different. Consider the case where the suspect's check-in history was changed at *t*2. Then, in a subsequent block of log attributes, the next log is modified. Similarly, when the suspect tries to hack these data or erase them from the cloud, this is treated as evidence and recorded in the subsequent block.

5.4. Reporting of Evidence. At the evidence review level, every one of the evidence within the graph of evidence is authenticated using a cryptographic digital signature that is kept together in the midst of the value of hash and data. Data should be signed earlier than being sent to the cloud according to our proposal. As a result, at what time an intruder could modify the evidence data, he or she should generate a digital signed signature.

For all evidence, the current transaction's hash value is stored at the blockchain. The hash significance of data stored in the cloud must match the Merkle tree root value of the block. The investigator compiles a report based on these findings and submits it to the court as a digital testimony. From acquisition to submission to juridictionary, algorithm, number 4 illustrates the collection process of evidence.

As a result, our designed architecture of cloud forensic, which incorporates blockchain and SDN technologies, allows for secure collecting evidence from the cloud. A strong authentication protocol stops unauthorised users from gaining access to the cloud environment, while a sensitivity aware encryption process improves data protection. Evidence storage using blockchain and SDN is an intellectual approach for distributed data protection. From evidence analysis to evidence reporting to the court, our designed architecture of cloud forensic facilitates the whole investigation.

5.5. Investigational Result Evaluation. Within this investigation result evolution, we compare the efficiency measurements of the designed architecture of cloud forensic with the earlier research contributions. We present our simulation environment in this section and at that time judge on our designed architecture of cloud forensic to the prior centralized log record process collection.

5.5.1. Configuration and Simulation. In a combined simulation platform, we configure our designed architecture for cloud forensic. Using CloudSim, we introduced an IaaS cloud environment in Python. Blockchain is the built data storage mechanism of IaaS cloud in Python Programming as described in the following Algorithm 5:

Both tests were run on Ubuntu OS by means of an Core-i7 Intel CPU running next to 2.80 GHz, 16 GB of RAM, and a 1000 GB SSD. The simulator version network 3.30 simulator, that is committed to network simulation for the software-defined networks, is also compatible with the cloud and blockchain environment. The Python platform's performance is merged by ns-3.30, in the direction to create a simulation environment.

The Ubuntu operating system underpins the entire work; we use NetBeans-8.2 for PYTHON blockchain setup, Network Simulator-3 for software-based network simulator, and CloudSim for IaaS cloud deployment.

Table 4 of our experiments explains the important parameters of simulation used in the direction of applying our designed architecture of cloud forensic. Prior to we get interested in the study, we will go through a real-world use of the proposed forensic scheme.

The Proof-of-Work principle is used by the miner to validate the blockchain. A corresponding block is generated for each piece of data that the user stores in the cloud environment and the stored hash values.

Use Case Diagram of Our Designed Architecture of Cloud Forensic Using Blockchain (FAuB). IaaS will be a cloud environment to be extremely versatile and can be used by any rising business. Many real-world implementations will benefit from our designed architecture of cloud forensic IaaS platform. In this paper, we look at one application of the proposed work in crime detection. Consider several voting centers that store their data such as voter records, financial information, maintenance information, personnel information, and surveillance information into IaaS cloud. Each data should be encrypted depending on top of the extent of data protection earlier than being outsourced to the cloud, as per our job. Furthermore, each voting center's administrator

