International Journal of Recent Trends in Science Technology & Management(IJRTSTM)
©2023 (IJRTSTM) | Volume 4 | Issue 3 | ISSN: 2584-0894
October-December 2025 | DOI: https://doi.org/10.5281/zenodo.18138084

Digital Forensics Framework for Investigating Cloud-Based Cyber Crimes

Yogesh T. Patil[1], Pallavi Soni[2]

[1,2]Assistant Professor, Faculty of Computer Application, Sigma University, Vadodara, India

Yogi007orama@gmail.com[1], pallavi1701@gmail.com[2]

Abstract

The rapid adoption of cloud computing has fundamentally transformed data storage, sharing, and access mechanisms globally, offering unprecedented scalability and flexibility. However, this shift has simultaneously introduced significant new challenges for digital forensic investigators in identifying, collecting, and preserving potential evidence of cybercrimes. Traditional forensic models, which are built upon the premise of local and static data acquisition, are demonstrably unequipped to handle cloud-specific features such as multi-tenancy, dynamic resource allocation, geographic distribution of data, and complex jurisdictional boundaries. This foundational incompatibility often results in delayed investigations, compromised evidence integrity, and substantial challenges to legal admissibility in court, highlighting a critical gap between modern technological infrastructure and current investigative capabilities.

This paper proposes a comprehensive Digital Forensics Framework (CDFF) specifically tailored for modern cloud-based cybercrime investigations, designed to overcome the limitations of conventional approaches. The CDFF is an end-to-end model that integrates structured processes with advanced, cloud-native technologies to ensure evidence integrity, data traceability, and legal admissibility. The framework is built around three core components: an automated evidence collection module leveraging API-based data retrieval from major cloud service providers; a robust, blockchain-based chain-of-custody ledger for the immutable recording of all handling procedures; and forensic log analytics incorporating machine learning to rapidly identify anomalous behavior within massive cloud logging streams. The framework utilizes an integrated, four-phase approach covering Identification & Authorization, Preservation & Acquisition, Analysis & Examination, and Reporting & Presentation.

The proposed CDFF is rigorously evaluated against critical forensic metrics, including time-to-evidence, completeness of data acquired, and the successful defense of the chain-of-custody against simulated tampering attempts. The results of the evaluation conclusively demonstrate that the CDFF significantly reduces investigation time while simultaneously providing a higher degree of evidence immutability and provenance compared to existing, non-integrated models. The successful combination of these components provides a viable, legally-sound blueprint that can be adopted by forensic practitioners to effectively secure and analyze evidence within the complex, distributed architecture of modern cloud infrastructures, thereby advancing the field of cloud digital forensics.

Article Information

Received: 25th October 2025

Acceptance: 28th November 2025

Available Online: 5th January 2025

Keywords: Cloud Forensics Framework (CDFF), Multi-Tenancy Challenges, Cybercrime Investigation Model, Evidence Acquisition and Preservation, Forensic Automation, Scalable Forensic Framework

1. Introduction

Digital forensics traditionally assumes direct, physical access to devices and local storage media. Cloud computing fundamentally alters these foundational assumptions: data and compute resources are abstracted from the underlying physical hardware, user environments are multiplexed on shared, multi-tenant resources, and critical logs and forensic artifacts may be ephemeral or dynamically allocated. This paradigm shift creates significant technical hurdles, including the 'noisy neighbor' problem, where evidence from multiple distinct users is intertwined, and the challenge of establishing a clear temporal link between malicious activity and its associated virtual machine. Consequently, investigators face compounded difficulties obtaining timely, forensically sound evidence while simultaneously preserving the crucial chain-of-custody and ensuring the evidence's legal admissibility across multiple jurisdictions.

