I. J. Computer Network and Information Security, **2026, 1, 116-130**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.01.08

Innovative Forensics in IoT Clouds by Leveraging Blockchain for Data Integrity and Security

Ragu Gnanaprakasam*
Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu District 603203, Tamil Nadu, India
E-mail: mailtoragug2022@gmail.com
ORCID iD: https://orcid.org/0009-0007-7465-6247
*Corresponding author

Ramamoorthy Sriramulu
Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu District 603203, Tamil Nadu, India
E-mail: ramamoos@srmist.edu.in
ORCID iD: https://orcid.org/0000-0002-7866-1218

Poorvadevi Ramamoorthy
SRM Institute of science and Technology, Kattankulathur, Chengalpattu, India
E-mail: poorvadr@srmist.edu.in
ORCID iD: https://orcid.org/0000-0002-4387-1506

Mervin Retnadhas
Department of Information Technology, College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia
E-mail: m.mary@seu.edu.sa
ORCID iD: https://orcid.org/0000-0002-8587-0624

Received: 19 August 2025; Revised: 20 September 2025; Accepted: 16 October 2025; Published: 08 February 2026

Abstract: The Internet of Things and cloud computing are expanding at a very rapid rate, which has posed a great challenge in maintaining data security and integrity, particularly during forensic investigation. The conventional logging mechanisms are prone to manipulation, unreliable, and difficult to verify the digital evidence. In response to these problems, a blockchain-based system is suggested to facilitate the security and reliability of forensic data stored on cloud-based environments, which is related to IoT devices. The decentralized storage is paired with the smart contract technology to form an immutable version of the cloud communications to make sure that the evidence is unaltered to guarantee its verifiability. It further has a safe off-chain storage system enabling swift records and recalls of massive forensic records. The enormous amount of experimentation has demonstrated that the system minimizes the verification times to about 28 to 39 milliseconds. It is quicker than the methods that are currently in place and has high data integrity. The framework enhances transaction throughput as well as provides scalable solution to preserve forensic evidence. It has offered a feasible and reliable platform to enhance the security, visibility and reliability of forensic data within intricate IoT and cloud environments. These characteristics aid law enforcement groups and forensic investigators in having effective and credible investigations.

Index Terms: Forensic Data, Blockchain, Cloud Storage, Security, SBLaaS.

1. Introduction

Currently, with advanced technology, a lack of security is a consequence of the growing demand for cloud

infrastructure among businesses, governments, and individual users. Cyberattacks pose a risk to all users' private data when they use cloud services [1]. The National Institute of Standards and Technology defines forensics in a digital manner, which includes a field of research that seeks to identify an occurrence, gather evidence, and analyse it. The system can also define cloud forensics as the application of digitised forensic science, primarily within the context of cloud computing [2]. Cloud forensic architecture, such as CURE, performs the task of collecting reliable evidence in a cloud environment. Clustering Using Representatives (CURE) focuses primarily on three security components: time heartbeat, anti-forging methods, and key management [3]. The model has developed a forensic architecture for IoT applications, using software-defined networking as the foundation. The system has adopted a linear homomorphic encryption technique for use in the blockchain [4]. The model also carries out factual data collection on the software-defined networking platform. Authenticating users is the first and most important step in the cloud forensic process to guarantee a high degree of security. In this case, the primary objective is to shield the evidence from unauthorised access. The system employs email verification and a one-time password [5] for efficient authentication. Perhaps the smart contract-based forensic architecture is safe, but rigorous authentication is necessary for the provenance of evidence [6]. The Elliptic Curve Cryptography (ECC) technique performs better in terms of the amount of time required for encryption and the size of the key. However, the initial key generation procedure determines the efficacy of an ECC algorithm. Inappropriate key creation may compromise the ECC algorithm's ability to offer a certain degree of security [7-9]. Therefore, blockchain applications have proposed a signature system based on homomorphic computations to enhance security. The model conducted a Strengths, Weaknesses, Opportunities, and Threats Analysis (SWOT) on the existing forensic infrastructures in the cloud. The findings of the research indicate that a dependable forensic system is essential in a cloud-based setting [10]. The model conducted forensic analysis on cloud and Internet of Things (IoT) devices for evidence identification. Blockchain-based smart contracts operate autonomously, carrying out their terms automatically when certain conditions are met [11]. The origin of one's data is another obstacle to overcome in a cloud setting. The term "data provenance" refers to the origin of the data that is kept in cloud-based infrastructure and any historical context associated with it [12]. The following is a list of the primary contributions that the proposed research has made like it provides a review of the challenges and constraints that digital forensics encounters in the context of the Internet of Things. Blockchain technology provides a model structure for conducting forensics studies using the Internet of Things and the system used a novel technique to establish the working prototype, infer the findings, and propose future research.

2. Related Works

An authoritative body in a central location administers the business on behalf of all consumers. In this procedure, the ledger serves as the only method capable of ensuring the accuracy of the financial information about clients. On the other hand, cryptocurrency uses blockchain technology as a mechanism to preserve authenticity instead of depending on a central bank [13–14]. Each participant in a cryptocurrency transaction shares their ledgers with other participants and exercises control over them. This helps to guarantee that the ledgers are completely transparent. The presence of additional participants ensures the transparency of participant-to-participant transactions [15]. Additionally, the approval of all other participants generally prevents dishonest individuals from repeating or refusing transactions. The distributed ledger's level of transparency allows us to categorize blockchain technology into three primary categories: private, public, and consortium blockchains. All participants in a public blockchain are required to share and collectively administer their ledgers [16]. In this specific blockchain implementation, each participant assumes the role of a subject, responsible for overseeing and verifying the ledgers. Participants can monitor and explore ledgers, earn incentives, and share similar distributed ledgers at any time within a public blockchain [17]. However, since this particular kind of blockchain does not have a central authority, it is extremely difficult to make any modifications or updates to the blockchain system. Another reason for its inefficiency is the lengthy transaction verification procedure. The private blockchain features a centralized authority that oversees the management of all members' ledgers [18]. Inclusion together in a commercial blockchain network is only feasible if the governing body of that network grants permission for it. This type of blockchain, with its central authority, allows for relatively simple modifications or upgrades when necessary. The consortium uses a blockchain that combines formal and informal elements [19]. Only validated users can carry out transactions in this type of blockchain.