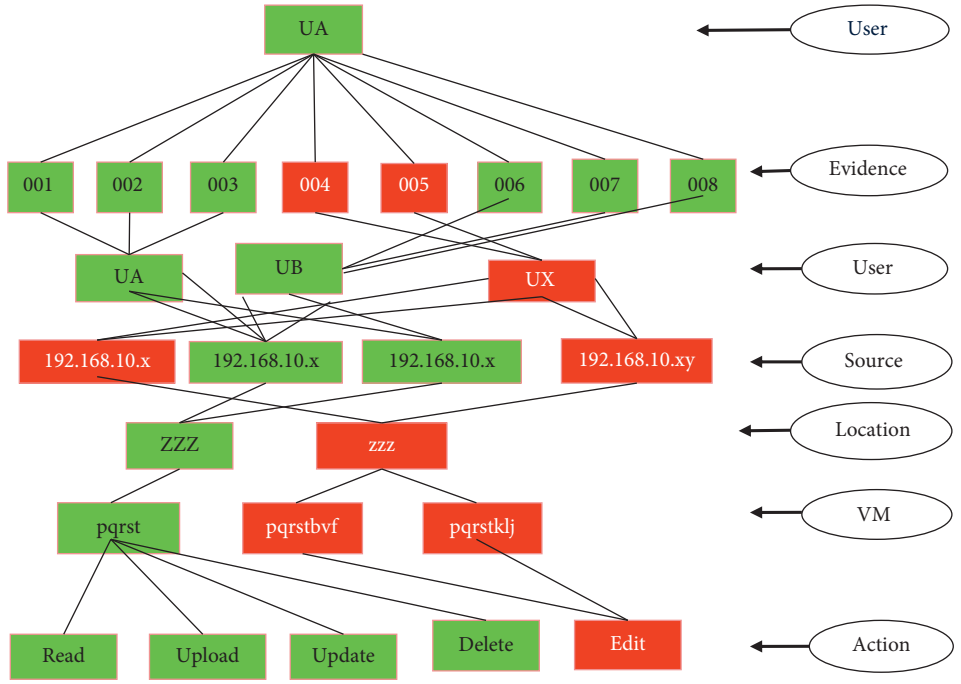

Figure 6: Evidence analysis scenario.

```
    Evidence as: input
    Evidence as: inputGraph of evidence as an: output
(1) Begin
(2)   Using the SBVM system, verify the investigator's identity.
(3)   Determine the case's evidence.
(4)   Collect evidence in the form of {Evidence_Identity, Time_stamp, IP_Source, uploaded_User, accessed_User,
    Performed_Actions, Hash_Tn, Block Hash, Location_Attribute, virtual machine server, and OFS ID} from blockchain.
(5)   Create an evidence graph by means of attributes of evidence.
(6)   For every one of the evidence
(7)     Ensure that {Block_Hash && IP_Source} are right and correct.
(8)     If this is the case (Verification D True)
(9)     Verify the signature//Validation of evidence
(10)      If this is the case (Signature is valid)
(11)        Collect reliable evidence
(12)      Else
(13)        Prepare illegitimate evidence
(14)      End if
(15)    End if
(16)  End for
(17) Prepare and share the copy of evidence with the jurisdictionary court.
(18) End
```

Algorithm 5: For forensic investigation.

must be CAS-registered. The SDN controller collects evidence designed for every one of the data stored within the cloud atmosphere and stores it on a blockchain. Additionally, each admin may use GNNSC to monitor their data.

Figure 7 depicts an example of the anticipated use case. Consider the case of a suspect who voted for a few hours at the polling center A. The suspect's information will then be found in the voting center A's election record file. Furthermore, video of the perpetrator in the polling center will be used in the data obtained from security cameras. This could aid detectives in locating the suspect as soon as possible. Any change made to the voter registration database and surveillance data is recorded within the blockchain as evidence. The perpetrator will erase or change the register of the voter registry and the data of surveillance contained inside the cloud if we do not have a good forensics mechanism architecture. Every evidence is preserved in the blockchain, that is, a distributed block ledger, in our

Table 4: Simulation configuration setting.

Parameters		Value
Number of users		120
Number of OFSs		8
Number of controllers		1
Number of cloud authentication servers (CAS)		1
Number of keys generated		120
Cuckoo	Maximum iteration	120
EL GAMAL	Number of hidden layers	4
	Key size	256
SHA-3	Block size	576
	Word size	64 bits
	Number of rounds	24
	Customized contract	GNNSC
	Maximum handles	2048
Cloud	Number of virtual machines	35
	Average RAM	512 MB
	Average bandwidth	1000000 MB
Simulation time	100 ms	

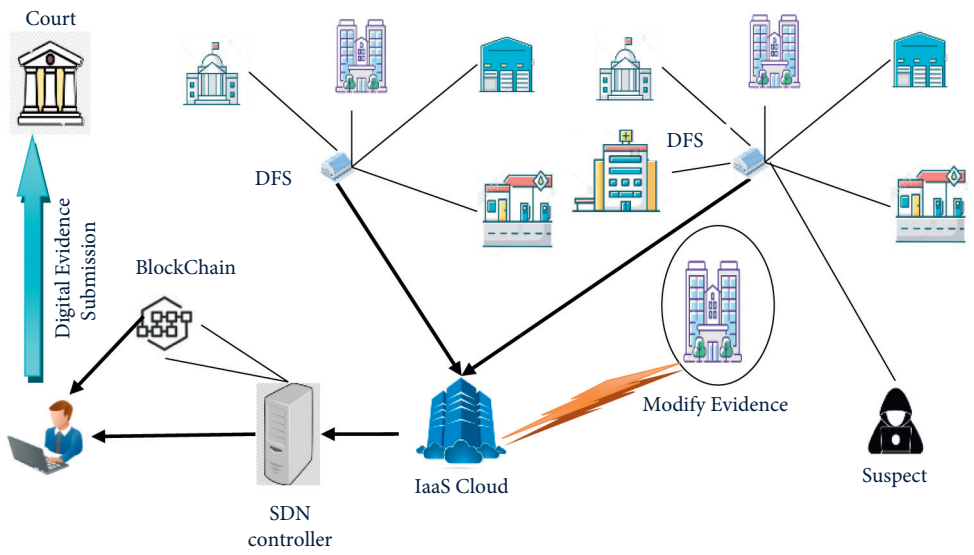

Figure 7: Digital forensic crime investigation case diagram.

proposed forensic architecture. We also store the VM logs in the blockchain as evidence. The investigator will obtain information from the blockchain even though the hacker changes and modifies the details on the cloud. Plotting an evidence graph with the collected data log will reveal whether there are any differences in the evidence. The investigator will pass the digital evidence from a CoC to the court based on the evidence obtained from blockchain.