The accelerating rise of cloud-enabled cybercrimes — ranging from large-scale, distributed ransomware attacks using cloud infrastructure to sophisticated, covert data exfiltration via compromised cloud storage accounts — makes it imperative to develop and validate robust, cloud-native forensic methodologies. Existing models often rely on cumbersome legal requests and slow manual processes, which fail in environments where evidence can vanish or be overwritten in minutes. The lack of a standardized, automated, and legally defensible process presents a major vulnerability for organizations and impedes effective criminal prosecution.

This research aims to address these critical shortcomings by designing, implementing, and rigorously evaluating a Cloud Forensics Framework (CFF) tailored explicitly for the unique complexities of cloud environments. The CFF is a methodological and architectural solution that addresses these challenges by leveraging cloud-provider APIs for controlled data acquisition, utilizing virtualization features (such as snapshots and images) for stable evidence preservation, integrating centralized log aggregation and analysis for comprehensive event correlation, incorporating cloud-based memory forensics for volatile evidence capture, and implementing an immutable ledger (e.g., blockchain) for a cryptographically secure chain-of-custody. Crucially, the proposed CFF is engineered to be vendor-agnostic, ensuring its applicability across major public cloud providers; extensible to handle complex hybrid and multi-cloud deployments; and practical for direct and timely use by incident response teams and forensic practitioners in real-world investigations. The successful evaluation of this framework will provide a state-of-the-art model for future cloud forensic responses.

2. Problem Statement

Existing forensic models (e.g., DFRWS, NIST SP guides) and conventional tools were meticulously developed for on-premises investigations, relying on the assumption of physical control over devices and static data. Consequently, they do not fully address the cloud's unique characteristics, which introduce critical points of failure in the investigative process:

- Evidence Volatility: Cloud resources, such as virtual machine instances, containers, and transient storage, may be terminated or reallocated quickly due to auto-scaling or maintenance routines. This ephemerality means critical evidence can be lost or overwritten before a warrant is executed or collection can begin.
- Multi-Tenancy: The sharing of underlying physical resources complicates evidence isolation and significantly risks contamination or commingling of unrelated tenants' data, potentially leading to privacy breaches and legal challenges regarding the scope of a forensic examination.
- Limited Physical Access: Investigators typically lack direct, low-level access to physical disks, server memory, or network infrastructure, preventing the use of established disk imaging and memory capture techniques. Access is limited to the Application Programming Interfaces (APIs) provided by the Cloud Service Provider (CSP).
- Distributed Logs and Artifacts: Critical evidence is dispersed across various services, regions, and even multiple providers (in a multi-cloud setup), making comprehensive data collection a logistical and technical nightmare. Correlating these disparate logs into a coherent timeline is exceptionally challenging.
- Chain-of-Custody Challenges: Ensuring tamper-evident proof of evidence handling and provenance across dynamic, automated cloud operations is a non-trivial legal and technical task. Traditional paper-based or even digital-signature models struggle to cope with the sheer volume of resource changes and transfers within a cloud system.

The fundamental consequence of these limitations is that current forensic methodologies lead to unreliable, incomplete, and often inadmissible evidence when applied to cloud-based cybercrimes. The time-to-evidence becomes prohibitively long, often exceeding the lifespan of the transient data. Thus, the overarching problem this research addresses is: How can we design, validate, and implement a forensically sound framework that ensures timely, cryptographically verifiable evidence acquisition and preservation in highly dynamic cloud environments, while rigorously maintaining legal admissibility and operational scalability across various cloud architectures? Addressing this problem is essential for maintaining the rule of law in the modern, cloud-centric digital landscape.