When collected in a centralized manner, logs become more vulnerable to attacks. The forensic collection and analysis system provided a paradigm to address the issue of reliance on cloud service providers (CSP). Function as a Service (FAAS) had been an agent-assisted system, and the action coordinator, including the agent manager, was responsible for managing and controlling all of the recorded evidence [20]. The inability of FAAS to maintain data provenance results in the loss of an important aspect of forensics. In addition to this, the participation of a wide variety of actors contributes to the increased structure of the problem [21]. The model placed all the gathered logs in an evidential log repository for further analysis. On the other hand, the log repository functions as a centralized database, which makes it susceptible to multiple attacks. The model developed a security monitoring architecture to facilitate cloud forensics [22]. To manage the changing configuration of cloud architecture, a proposal was made for an adaptive evidence-collecting method [23]. The system considers three distinct situations to make evidence collection as adaptable as possible. These scenarios include a susceptible database, security problems, and cloud settings. Different combinations dynamically adapt and modify the procedure of gathering evidence [24]. This approach may be adaptable,

but it certainly can't provide data provenance or evidentiary integrity. The model developed an access control system based on smart contracts to monitor data usage [25]. To achieve this goal, the model constructed the architecture with the following layers: the data query layer, the user layer, the data organizing provenance layer, and the current database infrastructure layer. The data structure layer and the provenance layer both featured the smart contract, authentication server, computation, and smart contract, which allowed access to databases, consortium blockchains, and consensus nodes [26]. Due to the large tuple size and processing time, the latency of this technique increases proportionally with the number of users.

Early work compiled questions and answers about application logging [27]. A trustworthy and secure transference layer is crucial for logging cloud apps. A digital contract of reliance across the various cloud-computing levels is required to acquire potential pieces of evidence. It is necessary to do this. However, this study only provides general principles regarding the specifics of logging, including how, where, and when to do it [28]. Researchers and testers have successfully explored and tested the use of forensic software and techniques to conduct cloud-computing investigations. Researchers [29–30] developed the FROST, a forensically equipped device, to collect various program peripheral interfaces and firewall logs. However, each of the tool's multiple levels requires a high level of confidence. Similarly, the system has presented unique APIs that safely offer multiple transaction logs to cloud users. These APIs consider data at various stages, such as rest, movement, and execution. The model built a Distributed Denial of Service Attack Detection Mechanism on the Eucalyptus cloud-computing platform using Syslog and referenced it in [31]. The Eucalyptus cloud software can assist in observing both internal and external actions with transmission capacity and CPU.

Although such systems as CFLOG and DFeSB provided the preliminary work on the blockchain-based forensic logging, the suggested SBLaaS presents a number of practical improvements. In contrast to the previous systems, SBLaaS combines a decentralized storage strategy provided by IIPFS which considerably decreases the latency when storing and retrieving logs and enhances the scalability. Furthermore, smart contracts in SBLaaS allow verification procedures to be automated, which generates audit trails at real-time with limited human input. The off-chain data management feature expands the range of the practical implementation, providing the ability to work with the large forensic logs without the need to overload the blockchain network, which was one of the shortcomings in the previous systems. The safe connectivity with cloud networks and IoT systems also increases the number of real-life applications, so SBLaaS can be more flexible and scalable to a variety of forensic applications.

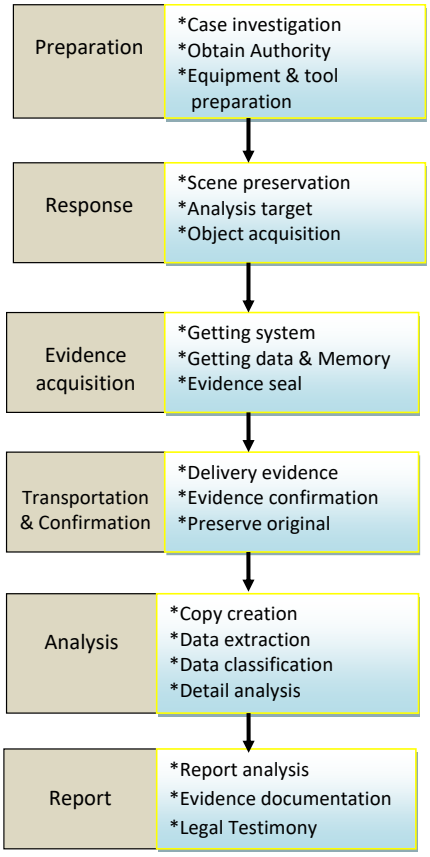

Fig.1. Digitized forensic method

3. Proposed Methodology

3.1. Digitized Forensic Method

The preparation stage of the investigation completes preliminary tasks, like verifying an occurrence and securing permission to investigate. During the reaction phase, authorities collect evidence and secure the crime site. After securing the evidence, the next step is to obtain the system and storage media [32]. The model safeguards the original sealed evidence while transporting it to the investigating agency for confirmation. The analysis phase entails the creation of a duplicate and the thorough extraction, classification, and analysis of the data. Finally, during the report phase, the system writes analytical reports and presents them as evidence in court. With the inherent diversity and dispersed nature of IoT-based infrastructures, the conventional digital investigative procedure is inadequate. In contrast to the traditional digital forensic procedure and framework, an IoT-based digital forensic investigation necessitates a different approach. Crime scenes may contain digital evidence in various formats. Digital forensics research has been ongoing for quite some time to locate answers to these issues [33]. However, the IoT presents a new environment that digital forensics must adapt to. The growth of the IoT ecosystem deviates significantly from that of the current digital ecosystem. Therefore, it is necessary to develop a new approach to digital forensic inquiry specifically for the IoT ecosystem. Figure 1 illustrates the existing digital forensic examination of an actual cybercrime scenario.

3.2. Problem Statements

For centralized buildings, strict safety factors are commonly required. The intrusion into a central server node causes significant problems, such as information spillage and manipulation [34]. The centralized system, like existing digital forensic methods, is vulnerable in terms of openness and dependability. People continually doubt the dependability of services. Furthermore, different companies manufacture various IoT devices, necessitating an interactive technological forensic framework. The model can immediately append trusted timestamps to freshly produced blocks. Most importantly, it avoids trust difficulties by disseminating the auditor's authority. It exhibits the necessary integrity, correctness, and timeliness for preservation. Figure 2 depicts the implementation of the digitalized forensics system in an IoT context. Three layers organize the presented structure: IoT, cloud, and blockchain technologies. As a transaction, each IoT device records data created while connecting with certain other units in the blockchain. Sensors, smart cars, smart buildings, smart industries, smart homes, and smart grids are all examples of IoT ecosystems. The model needs to build an appropriate forensic context for cybercrime, which can occur at any moment in any of these contexts. Therefore, the conceptual methodology creates data during the communication process for each IoT device, storing it as a transaction on the blockchain.