5.5.2. Comparative Analysis. This section compares our designed architecture of cloud forensic to the current CFLOG [5] framework for safely collecting digital evidence. In CFLOG, the evidence is collected and stored in a centralized fashion, which is a major contrast between current forensic infrastructure and CFLOG. As mentioned in Section 3, this causes several problems. We designed an architecture of cloud forensic that collects in addition to storing digital information safely using SDN and blockchain technologies to overcome these challenges.

(1) Response Time Comparison. The time in use for users on the way to get a response to a data request is known as response time. The number of users interested in the forensic method validates this metric. In supplementary terminology, response time refers to the time it takes the forensic method to provide the necessary information or documentation to the users.

In Figure 8, the designed architecture of cloud forensic SDN-blockchain-based forensic framework is compared with the current CFLOG framework, which has a centralized framework. The numeral of requests of users increases by means of the increase inside the number of users in both

works, so the response time steadily increases with the increasing user numbers. Still, for more user number, our designed architecture of cloud forensic responds to the requested users easily. That use of software-defined network technologies improves the ability of scaling or the ability to accommodate a large number of users at the same time. As a result, any cloud user can link to the server of the cloud instantly as well as download the data requested by users. Similarly, the prosecutor should be able to obtain information from the blockchain without having to wait for the SDN controller to respond.

As a result, the proposed forensic architecture reduces the time of response. CSP performs together data managing as well as evidence collection in a consolidated fashion in CFLOG, which increases the response time when there are a large number of users. The CFLOG system takes 100 ms to answer in the presence of 120 users, while the designed architecture of cloud forensic system takes 72 ms intended for the identical amount of user numbers. As a result, the designed architecture of cloud forensic outperforms the CFLOG system by 27%.

(2) Evidence Insertion Time Comparison. The point in the time it takes to (or create) insert digital data of evidence collected on a server of the cloud is known as evidence insertion time. It can know how to exist and describe at the same time as the time it takes SDN plane controller to generate a proof meant for the CSP's stored data inside our analysis.

The insertion of evidence period as a function of the different user numbers is shown in Figure 9. When the user number grows, so does the volume of data that must be alive stored and the number of pieces digital evidence that must be generated. As a result, the amount of time it takes to insert evidence increases as the number of users increases in all works. Every one of the evidence should be unruffled and stockpiled in a consolidated way beneath the supervision of CSP in the CFLOG process.

As a result, the centralized evidence collection procedure lengthens the time it takes to insert evidence. In addition, we protect the history of data in our work, which means that each change to data is treated as evidence and incorporated into the blockchain. The SDN controller, on the other hand, is in charge of creating and preserving documentation without the intervention of CSP. As a result, relative to previous work, evidence insertion in blockchain takes less time.

(3) Evidence Verification Time Comparison. The time it takes an investigator in the direction of collecting and validating the evidence commencing a blockchain is known as evidence verification time.

The time taken for verification of evidence within the CFLOG process and the proposed forensic system is compared in Figure 10. The proposed automated forensic technology achieves the shortest possible time for evidence verification. The investigator would use CSP to collect evidence in the CFLOG process, and the verification is done in the conventional method. Instead of CSP, the investigator in the suggested work aggregates all evidence from the controller. In addition, for the improved studies, evidence testing is carried out by creating a graph of evidence. Furthermore, we suggested SHA-3-based hash computation to maintain evidence consistency while reducing time consumption. As a result, we gain evidence integrity with the least amount of time spent on evidence verification.

In the presence of ten users, CFLOG takes 62 milliseconds to collect and validate digital evidence, while the planned digital forensics FAuB takes just 37 milliseconds, reducing the verification time by nearly half.

(4) Computational Overhead Comparison. The bandwidth amount used in the direction of executing a particular activity (transfer data, reading, update, generation of evidence, and verification of evidence) within the system of forensic is known as computational overhead.

Figure 11 depicts a comparison of computational overhead based on different user numbers. Because the amount of data on the way to be interpreted grows in tandem with the number of users, the computational overhead increases. The computational overhead is raised in the absence of blockchain technologies owing to centralized device administration. Both data and evidence collection in CFLOG occurs in CSP, which raises the overhead.

The suggested forensic method, on the other hand, keeps indication processing like collection, hash reckoning, and conservancy on the SDN controller, reducing the total computational overhead. Furthermore, incorporating SDN technology increases scalability without adding overhead. Thus, the proposed digital forensic infrastructure adds 8 KB of overhead for ten cloud customers, while the CFLOG framework adds 10 KB of overhead.

(5) Total Change Rate Comparison. The rate of total change is calculated by dividing the amount of evidence modification by total evidence existing within the forensic framework facing problems with the old CFLOG system as shown in Figure 12. When a hacker person changes data to organize on the way to destroy evidence, the net modification rate rises. The collected data must be accurate, and the evidence's accuracy must be maintained for an effective forensic method. Since only registered users are included in the proposed forensics scheme, any information along with data of unauthorised users is refused. Furthermore, we use blockchain technology based on top of the SHA-3 algorithm to maintain the credibility of evidence.