3. Objectives

Phase	Original Objective	Extended and Specific Objective
I. Foundational Analysis	Analyze limitations of current forensic methodologies in cloud environments.	Systematically analyze and document the technical and legal limitations of at least three leading traditional forensic models (e.g., DFRWS, NIST SP 800-89) when applied to multi-tenant, volatile, and distributed cloud computing environments, specifically identifying gaps in evidence capture completeness and time-to-evidence.
II. Framework Design	Propose a cloud-native forensic framework supporting identification, acquisition, preservation, analysis, and reporting.	Design and formally specify a vendor-agnostic, five-stage Cloud Forensics Framework (CFF) that defines new, cloud-native processes for automated identification, API-based acquisition, cryptographic preservation, distributed log correlation, and standardized reporting.
III. Implementation	Implement a prototype utilizing cloud APIs, VM snapshotting, log aggregation, memory analysis, and blockchain-based chain-of-custody.	Develop a working prototype of the CFF using a public cloud platform (e.g., AWS or Azure) to demonstrate core functionalities, specifically implementing: (a) secure data retrieval via CSP APIs, (b) real-time evidence immutability using a blockchain-based chain-of-custody ledger, and (c) a module for volatile memory analysis on virtual

Phase	Original Objective	Extended and Specific Objective
		machines (VMs).
IV. Evaluation	Evaluate the framework across representative incident scenarios and measure acquisition time, integrity preservation, and scalability.	Rigorously evaluate the CFF prototype against at least three distinct cloud-enabled cybercrime scenarios (e.g., data exfiltration, cryptojacking, compromised credentials). The evaluation will quantitatively measure key performance indicators: evidence acquisition time reduction, integrity preservation rate (via hash comparison), and scalability across varying resource loads.
V. Conclusion & Impact	Provide guidelines and best practices for operational adoption.	Formulate a comprehensive set of operational guidelines and best practices for forensic practitioners and incident response teams, detailing the necessary organizational policies, technical skills, and legal prerequisites for the practical, effective, and ethical adoption of the CFF in real-world hybrid/multi-cloud investigations.

4. Methodology

The research methodology follows five stages: literature review, framework design, prototype implementation, experimental evaluation, and analysis.

4.1 Literature Review

Conduct a systematic literature review of cloud forensics research, standards (NIST, ISO), vendor documentation (AWS, Azure, GCP), and existing forensic tools (Volatility, Sleuth Kit, FTK, ELK). Identify gaps and best practices.

4.2 Framework Design

Define the CFF architecture, components, interfaces, and workflows. Emphasis on:

- API-driven evidence acquisition (cloud provider logs, object storage, VM metadata)
- Forensic snapshotting of VMs and persistent volumes
- Memory capture and analysis for running instances
- Centralized log collection and correlation (ELK stack)
- Tamper-evident chain-of-custody (lightweight blockchain ledger)
- Role-based access and audit trails

4.3 Prototype Implementation

Implement CFF prototype using:

- Cloud platforms: AWS (IaaS) and OpenStack (private cloud)
- Tools: Volatility (memory analysis), Sleuth Kit & Autopsy (disk/file analysis), ELK (log ingestion and search), custom scripts to call provider APIs for snapshots and metadata
- Chain-of-custody: Hyperledger Fabric or a simplified permissioned blockchain for logging evidence transfer events

4.4 Experimental Scenarios

Design and execute three incident simulations:

- Scenario A: Data Exfiltration — unauthorized upload of confidential files from an EC2 instance to external cloud storage.

- Scenario B: Insider Tampering — privileged user modifies logs to hide malicious activities.
- Scenario C: Ransomware Infection — filesystem encryption of attached volumes across tenants.

For each scenario, measure:

- Time to identify and begin evidence acquisition
- Time to complete VM/image snapshot and memory capture
- Evidence integrity (SHA-256 hash matching across stages)
- Scalability (number of concurrent instances handled)
- Chain-of-custody tamper detection (attempted and detected modifications)

4.5 Evaluation

Compare CFF metrics against a baseline (traditional manual forensic workflow adapted for cloud by investigators) and analyze improvements and limitations.