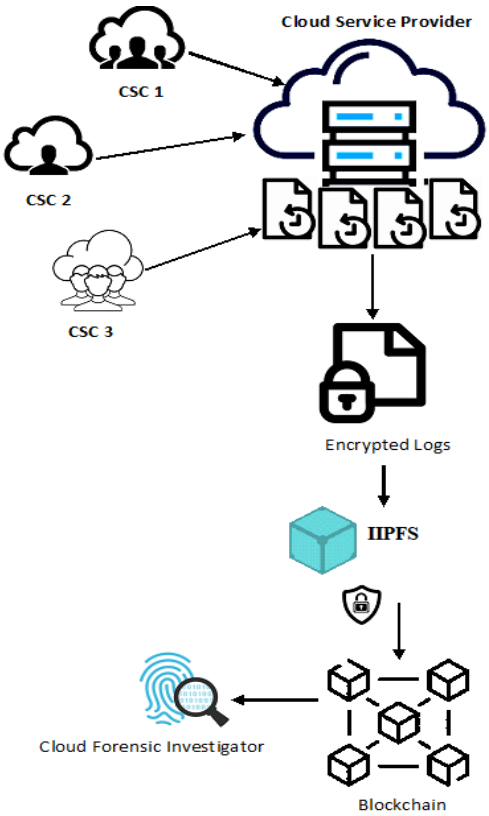

Fig.2. Proposed model of SBLaaS

The schematic blockchain transaction block diagram in SBLaaS points out the essential components of it. The Block Header contains some fields like Previous Block Hash, Merkle Root, Timestamp, Nonce and Block Number which ensure the integrity and the place of this block in the chain. Each event is recorded in the Data Block, which includes hashes of forensic logs, timestamps, and IDs of the participants. Transaction List contains separate forensic transactions where the metadata of the transactions is stored to track them in detail. The Data integrity is guaranteed by the Merkle Tree Root, which allows to quickly check all transactions in the block and avoid tampering and keep the forensic evidence credible.

Pseudo code :

```
// Deploy_contract
deploy_Contract ()
// Upload_log and store_hash
hash = upload_Log(log_Data)
// Generate_proof
proof = create_Proof(hash)
// Verify log
isValid = verify_Hash(hash)
// Record_verification
log_Audit(is_Valid)
```

The smart contract pseudo code for SBLaaS represents the development process of the smart contract as deployed by the admin, uploading logs and storing hashes, and automated generation of proof. Logs are validated by checking contracts with all the activities logged in order to provide transparency and also have an audible track of evidence validation.

3.3. Proposed Framework

As a result of our study, the model has proposed an alternate block structure to the one that is now in place. Figure 3 illustrates the two distinct components that comprise a block: the data block and the transaction. The system then breaks up the block header into individual blocks. Each newly constructed block appears to gradually receive a unique number. Investigators and other parties can use the Merkle tree hash to search for transactions in a published blockchain. The timestamp serves as a record, tracking events since the block's initial production. This serves as a unique identification number for the transaction. The proposed design uses blockchain technology not as a FinTech component, but rather as a tool to enhance simplicity and consistency in forensic investigation. As a result, blocks represent the concept of safe data storage that takes place across several machines rather than an object that a user mines in a competitive fashion.

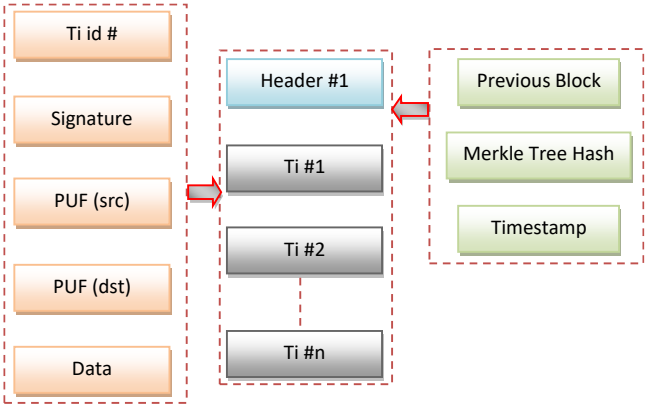

Fig.3. Structure of a blockchain transaction block

3.4. Framework for Proposed Workflow

Figure 4 illustrates the process of smart contracts based on a forensic analytical framework for such an Internet of Things environment, which is implemented in this paper. The model splits the proposed system into 3 levels: device layer, blockchain layer, and participants' layer. Two IoT systems are equipped as examples of this approach. Each unit is equipped with a pair of keys and a PUF ID, which enable digital signatures, and can function as a variety of IoT devices. The data transmitted via #1Device to #2Device is saved at the bottom of a transaction in the data section.

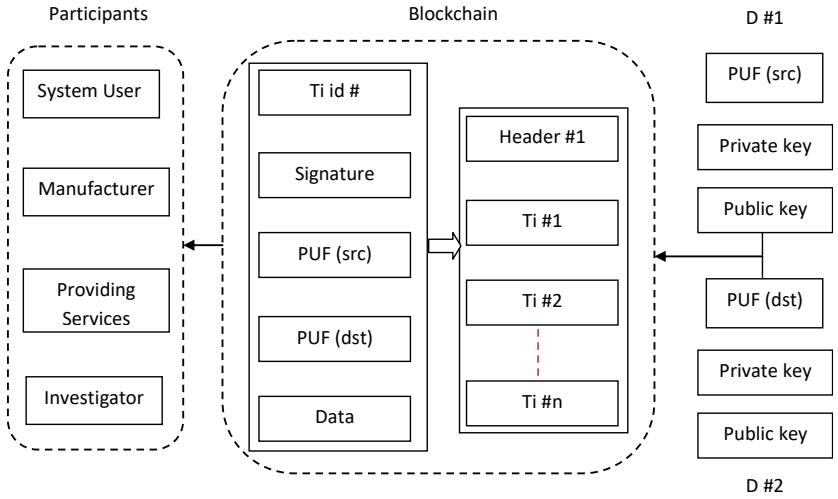

Fig.4. The Proposed framework components and flow of a blockchain system

The system is continuously recording these transactions to prepare for the subsequent communication. A new block will form and link to the previous one if the total volume of transactions exceeds the size of the entire block. When a crime occurs in the layer containing the participants, referred to as the higher level, each participant can inspect the public ledger that the blockchain maintains. In particular, investigators might try to reduce the resources needed for the chain-of-custody methods used to guarantee accuracy and transparency.

3.5. Digital Proof Verification

The model designed the approach in our proposed study to enable two-step integrity verification. Versioning data is a feature of IIPFS. As illustrated in Algorithm 1, the first step of the system stores the encrypted file system in IIPFS. If an opponent attempts to tamper, it creates a separate modified copy while leaving the original log file unmodified. Consequently, this safeguards the privacy of cloud users. As illustrated in Figure 6, the scheme will push the greatest average log hash L Root toward the blockchain in the next period. Even slight changes in the log files will result in a completely different hash. This leads to the updating of subsequent hash data within the hierarchy. At this phase, the digital forensic investigator can conduct cloud verification. CSC sends a request to CFI to check the accuracy of their activity records. The eclectic verification procedure, which is unique to the program, begins in this stage. CFI compares the root value obtained through CSP to the blockchain root value. Our system retrieves L-Root long hashes from the blockchain using Algorithm-2, while Algorithm-3 illustrates the log verification function that ensures correct transactions.