According to our findings, the proposed forensic method modifies 11.1% of the evidence. However, since we guarantee credibility, CoC, and PoO for evidence, this alteration is also registered as evidence in the blockchain. Because (i) centralized infrastructure ever since CSP can be able to be malicious, (ii) node single vulnerability (an attacker just wants to break CSP's), (iii) no credibility is protected, as well as (iv) interference to unauthorised user's accessing, approximately 60% of evidence is changed in the CFLOG process. We overcome all issues by means of the help of blockchain and SDN technologies that reduces the system's overall change total rate.

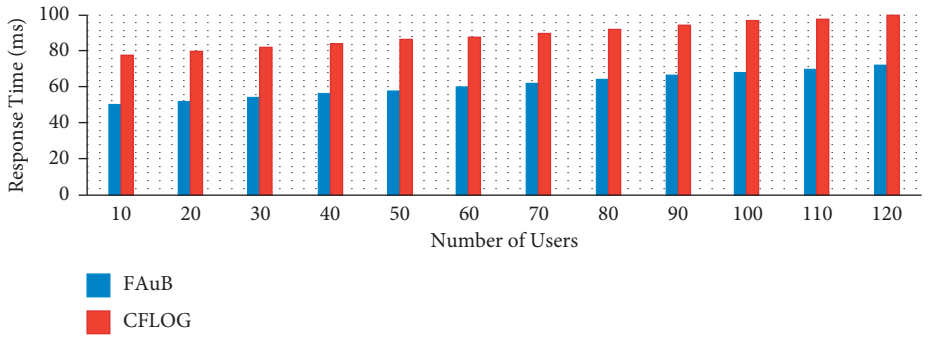

Figure 8: Response time comparison analysis.

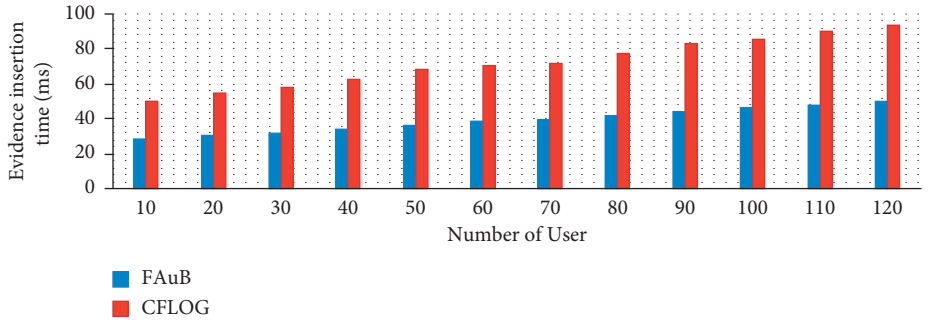

Figure 9: Evidence insertion time comparison analysis.

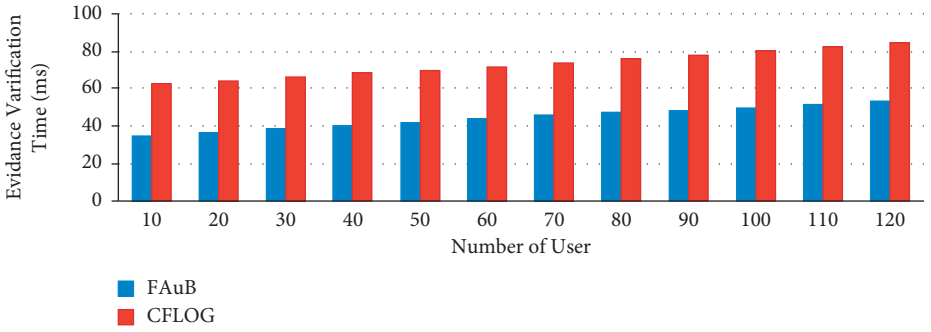

Figure 10: Evidence verification comparison analysis.

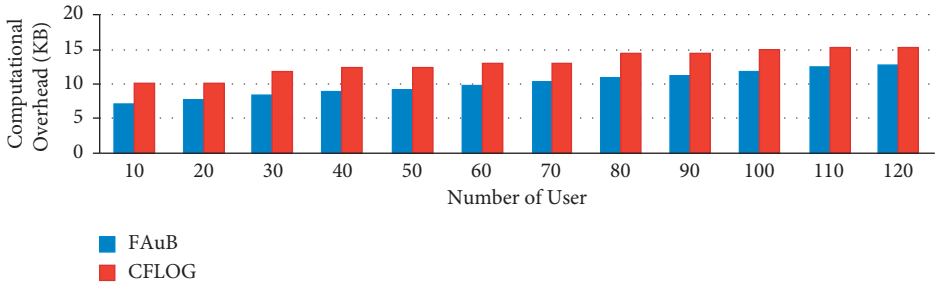

Figure 11: Computational overhead comparison analysis.

Table 5 compares the cumulative outcomes of the CFLOG process and the proposed forensic system in terms of performance measurements. We will see that each metric has improved with the proposed digital forensic FAuB architecture.