5. System Design

5.1 Architecture Overview

The Cloud Forensics Framework comprises the following modules:

1. Detection & Alerting Module
   - Sources: IDS/IPS, SIEM alerts, user reports, cloud-native alerts (e.g., AWS GuardDuty).
   - Function: Trigger investigation workflows and collect initial metadata.
2. Evidence Acquisition Module
   - Components:

    - API Collector: Uses provider APIs (AWS SDK, OpenStack APIs) to fetch CloudTrail, audit logs, object storage metadata, VM metadata.
    - Snapshot Manager: Initiates VM snapshots, volume snapshots, and image export.
    - Memory Grabber: Uses in-guest agents (where available) or hypervisor-assisted memory dumps for live instances.

3. Log Aggregation & Correlation
    - ELK Stack ingests logs (cloud logs, application logs, network flow logs) and provides correlation and timeline construction.
4. Analysis Module
    - Disk & File Analysis: Sleuth Kit / Autopsy
    - Memory Analysis: Volatility plugins
    - Timeline Analysis: Correlate events across logs, snapshots, and memory artifacts.
5. Chain-of-Custody Ledger
    - A permissioned blockchain records metadata for each acquired artifact: timestamp, hash, collector identity, operation type, and retention policy. Each ledger entry is immutable and auditable.
6. Reporting & Case Management
    - Automated report generator producing a forensically-sound report with artifact hashes, acquisition steps, and analyst notes.

5.2 Workflow (High-level)

1. Alert triggers → Investigator initiates CFF case.
2. API Collector pulls relevant logs and metadata.
3. Snapshot Manager creates VM/volume snapshots and exports copies to a secured evidence store.
4. Memory Grabber captures volatile memory where feasible.
5. Each artifact is hashed (SHA-256) and hash recorded to the Chain-of-Custody Ledger.

6. Logs and artifacts are ingested into ELK and analysis tools for correlation and deep inspection.
7. Findings compiled into a final forensic report and retained per policy.

5.3 Security and Privacy Considerations

- Strict role-based access control (RBAC) for evidence operations.
- Data minimization to avoid collecting unrelated tenant data.
- Legal and jurisdictional compliance checks prior to evidence acquisition (warrants, provider policies).

6. Implementation (Prototype)

The core objective of the implementation phase was to develop a working prototype of the Cloud Forensics Framework (CFF) to validate its design and operational efficiency. This required the selection and integration of various cloud-native and open-source components, detailed below:

1. Prototype Environment and Architecture

The CFF prototype was deployed and tested across a **hybrid cloud environment** to ensure vendor-agnosticism and broader applicability:

- **Public Cloud: Amazon Web Services (AWS)**, specifically in the us-east-1 region, was utilized to test the core challenges of multi-tenancy and API-based acquisition, leveraging its comprehensive set of forensic-relevant services (e.g., S3, EC2).
- **Private Cloud:** An **OpenStack-based private cluster** was established to test the CFF's utility in corporate environments where investigators retain some level of hypervisor control, thus addressing hybrid deployment scenarios.
- **Log and Analysis Backbone:** The **ELK stack (Elasticsearch, Logstash, Kibana)** was deployed on a secure, dedicated Virtual Machine (VM) to serve as the centralized

platform for **log aggregation and correlation**. This enabled rapid searching and timeline reconstruction across diverse, distributed cloud logs.

- **Chain-of-Custody Ledger:** A **Fabric-based ledger** was implemented on a small, permissioned blockchain network. This immutable ledger provides cryptographic integrity for the entire chain-of-custody, recording and timestamping every forensic action, including evidence acquisition, transfer, and analysis access.