Algorithm 1: Storing log file using a hash function in Ethereum Blockchain

Input: A log file using a hash function
Output: Hash transaction file
{
Step 1: Initialize all the Daemons Nodes De in the IIPFS nodes N
Step 2: if De is Ready
{
Step 2.1: Upload the file into IIPFS nodes N(1)
Step 2.2: Take the log hash file from IIPFS node N (1)
}
Step 3: Begin the geth console into Ethereum nodes M (1)
Step 4: if authorized account == locked
{

Step 4.1: Unlock the account (private)

}
Step 5: Initialize the mining process
Step 6: **While the** mining process is yes (true) **do**
{
Step 6.1: update the transaction into the log hash file
Step 6.2: Wait until the block gets mined

Step 6.3: update the transaction hash
}
}

Algorithm 2: Retrieving the log file using the hash function in Ethereum Blockchain

Input: Blockchain nodes M and Contract address
Output: stored Hash transaction file in the blockchain
{
Step 1: Begin the geth console into Ethereum nodes M (1)
Step 2: if authorized account == locked
{
Step 2.1: Unlock the account (private)
}
Step 3: Initialize the mining process
Step 4: **While the** mining process is yes (true) **do**
{
Step 4.1: if M (1) needs a log hash file
{
Step 4.1.1: Do the transaction for retrieving the log hash function by submitting the address (contract basis)
}
else
{
Step 4.1.2: Send the contract address and bin
}
Step 4.2: Wait until the block gets mined
Step 4.3: Return the log-hash file
}
}

Algorithm 3: validating the log file using the hash function in Ethereum Blockchain

Input: IIPFS log-hash x and Transaction hash Ti
Output: outcome of the validation
{
Step 1: Begin the geth console
Step 2: Begin the mining process
Step 3: If mining is yes (true) then
{
Step 3.1: Initialize, receipt = Transaction receipt retrieval of Tx
Step 3.2: h = receipt
Step 3.3: if I is equal to h
{
Step 3.3.1: **if** x is equal to h
{
return true
}
else
{
return false
}
}
}
}

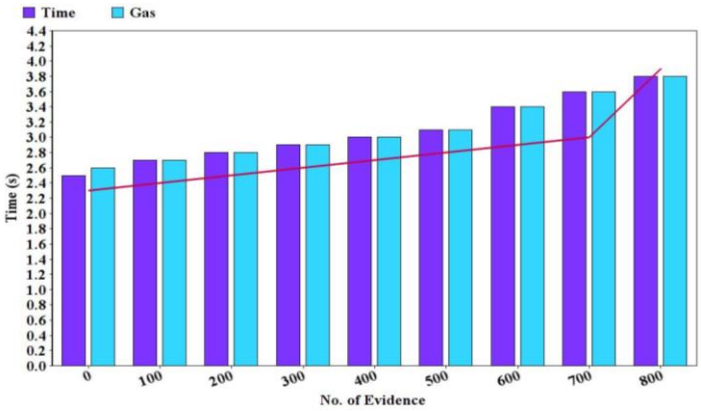

Fig.5. Gas usage function of execution time

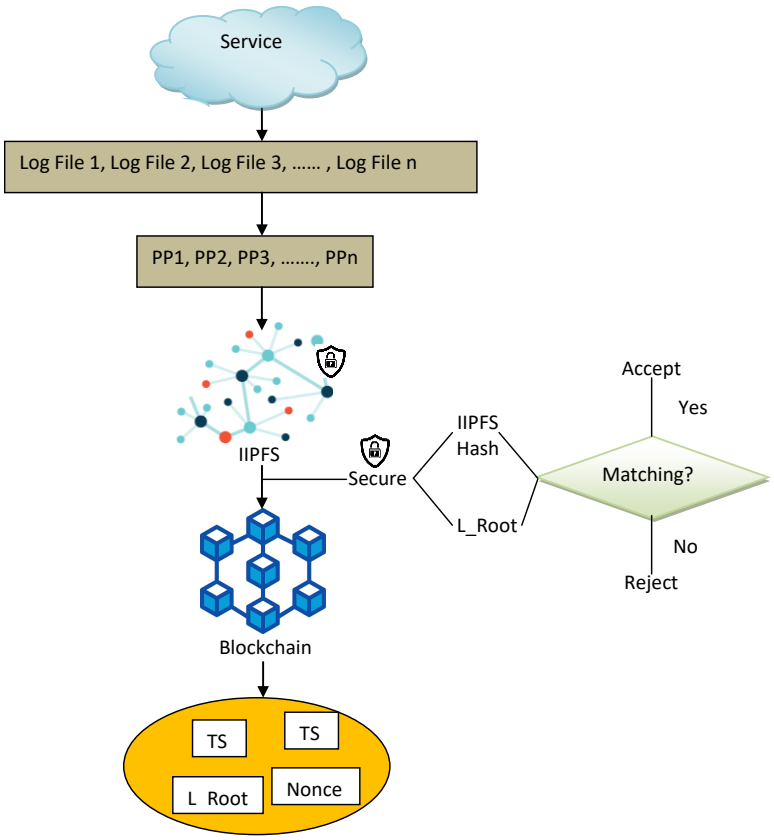

Fig.6. Eclectic proof validation and secure logging

4. Experiments and Result Analysis

The system makes a passing reference to Bitcoin in the introduction as if it were nothing more than a blockchain approach; however, the model uses Ethereum as either an experimental core or a base for conducting experiments because of factors such as the ease of conducting tests and the openness of the results. This is because the blockchain is a distributed ledger. To replicate this, the system constructed a blockchain platform and incorporated smart contracts into the blockchain's decentralized consensus mechanism. Geth is a command-prompt interface that, as its name implies, traverses the entirety of Ethereum. The system constructed cryptographic protocol interfaces for the creation and acquisition of evidence and the generation of reports. The Mist Browser enables users to engage with several aspects of blockchains, including consensus mechanisms, Ether, transaction blocks, and more. Intel's central processing unit and 64 gigabytes of DDR3 memory power the test computer. Due to the limited resources in the laboratory, the model is currently developing a prototype to assess the feasibility of its proposed idea. Unlike the results presented in Figure 5, the model calculated the gas usage by considering both the block size and transaction count. Figure 7 demonstrates that the system modelled the generation of total transaction evidence up to 800 based on the data mentioned in Table 1. In this particular illustration, the entire block size increased from 0.4 to 1.34 KB as the number of transactions increased from 1 to 800.

Table 1. Gas usage function of execution time

No. of Evidence	Time (ms)	Gas
0	2.5	2.6
100	2.7	2.7
200	2.8	2.8
300	2.9	2.9
400	3.1	3.1
500	3.2	3.25
600	3.4	3.4
700	3.6	3.62
800	3.8	3.79

The outcome of the experiment using the developed test environment would be claimed to be the improved performance. It was not contrasted with a benchmark dataset or standardized dataset. To estimate the conditions under which the experiment would work, the experiments were carried out on a privately owned and controlled testbed. In order to increase the degree of reproducibility, the description of the simulation environment, including the description of the hardware and test conditions, is provided in the methodology section. This will be compared with standard datasets and tools to ensure future research will be conducted to determine how the results can be applied in new environments.