(6) Efficiency of CB-EL GAMAL with Cuckoo Algorithm. The elliptic curve cryptography (ECC) algorithm is regularly used design for digital signature concept in blockchain technology. On the other hand, there are

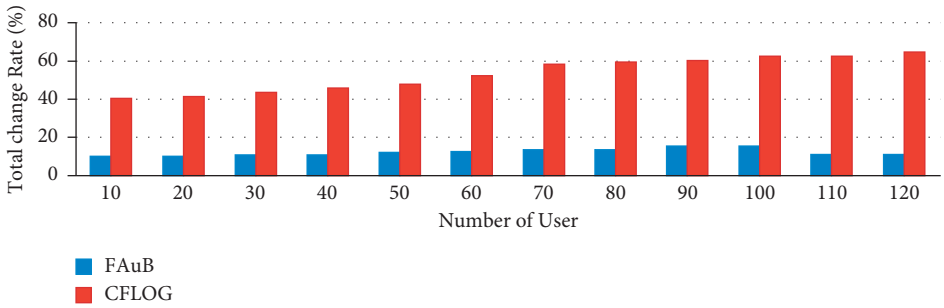

Figure 12: Total change rate comparison analysis.

several issues with key generation, encryption, and decryption. We suggested the CB-EL GAMAL algorithm with the cuckoo search optimization algorithm for key generation to improve the conventional ECC algorithm. As a result, we compare our proposed CB-EL GAMAL algorithm with the Paillier encryption algorithm proposed for blockchain technology using the cuckoo search optimization algorithm.

The suggested CB-EL GAMAL algorithm is examined in detail from Figures 13–16. For a stable blockchain architecture, the Paillier encryption algorithm is proposed. The Paillier encryption scheme, on the other hand, quickly improves key generation, encryption, and decryption times. The Paillier scheme consumes more time because it involves massive homomorphic computations.

On the other hand, data encryption is necessary in the environment of cloud and here the determination by several users. The algorithm of Paillier takes an average of 500 milliseconds to generate a key. Encryption and decryption, on the other hand, necessitate a significant amount of time, which is incompatible with the cloud environment.

The proposed CB-EL GAMAL algorithm, on the other hand, reduces the key generation time by using the cuckoo algorithm, which has a quick convergence time. Similarly, the CB-EL GAMAL algorithm's deep architecture reduces the time taken for encryption and decryption. As a result, the suggested SA-ECC algorithm outperforms the traditional algorithm in terms of increasing protection without increasing time consumption.

(7) SHA-3 Algorithm Efficiency. The most widely used hashing algorithm is used in blockchain technology. Hash computation in our proposed forensic scheme to increase the hash computation time and security standard is calculated by the SHA-3 algorithm.

Graph 10 compares the hash computation time of the proposed SHA-3 algorithm with that of the previous (SHA-256) 2 algorithm. In this review, SHA-3 reduces the calculation time of hash for 100 users to 16 milliseconds lacking sacrificing security. Inside general, SHA-3 outperforms SHA-256 against a variety of security threats, including length extension attacks. As a result, Merkle tree SHA-3 algorithm can construct a tree and increase protection without adding time to the process.

Overall, the proposed digital forensic FAuB architecture outperforms the current CFLOG scheme according to the report. The use of blockchain and SDN technologies increases the efficiency and scalability of the system.

Table 5: Analysis and comparison.

Performance analysis parameter	CFLOG	Proposed digital forensic architecture
Computational overhead time in KB	12.5	9.10
Evidence verification time in ms	70	42.1
Evidence insertion time in ms	71	44.2
Response time in ms	88.5	65.3
Total change rate in %	52	11.1

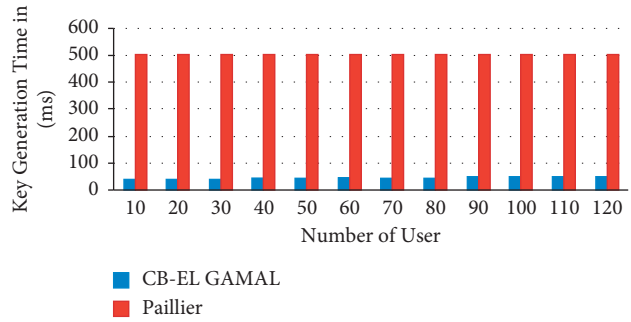

Figure 13: Key generation comparison analysis.

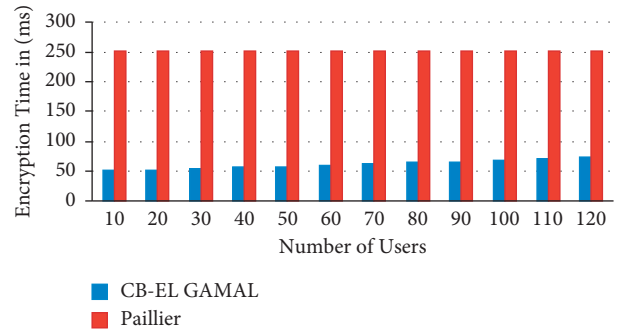

Figure 14: Encryption time comparison analysis.