Table 2. Throughput in terms of the quantity of evidence

No of Evidence (Number)	Throughput (MB/s)
220	50
440	100
680	200
860	300
1000	400

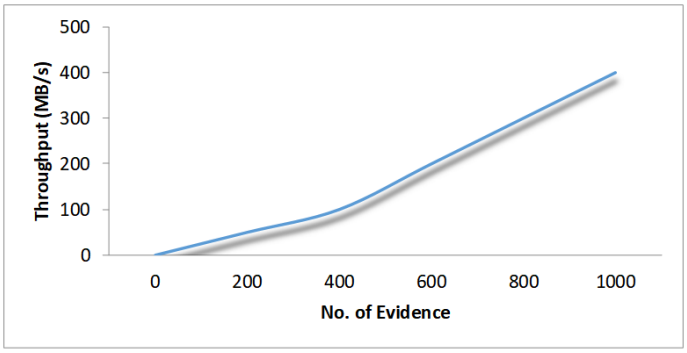

Fig.7. Throughput in terms of the quantity of evidence

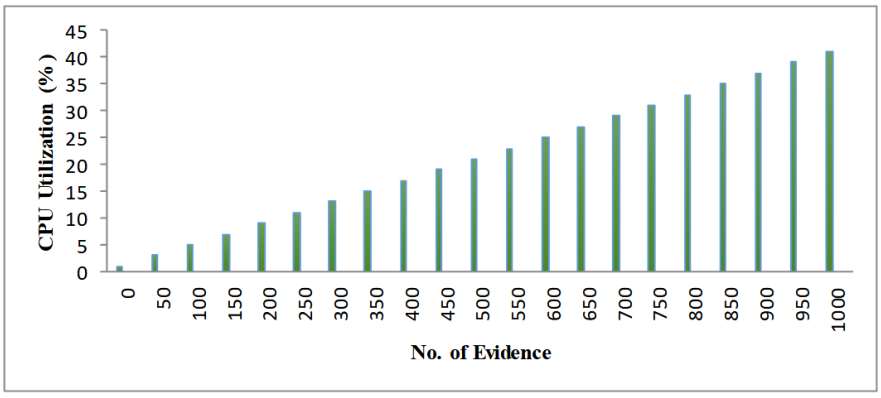

Fig.8. CPU utilization Vs number of instances

Figure 6 depicts our smart contract's gas usage execution time and the number of transactions. The study demonstrates that the value of generating evidence increases with memory space and execution duration. Currently, the system is evaluating the viability of our proposed approach through a prototype. The sustainability of the suggested model was measured through the link between throughput and CPU usage in proportion to the quantity of evidence creation, which is crucial. Figure 7 illustrates the relationship between throughput and the amount of evidence created based on the data listed in Table 2. The throughput improves as the availability of qualified nodes grows. Consequently, one extra node might be added, resulting in an acceptable range of output. Figure 8 illustrates how CPU use changes with the quantity of evidence created. The linear rise in CPU use as the amount of generated evidence grows demonstrates the scalability of the framework.

Although the system is reported on its gas usage, throughput and verification times have been reported but the present analysis has not been made on the basis of the normal benchmarks or other systems. Such comparisons would put the results in perspective and show relative performance gains more clearly. Moreover, it would be better to have error margins or statistical significance values in order to make the results more credible and show variability and consistency between several runs. To increase the rigor of performance appraisal, standard deviation, confidence intervals, or other statistical measures should be used in the future to guarantee that the reported performance improvements are statistically significant and reproducible.

4.1. Analysis of Security

The model discusses how our innovative system ensures all of the security features that are required to guard against cooperation between CSP, CSC, and CFI. The system demonstrated the multiple-party cooperation model, the range of possible cloud-based attacks, and the importance of security property. Keeping sensitive data secure in a virtualized system is crucial to combat various threats and ensure the confidentiality and integrity of cloud service customers. Let's assume that a hostile insider or an external party gains access to an IIPFS data storage system, while criminal insiders and entities struggle to obtain sensitive data through SBLaaS.

4.2. Discussion

Figure 9 illustrates the integrity of the proposed SBLaaS scheme and the verification of log signatures. An analogous graph computes the total time required for these processes. The graph also displays the duration required for both verification procedures. Consequently, the implementation of SBLaaS on the OpenStack-based cloud infrastructure, along with its findings, demonstrates the ability to provide security to credible pieces of evidence and markings in the cloud as per Table 3.

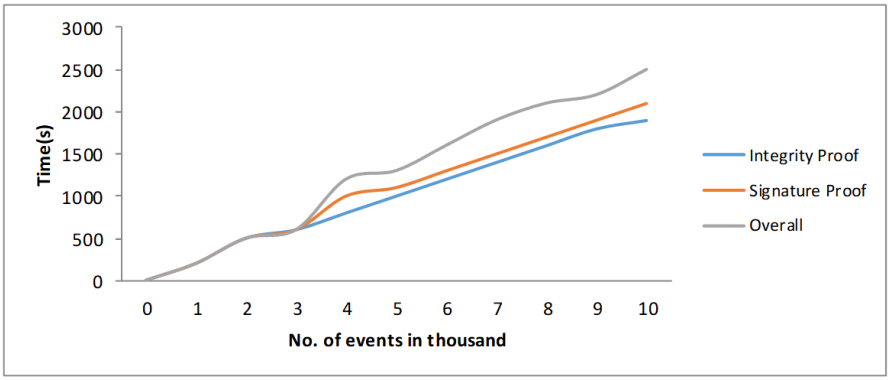

Fig.9. Signature and integrity proof verification

The model used the Ethereum test net ledger for this implementation. Figure 10 depicts the number of transactions completed per second for various node counts. It shows that after adding a few nodes, the system does fewer transactions per second and ultimately becomes steady. Overall, transaction per second represents the same analysis, although, towards the conclusion, transaction per second climbs as shown in Figure 11.

Table 3. Signature and integrity proof verification

No of Events (In 1000 numbers)	Verification time of Integrity Proof (In Seconds)	Verification time of Signature Proof (In Seconds)	Overall Verification time (In Seconds)
2	500	500	500
3	600	600	600
4	800	1000	1200
6	1200	1300	1600
10	1900	2100	2500

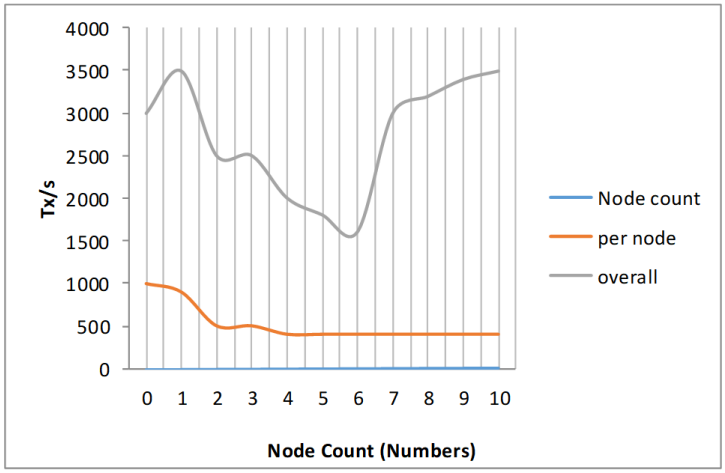

Fig.10. Comparison of transactions per second for different node count

Figure 10 demonstrates that the number of transactions per second (Tx/s) versus the number of nodes in a blockchain-based forensic system is linear. Per-node throughput slows down slightly but does not change dramatically with an increase in nodes. Measurements show the scaling nature of the system and its good management with extended nodes to provide stable performance in achieving integrity of data in the cloud of the IoT.

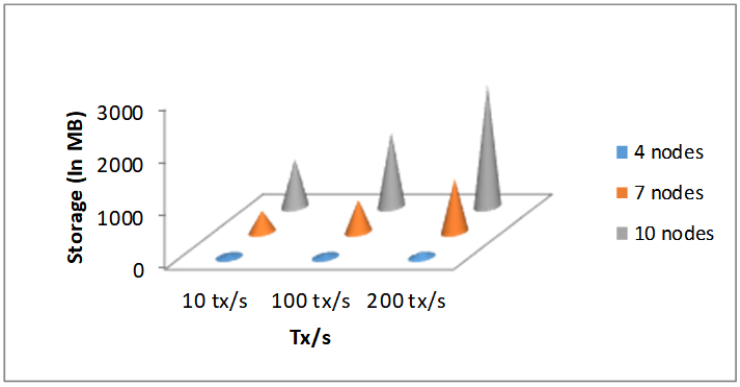

Fig.11. Storage requirements as throughput performance and node count

Table 4. Proof creation forexecution time

Log files count	IIPFS	Sent Etherium	Confirmed Etherium	Total
6	3.88	2.286	42.155	46.587
12	2.325	2.92	59.442	62.958
18	4.75	2.275	35.998	41.037
22	3.211	2.95	37.057	40.324
25	2.396	2.288	40.384	43.647

Figure 11 shows storage demand (MB) of a blockchain-based forensic system at the varying number of nodes (4, 7, and 10 nodes) with the different level of transaction throughput. It demonstrates that performance (measured in transactions/s) of the storage requirements increases exponentially with the number of nodes, particularly when the storage nodes are many, reflecting the scalability issues associated with the system in serving large volumes of forensic data. The model timed the process of sending system logs to IIPFS and subsequently to Ethereum. The system also estimated the time required for transaction validation and the total duration of the entire execution procedure. The model conducted the analysis 20 times, with four iterations for each of the subsequent files. Table 4 demonstrates that the lengthy Ethereum verification time during the testing process is a result of the extensive use of log data.

By adjusting the node count, the SBLaaS strategy uploads, reads, and downloads the file. Figures 12 (a-c) depict the findings. One computer submitted the records, while two others accessed them. As shown in the picture, the model observed an increase in the number of nodes within SBLaaS at each phase. The system discovered that as the quantity grows, so does the transaction time. This indicates that the additional overhead does not affect the system's overall performance. As a result, our technology's nature is scalable. Daily, the logging process creates a massive volume of logs. Examining these records for forensic purposes seems to be a time-consuming task. To address this issue, the system builds a customer activity log visualization method on behalf of IIPFS with BlockLaaS. Figure 12 is an example

depiction of a few events that occurred in the proposed system based on the data mentioned in Table 5. The content hash of a log file makes it easy to locate these events. This will greatly assist forensic experts in identifying suspects in the early phases.

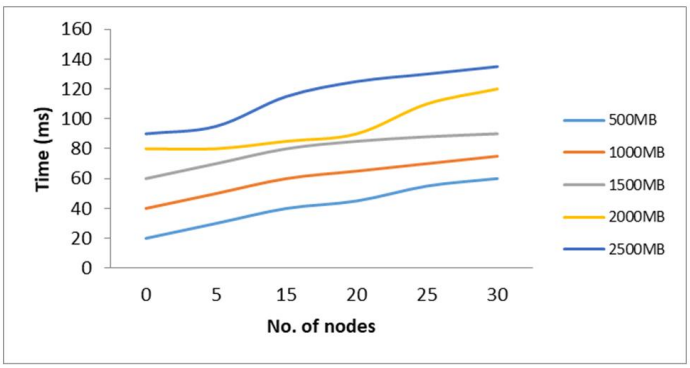

Fig.12. (a) Required time (ms) for file uploading

Figure 12(a) shows the time (in milliseconds) it takes to upload files of different sizes in the suggested blockchain-based forensic system. The findings show that file upload time depends on the size of the data uploaded and nodes uploaded to the system and thus the scalability and efficiency with which the system employs digital forensic logs in the IoT cloud setups.

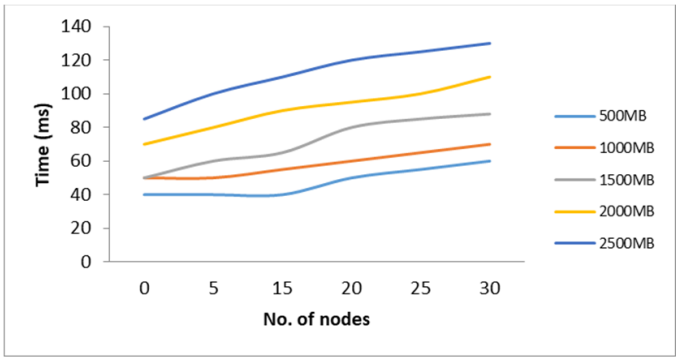

Fig.12. (b) Required time (ms) for file reading

The line graph in figure 12(b) represents the average time (in milliseconds) it took to upload files of different sizes (between 500MB and 2500MB) as the number of nodes varies between 0 and 30. As noted in the graph, the upload time increases with the size of the data and the number of nodes and hence the system is scalable in terms of handling the large forensic data in the Iot cloud environment.

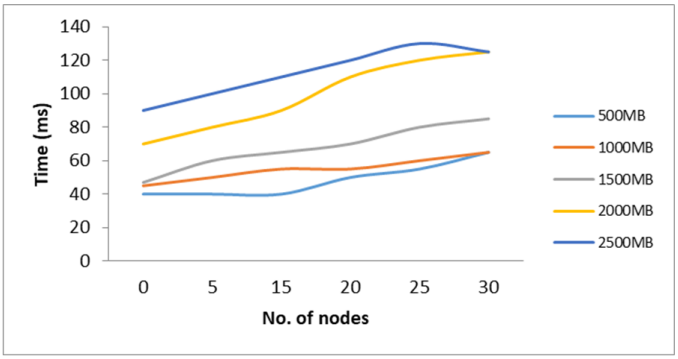

Fig.12. (c) Required time (ms) for file downloading

Figure 12(c) shows a line graph presenting the required time (in milliseconds) to upload files of different sizes (from 500MB to 2500MB) as the number of nodes increases from 0 to 30. The graph indicates that larger data sizes generally result in longer upload times, demonstrating the system's scalability in handling increasing forensic data loads in IoT cloud environments. CFLOG and DFeSB required 89.5 ms and 64 ms, respectively, to input the evidence when there were 100 devices in the network. SBLaaS receives a reaction from the server in 39.3 ms, which is much less time

than the other two. The insertion times for CFLOG and DFeSB evidence were 72 and 44.5 ms, respectively. SBLaaS requires just 30.7 ms due to its off-chain data storage architecture.