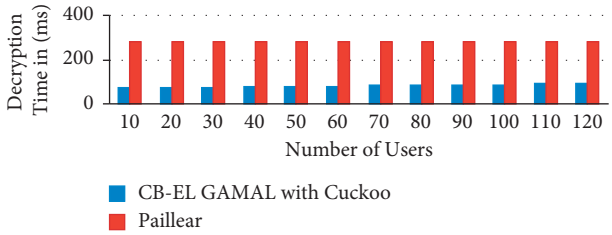

Figure 15: Decryption time comparison analysis.

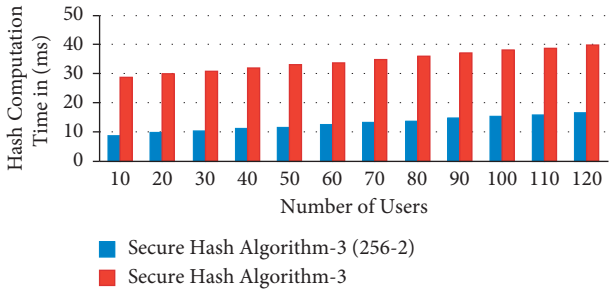

Figure 16: Hash computational time comparison analysis.

6. Conclusion

In this research work, with blockchain technology, a valuable architecture of digital forensic is proposed to gather and safeguard unfailing evidence from the substructure as a service cloud environment. Cloud authentication server CAS, with a secure verification mechanism known as the SBVM, authenticates all cloud users. The CB-EL GAMAL algorithm was proposed for data security. The cuckoo algorithm is proposed to generate secret key. A block in the controller is formed for every evidence stowed in the cloud. The integrity of evidence is ensured in every block by SHA-3-512-based hash tree building. All evidence is collected, and blockchain technology maintains evidence integrity, data origin, data link, digital evidence, ownership evidence, and custody chain. GNNSC is deployed in the system to trace data activities. The CB-EL GAMAL algorithm is proposed for data protection. The cuckoo algorithm generates optimum keys before that. At the controller, a block is spawned for each piece of cloud data. Merkle tree structure based on SHA-3 guarantees the consistency of evidence in each block. All documentation is collected, and the chain of custody and proof of ownership (CoC and PoO) are maintained using blockchain technology. GNNSC is installed in the system to monitor data events. Finally, the use of a graph for evidence analysis simplifies the evidence analysis. Overall, the forensic device is investigated using a Python and ns-3.30 simulation environment. Experimental findings suggest that the proposed forensic architecture outperforms the current unified forensic system. To improve the digital forensic infrastructure, we want to integrate network forensics in software-based networks as well as cloud forensics in the future [58–60].

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available but are available from the corresponding author who was an organizer of the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. A. Syed, M. Shahzad, and S. Farhan, "Analysis of cloud forensics techniques for emerging technologies," in *Proceedings of the International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, Tirana, Albania, December 2020.

[2] N. Kumar and I. Chana, "Load balancing and job migration techniques in grid: a survey of recent trends," *Wireless Personal Communications*, vol. 79, pp. 2089–2125, 2014.

[3] N. Rathore and I. Chana, "Job migration with fault tolerance based QoS scheduling using hash table functionality in social Grid computing," *Journal of Intelligent and Fuzzy Systems*, vol. 27, no. 6, pp. 2821–2833, 2014.

[4] A. Ahmed, F. A. Hany, and B. W. Gary, "Expert review of a cloud forensic readiness framework for organizations," *Journal of Cloud Computing*, vol. 8, p. 11, 2019.

[5] V. Sharma, R. Kumar, and N. Kumar Rathore, "Topological broadcasting using parameter sensitivity-based logical proximity graphs in coordinated ground-flying ad hoc networks," *Journal of Wireless Mobile Networks Ubiquitous Computing and Dependable Applications (JoWUA), SCOPUS indexed*, vol. 6, no. 3, pp. 54–72, 2015.

[6] A. K. Abdullahi, J. Aman, N. Y. Mohd, M. Aminu, K. I. Mohamad, and R. M. N., "Evidence collection and forensic challenges in cloud environment," *MACE Technical Journal (MTJ) MTJ*, vol. 1, no. 1, pp. 2710–6632, 2019.

[7] O. Akter, A. Arnisha, A. Akther, M. A. Uddin, and M. Manowarul Islam, "Cloud forensics: challenges and blockchain based solutions," *International Journal of Wireless and Microwave Technologies*, vol. 10, no. 5, pp. 1–12, 2020.

[8] N. Kumar, "Dynamic threshold-based load balancing algorithms," in *Wireless Personal Communication*, vol. 91, pp. 151–185, no. 1, Springer Publication, New-York, NY, USA, 2016.

[9] N. K. Rathore and I. Chana, "Job migration policies for grid environment," *Wireless Personal Communications*, vol. 89, no. 1, pp. 241–269, 2016.

[10] A. K. Samuel and J. Suhardi & Tutun, "Modeling cloud forensics readiness using MetaAnalysis approach," in *Proceedings of the IEEE, International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung, Indonesia, 2020.

[11] A. K. Haider, E. Gregory, and D. Herbert, "Blockchain for modern digital forensics: the chain-of-custody as a distributed ledger," in *Part of the Advanced Sciences and Technologies for Security Applications Book Series (ASTSA)*, Springer, Berlin, Germany, 2019.