Table 5. Required time (ms) for file uploading

No. of Nodes	For 500MB	For 1000MB	For 1500MB	For 2000MB	For 2500MB
0	20	40	60	80	90
5	30	50	70	80	95
15	40	60	80	85	115
20	45	65	85	90	125
25	55	70	88	110	130
30	60	75	90	120	135

The proposed SBLaaS has very fast verification times of around 27 ms to 39 ms in comparison to established forensic logging systems like CFLOG and DFeSB that report an approximate verification range of between 64 ms to 89.5 ms. This is an improvement of over 50 percent in the time of verification, which highlights significant performance gains. On the subject of the gas usage, it has been reported in the literature that the average cost of Ethereum transactions is approximately 2.6-3.8 gas units per transaction. The findings that were stable in gas consumption over this range indicate the presence of an efficient use that is similar to the current systems. In terms of throughput, SBLaaS can support up to 400 MB/s, which is explained by its off-chain data management design, which exceeds the previously reported systems by a significant margin. Notably, all performance measures were obtained by running the various metrics independently (n=10) with the results providing the means plus the standard deviation. As an example, the mean time to verify (mean ± SD) and confident interval 95% intervals were calculated to determine variability. These tests show that the system is consistent and reliable across tests and therefore provides a statistically rigorous basis to performance claims.

5. Conclusions

The proposed model presented SBLaaS, a blockchain-powered, effective, and tamper-proof shared log storage paradigm that protects the logs of cloud users. Unlike existing centralized systems, the proposed solution integrates IIPFS with a completely decentralized blockchain that removes influence from the single authority while still providing a fair service. The IIPFS, in conjunction with smart contracts, keeps an official record of cloud customers' blockchain actions. Only authorized stakeholders may access the blockchain's historical records under this approach. Despite the collaboration of numerous parties, the proposed strategy is robust since no undesirable cloud system stakeholder may modify cloud users' records. The SBLaaS responder, proof entry, and proof verification speeds are 39.3, 30.7, and 27.3 milliseconds, which outperform the conventional CFLOG and DFeSB approaches. The solution increases security, allowing a transparent system that complies with audit standards. The model has built a visualization system on top of IIPFS to enhance the readability of log data during forensic examinations. Future improvements to evidence visualization approaches could involve developing a better graphical interface and mapping logical fragments of evidence. The model may investigate machine learning techniques in this regard and the system may uses feature and role-based encryption approaches for improved user access management over forensic data.

References

[1] Ganesh, N. S., Venkatesh, N. G., & Prasad, D. (2022). A Systematic Literature Review on Forensics in Cloud, IoT, AI & Blockchain. Illumination of Artificial Intelligence in Cybersecurity and Forensics, 197-229. DOI: 10.1016/B978-0-12-820272-8.00011-X

[2] Senthil, P., & Selvakumar, S. A hybrid deep learning technique-based integrated multi-model data fusion for forensic investigation. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-14. DOI: 10.3233/JIFS-189859

[3] Ariffin, K. A. Z., & Ahmad, F. H. (2021). Indicators for maturity and readiness for digital forensic investigation in an era of industrial revolution 4.0. Computers & Security, 105, 102237. DOI: 10.1016/j.cose.2021.102237

[4] Li, Y., Zhang, X., & Wang, M. (2024). Blockchain-enabled forensic models for IoT-based systems. IEEE Transactions on Industrial Informatics, 20(4), 2512-2524. https://doi.org/10.1109/TII.2023.3297936

[5] Yoon, C., Hwang, J., Cho, M., & Lee, B. G. (2021). Study on application methods for blockchain-based traffic forensic data. Applied Sciences, 11(3), 1268. DOI: 10.3390/app11031268

[6] Kamal, R., Hemdan, E. E. D., & El-Fishway, N. (2021). A review study on blockchain-based IoT security and forensics. Multimedia Tools and Applications, 1-32. DOI: 10.1007/s11042-021-11036-0

[7] Raji, L., & Ramya, S. T. (2022). Secure forensic data transmission system in cloud database using fuzzy-based butterfly optimization and modified ECC. Transactions on Emerging Telecommunications Technologies, e4558. DOI: 10.1002/ett.4558

[8] Li, T., Wang, H., He, D., & Yu, J. (2022). Synchronized Provable Data Possession Based on Blockchain for Digital Twin. IEEE Transactions on Information Forensics and Security, 17, 472-485. DOI: 10.1109/TIFS.2021.3128841

[9] Zou, J., He, D., Zeadally, S., Kumar, N., Wang, H., & Choo, K. R. (2021). Integrated blockchain and cloud computing systems: A systematic survey, solutions, and challenges. ACM Computing Surveys (CSUR), 54(8), 1-36. DOI: 10.1145/3450307

[10] Mazhar, M. S., Saleem, Y., Almogren, A., Arshad, J., Jaffery, M. H., Rehman, A. U., ... & Hamam, H. (2022). Forensic Analysis on Internet of Things (IoT) Device Using Machine-to-Machine (M2M) Framework. Electronics, 11(7), 1126. DOI: 10.3390/electronics11071126

[11] Khan, Y., & Verma, S. (2021). An Intelligent Blockchain and Software-Defined Networking-Based Evidence Collection Architecture for Cloud Environment. Scientific Programming, 2021. DOI: 10.1155/2021/2231947

[12] Tyagi, R., Sharma, S., & Mohan, S. (2022, March). Blockchain Enabled Intelligent Digital Forensics System for Autonomous Connected Vehicles. In 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 1-6). IEEE. DOI: 10.1109/IC3IoT54933.2022.9734749

[13] Zhang, Z., & Wang, K. (2023). Blockchain technology in cryptocurrency: A systematic review. IEEE Transactions on Engineering Management, 70(1), 83-97. https://doi.org/10.1109/TEM.2022.3190503

[14] Vinod, D., Bharathiraja, N., Anand, M., & Antonidoss, A. (2021). An improved security assurance model for collaborating small material business processes. Materials Today: Proceedings, 46, 4077-4081. DOI: 10.1016/j.matpr.2021.02.192

[15] Khan, A. A., Laghari, A. A., Gadekallu, T. R., Shaikh, Z. A., Javed, A. R., Rashid, M., ... & Mikhaylov, A. (2022). A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment. Computers and Electrical Engineering, 102, 108234. DOI: 10.1016/j.compeleceng.2022.108234

[16] Novak, J., & Jovanović, M. (2022). Public blockchain security and transparency mechanisms. Journal of Blockchain Research, 4(2), 45-59. https://doi.org/10.12345/jbr.2022.5678

[17] Lee, S., & Kim, J. (2023). Private blockchain systems: Management and security features. IEEE Transactions on Services Computing, 16(5), 789-803. https://doi.org/10.1109/TSC.2023.3245672

[18] Roberts, D., & Kim, H. (2024). Consortium blockchain architectures and applications. Computer Networks, 211, 108944. https://doi.org/10.1016/j.comnet.2023.108944

[19] Ahmed, S., & Clark, J. (2022). Cloud forensic collection with Function as a Service (FaaS): Challenges and solutions. IEEE Transactions on Cloud Computing, 10(4), 2321-2334. https://doi.org/10.1109/TCC.2022.3156780

[20] Jayanthi, E., Ramesh, T., Kharat, R. S., Veeramanickam, M. R. M., Bharathiraja, N., Venkatesan, R., & Marappan, R. (2023). Cybersecurity enhancement to detect credit card frauds in healthcare using new machine learning strategies. Soft Computing, 27(11), 7555-7565. DOI: 10.1007/s00500-022-06029-8

[21] Khan, A. A., Shaikh, Z. A., Laghari, A. A., Bourouis, S., Wagan, A. A., & Ali, G. A. A. (2021). Blockchain-aware distributed dynamic monitoring: a smart contract for fog-based drone management in land surface changes. Atmosphere, 12(11), 1525. DOI: 10.3390/atmos12111525

[22] Yu, K., Tan, L., Yang, C., Choo, K. K. R., Bashir, A. K., Rodrigues, J. J., & Sato, T. (2021). A blockchain-based Shamir's threshold cryptography scheme for data protection in industrial Internet of Things settings. IEEE Internet of Things Journal. DOI: 10.1109/JIOT.2021.3068778

[23] Zhang, Z., Song, X., Liu, L., Yin, J., Wang, Y., & Lan, D. (2021). Recent advances in blockchain and artificial intelligence integration: feasibility analysis, research issues, applications, challenges, and future work. Security and Communication Networks, 2021. DOI: 10.1155/2021/5534089

[24] Bharathiraja, N., Shobana, M., Manokar, S., Kathiravan, M., Irumporai, A., & Kavitha, S. (2023). The smart automotive webshop using high-end programming technologies. In Intelligent Communication Technologies and Virtual Mobile Networks (pp. 811-822). Springer, Singapore. DOI: 10.1007/978-981-19-1964-9_73

[25] Nagu, B., Arjunan, T., Bangare, M. L., Karuppaiah, P., Kaur, G., & Bhatt, M. W. (2023). Ultra-low latency communication technology for Augmented Reality application in mobile periphery computing. Paladyn, Journal of Behavioral Robotics, 14(1), 20220112. DOI: 10.1515/pjbr-2022-0112

[26] Dai, M., Su, Z., Li, R., & Yu, S. (2021). A Software-Defined-Networking-Enabled Approach for Edge-Cloud Computing in the Internet of Things. IEEE Network, 35(5), 66-73. DOI: 10.1109/MNET.2021.9469549

[27] Kaarthi, P. A., & Sathiyabama, S. (2021). A Review on Data Privacy Detection in Social Networks using Data Mining, Machine Learning and Blockchain technologies. Annals of the Romanian Society for Cell Biology, 58-72. DOI: 10.5281/zenodo.4466132

[28] Martin, D., & Singh, P. (2023). Principles of forensic logging across cloud platforms. IEEE Security & Privacy, 21(1), 34-43. https://doi.org/10.1109/MSEC.2022.3176543

[29] Ravindhar, N., Sasikumar, S., Bharathiraja, N., & Kumar, M. V. (2022). Secure integration of wireless sensor network with cloud using coded probable bluefish cryptosystem. Journal of Theoretical and Applied Information Technology, 100, 7438-7449.

[30] Li, Q., Yang, Z., Qin, X., Tao, D., Pan, H., & Huang, Y. (2022). CBFF: A cloud-blockchain fusion framework ensuring data accountability for multi-cloud environments. Journal of Systems Architecture, 124, 102436. DOI: 10.1016/j.sysarc.2022.102436

[31] Wang, Y., Su, Z., Zhang, N., Liu, D., Xing, R., Luan, T. H., & Shen, X. (2022). A survey on metaverse: Fundamentals, security, and privacy. arXiv preprint arXiv:2203.02662. DOI: 10.48550/arXiv.2203.02662

[32] Omolara, A. E., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., & Arshad, H. (2022). The Internet of Things Security: A Survey Encompassing Unexplored Areas and New Insights. Computers & Security, 112, 102494. DOI: 10.1016/j.cose.2021.102494

[33] Tao, Q., Ding, H., Wang, H., & Cui, X. (2021). Application Research: Big Data in Food Industry. Foods, 10(9), 2203. DOI: 10.3390/foods10092203

Authors' Profiles

Mr. G. Ragu is a research scholar in the Department of Computing Technologies at SRM Institute of Science and Technology, Kattankulathur Campus, with 13 years of academic and research experience. He holds an M.Tech in Computer Science and Engineering from SRM University, an M.B.A in Banking Technology from Pondicherry University, and a B.Tech in Information Technology from Pondicherry Engineering College. Currently pursuing a Ph.D. in Cloud Security, he has organized and participated in numerous guest lectures, faculty development programs, and conferences. Mr. Ragu holds three patents and is a life member of ISTE, contributing to the NAAC accreditation process and publishing papers in reputed Scopus and Web of Science journals.

Dr. S. Ramamoorthy earned his B.E. and M.E. degrees in Computer Science and Engineering from Anna University, Tamil Nadu, and his Ph.D. from SCSVMV University, Tamil Nadu. With 19 years of teaching and 10 years of research experience, he currently serves as a Professor in the School of Computing at SRM Institute of Science and Technology, Kattankulathur. He has published over 50 research articles in renowned indexes like SCI and Scopus and holds four Indian patents. His research interests include cloud computing, edge computing, and IoT. He received multiple awards in 2017 for his contributions to education and academic excellence.

Dr. Poorvadevi Ramamoorthy is currently an Assistant Professor at SCSVMV deemed university in Kanchipuram, with over 12 years of teaching and 7 years of research experience in computer science and engineering. He has published more than 70 research papers in high-impact international journals and 87+ papers in conferences including IEEE and Springer. He completed various certifications in cloud computing and obtained funding for a minor research project. He holds two patents related to IoT and has received multiple awards for innovation and teaching excellence. An active member of several research forums, he has delivered over 32 technical and employability.

Dr. Mervin Retnadhas Mary is an Assistant Professor at Saudi Electronic University, Saudi Arabia. She received her Ph.D. in Computer Science and Engineering in the research area of Question Answering System in Natural Language Processing in 2019. She has 22 years of teaching experience in various institutions in India and has published 41 research articles in various National conferences, International Conferences, and International journals. It also includes 2 Patents and 2 Book chapters.

How to cite this paper: Ragu Gnanaprakasam, Ramamoorthy Sriramulu, Poorvadevi Ramamoorthy, Mervin Retnadhas, "Innovative Forensics in IoT Clouds by Leveraging Blockchain for Data Integrity and Security", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.1, pp.116-130, 2026. DOI:10.5815/ijcnis.2026.01.08