[12] A. Akbarzadeh and E. Shadkam, "The study of cuckoo optimization algorithm for production planning problem," *International Journal of Computer Applications in Technology*, vol. 2, no. 3, 2015.

[13] N. K. Jain, N. K. Rathore, and A. Mishra, "An efficient image forgery detection using biorthogonal wavelet transform and improved relevance vector machine," *Wireless Personal Communications*, vol. 101, no. 4, pp. 1983–2008, 2018.

[14] N. Jain, N. Rathore, and A. Mishra, "An efficient image forgery detection using biorthogonal wavelet transform and improved relevance vector machine with some attacks," *Interciencia Journal*, vol. 42, no. 11, pp. 95–120, 2017.

[15] D. Choudhary and S. Malasri, "Machine learning techniques for estimating amount of coolant required in shipping of temperature sensitive products," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 10, pp. 67–70, 2021.

[16] N. K. Rathore, D. Pandey, R. I. Doewes, and A. Bhatt, "A novel security technique based on controlled pixel based encryption of image blocks for sharing a secret image," in *Wireless Personal Communication,* Springer Publication, New York, NY, USA, 2021.

[17] E. H. Ezz and D. H. Manjaiah, "An efficient digital forensic model for cybercrime investigation in cloud computing," *Multimedia Tools and Applications*, vol. 80, pp. 14255–14282, Springer, Berlin, Germany.

[18] R. Neeraj and I. Chana, "Variable threshold-based hierarchical load balancing technique in Grid," *Engineering with computers*, vol. 31, pp. 597–615, 2015.

[19] K. Mndeep, K. Navreet, and K. Suman, "A literature review on cyber forensic and its analysis tools," *International Journal of Advanced Research In Computer And Communication Engineering*, vol. 5, no. 1, 2016.

[20] L. Pradeep and N. Rathore, "Load balancing algorithm in distributed network," *Solid State Technology*, vol. 63, no. 2s, 2020.

[21] N. Jain, A. Mishra, and N. Kumar, "Image forgery detection using singular value decomposition with some attacks," in *National Academy of Science Letters,* Springer Publication, Berlin, Germany, 2020.

[22] P. Srivastava and A. Choudhary, "Evolving evidence gathering process: cloud forensics," in *Proceedings of the International Conference on Big Data, Machine Learning and their Applications*, vol. 150, Springer Nature Singapore Pte Ltd., Allahabad, India, July 2021.

[23] N. Rathore, U. Rawat, and S. C. Kulhari, "Efficient hybrid load balancing algorithm," *National Academy of Science Letters*, Springer Publication, Berlin, Germany, 2020.

[24] M. G. Al-Thani, D. Yang, and D. y. Yang, "Machine learning for the prediction of returned checks closing status," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 6, pp. 19–26, 2021.

[25] N. Kumar and P. K. Singh, "A comparative analysis of fuzzy based load balancing algorithm," *Journal of Computer Science*, vol. 5, no. 2, pp. 23–33, 2017.

[26] H. Singh and N. Kumar, "Analysis of grid simulators architechture," *Journal of Mobile Applications and Technologies (JMT)*, vol. 4, no. 2, pp. 32–41, 2017.

[27] N. Kumar, "A review towards: load balancing techniques," *Journal of Power Systems Engineering (JPS)*, vol. 4, no. 4, pp. 47–60, 2017.

[28] N. Kumar, "Efficient agent-based priority scheduling and load balancing using fuzzy logic in grid computing," *Journal of Computer Science*, vol. 3, no. 3, pp. 11–22, 2015.

[29] P. Liwen, L. Jing, and Li. Jin, "Information fusion-based digital forensics framework in cloud environment," in *Proceedings of the 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, IEEE, Chengdu, China, 2020.

[30] P. R. Brandao, "Computer forensics in cloud computing systems," *Budapest International Research in Exact Sciences (BirEx) Journal*, vol. 1, no. 1, pp. 71–86, 2019.

[31] N. Kumar, "Faults in grid," *International Journal of Software and Computer Science Engineering, MANTECH PUBLIATIONS*, vol. 1, no. 1, pp. 1–19, 2016.

[32] R. K. T. Siva and A. S. N. Chakravarthy, *Intrusion Detection System for Cloud Forensics Using Bayesian Fuzzy Clustering and Optimization Based SVNN*, Springer-Verlag GmbH Germany, part of Springer Nature, Berlin, Germany, 2020.

[33] R. Neeraj, "Installation of Alchemi.net in computational grid," *i-manager's Journal on Computer Science*, vol. 4, no. 2, pp. 1–5, 2016.

[34] R. A. Rahman, S. Masrom, S. Masrom, N. B. Zakaria, and S. Halid, "Auditor choice prediction model using corporate governance and ownership attributes: machine learning approach," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 7, pp. 87–94, 2021.

[35] K. Neeraj, "Ethical hacking & security against cyber crime," *Journal of Information Technology*, vol. 5, no. 1, pp. 7–11, 2016.

[36] F. Khan and N. Rathore, "Internet of Things a review article," *Journal of Cloud Computing*, vol. 5, no. 1, pp. 20–25, 2018.

[37] N. Kumar and F. Khan, "Survey of IoT," *Journal of Cloud Computing, ManTech Publication*, vol. 1, no. 1, pp. 1–13, 2018.

[38] N. Rathore, "Map reduce architecture for grid," *Journal of Software Engineering*, vol. 10, no. 1, pp. 21–30, 2015.

[39] A. Nahar and S. Sharma, "Machine learning techniques for diabetes prediction: a Review, 2020," *International Journal of Emerging Technology and Advanced Engineering (ISSN 2250–2459)*, vol. 10, no. 3, pp. 28–34, 2020.

[40] N. Kumar, "Checkpointing: fault tolerance mechanism," *Journal of Cloud Computing*, vol. 3, no. 4, pp. 27–34, 2016.

[41] F. Ye, Y. Zheng, X. Fu, B. Luo, X. Du, and M. Guizani, "TamForen: a tamper-proof cloud forensic framework," in *Transactions on Emerging Telecommunications Technologies*, p. e4178, John Wiley & Sons, Hoboken, NJ, USA, 2020.

[42] N. Kumar and J. Rathore, "Efficient checkpoint Algorithm for distributed system," *International Journal of Engineering and Computer Science (IJECS), E-ISSN*, vol. 1, no. 2, pp. 59–66, 2019.

[43] I. Chana and N. Kumar, "Checkpointing algorithm in alchemi.NET, pragyaan: journal of information technology, IMS dehradun," *IEEE, CSI and MPCET*, vol. 8, no. 1, pp. 32–38, 2010.

[44] A. Goel and R. K. Bhujade, "A functional review, analysis and comparison ofposition permutation based image encryption techniques," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 7, pp. 97–99, 2020.

[45] Neeraj, "GridSim installation and implementation process," *Journal of Cloud Computing*, vol. 2, no. 4, pp. 29–40, 2015.

[46] N. Kumar and I. Chana, "Report on hierarchal load balancing technique in grid environment," *Journal of Information Technology*, vol. 2, no. 4, pp. 21–35, 2013.

[47] S. Meshram, S. Kumar, and S. Shukla, "Enhanced robust and invisible of digital imageusing discrete cosine transform technique and binary shifting technique," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 10, pp. 113–118, 2020.

[48] D. Pandey, U. Rawat, and N. Kumar Rathore, "Distributed biomedical scheme for controlled recovery of medical encrypted images," in *Innovation and Research in BioMedical Engineering,* Elsevier, Amsterdam, Netherlands, 2020.

[49] N. Rathore, "Performance of hybrid load balancing algorithm in distributed web server system," in *Wireless Personal Communication*, vol. 101, pp. 1233–1246, no. 4, Springer Publication, New York, NY, USA, 2018.
[50] N. Kumar Rathore, "Checkpointing: fault tolerance mechanism," *Journal of Cloud Computing*, vol. 3, no. 4, pp. 27–34, 2016.
[51] R. Bhatt, P. Maheshwary, P. Shukla, P. Shukla, M. Shrivastava, and S. Changlani, "Implementation of fruit fly optimization algorithm (FFOA) to escalate the attacking efficiency of node capture attack in wireless sensor networks (WSN)," *Computer Communications*, vol. 149, pp. 134–145, 2020.
[52] M. Saad Hamid, N. A. Manap, R. A. Hamzah, and A. F. Kadmin, "Stereo matching algorithm based on hybrid convolutional neural network and directional intensity difference," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 6, pp. 87–96, 2021.
[53] D. Pathak and A. Verma, "Efficient and improved smart parking system based on IoT," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 3, pp. 22–27, 2020.
[54] D. A. Pereira, R. R. Muñoz, and R. R. Muñoz, "Information system for integrated medical records with access via IOT technology," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 4, pp. 6–17, 2021.
[55] E. D. Madyatmadja, T. R. Yulia, T. R. Yulia, D. J. M. Sembiring, and S. M. B. P. Angin, "IoT usage on smart campus: a systematic literature review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 5, pp. 45–52, 2021.
[56] K. Vijayalakshmi, "Comparitive approach of data mining for diabetes prediction and classification," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 2, pp. 19–26, 2020.
[57] V. K. Gugulothu and S. K. Mohan Rao, "Classification of IRS LISS-III IMAGES by usingartificial neural networks," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 4, pp. 24–31, 2020.
[58] Y. Peng and Z. Zheng, "Spectral clustering and transductive SVM based hyperspectral image classification," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 4, pp. 72–77, 2020.
[59] N. R. Adytia and G. P. Kusuma, "Indonesian license plate detection and identification using deep learning," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 7, pp. 1–7, 2021.
[60] R. Chakraborty, S. Sanyal, and P. Das, "IoT based thermal signature detector with alarm & e-mail notification with integrated social gathering screening using computer vision," *International Journal of Emerging Technology and Advanced Engineering*, vol. 10, no. 4, pp. 164–171, 2020.