2. Component Integration and Functionality

The CFF is realized through the following integrated components, ensuring an automated and forensically sound workflow:

- **Automated Acquisition Scripts:** Core **Python scripts** were developed using the **Boto3 (AWS SDK)** and the **OpenStack SDK**. These scripts automate the forensic acquisition process by programmatically invoking native cloud APIs to create forensically sound **VM snapshots (disk image creation)** and securely export designated logs and data. This API-centric approach ensures rapid capture before evidence volatility becomes a factor.
- **Forensic Agents:** An **optional, lightweight agent** was developed and deployed on test VMs. The primary purpose of this agent is to facilitate **volatile memory capture** when direct hypervisor-level memory access is restricted (common in public cloud IaaS) or as a backup mechanism, ensuring that all volatile artifacts are captured.
- **Analysis Workflow:** The acquired evidence is processed using a suite of specialized tools: **Autopsy** was utilized for traditional filesystem artifact analysis, **Volatility Framework** was employed for in-depth analysis of captured memory images, and **custom Python scripts** were developed to automate the correlation of analysis findings with the centralized logs from the ELK stack to build a unified timeline.
- **Secure Evidence Store:** All acquired evidence is stored in dedicated repositories designed for integrity: a **secure Amazon S3 bucket** with **Object Lock** enabled (where available) was used for public cloud evidence, ensuring write-once-read-many (WORM) compliance. For the private cloud, local secure storage with equivalent access logging

and encryption controls was used, ensuring non-repudiation and chain-of-custody for all preserved data.

This integrated implementation demonstrates the CFF's capability to orchestrate complex forensic tasks across multi-vendor cloud environments using automated, verifiable, and forensically sound methods.

7. Result Analysis

7.1 Experimental Setup Recap

- 30 test instances across two providers (20 in AWS, 10 in OpenStack).
- Simulated incidents executed during a 48-hour window.
- Baseline: manual investigator workflow (requesting snapshots via console, manual log downloads).

7.2 Quantitative Results

Metric	**Baseline (manual)**	**CFF Prototype**	**Improvement**
Average time to start acquisition (minutes)	28.4	9.2	67.6% faster
Average time to complete VM snapshot & export (minutes)	52.1	34.0	34.7% faster
SHA-256 integrity check mismatches	0 / 90	0 / 150	—
Concurrent instances handled	5	25	5x
Chain-of-custody tamper detection (simulated)	Not present	Detected & logged	Significant

Notes: Times include orchestration overhead and assume API rate limits typical of public clouds. The CFF prototype reduced manual wait times by automating API calls and parallelizing snapshot operations. Integrity checks matched across all artifact transfers.

7.3 Qualitative Findings

- Timeliness: API-driven automation drastically reduced time-to-acquisition, critical when volatile evidence may be lost.
- Integrity & Auditability: Hashes recorded on the blockchain ledger provided immutable provenance; auditors found reports easier to validate.
- Scalability: Parallel snapshotting and centralized correlation scaled to medium-sized tests; larger enterprise-scale scenarios will require rate-limit and cost considerations.
- Legal & Privacy Constraints: Provider policies and legal jurisdiction checks sometimes introduced unavoidable delays (e.g., cross-border data), emphasizing need for pre-established agreements and SLAs with CSPs.

7.4 Limitations Observed

- Memory Capture Constraints: In some configurations, hypervisor-level memory capture was not available; in-guest agents required administrative control.
- Provider Heterogeneity: Differences in API semantics across providers require abstraction layers and provider-specific modules.
- Costs: Snapshotting and storage for retention increased costs; cost-benefit analysis recommended before large-scale deployment.

8. Future Research Directions

The successful implementation and evaluation of the Cloud Forensics Framework (CFF) establish a robust foundation for cloud-native investigations. Building upon this, future research should strategically focus on extending the CFF's capabilities to meet the rapidly

evolving enterprise requirements and address the continuous innovation within cloud architectures.

1. Advanced Multi-Cloud and Hybrid Orchestration

A key area for extension is the development of sophisticated techniques for **multi-cloud and hybrid orchestration at the enterprise scale**. This goes beyond simple support for multiple vendors; it requires designing an **agnostic control plane** capable of dynamically initiating forensic acquisition across different, disparate Cloud Service Providers (CSPs) and local on-premises infrastructure simultaneously. Research must focus on automated **resource mapping and correlation** across heterogeneous APIs and log formats (e.g., mapping an AWS resource ID to an Azure equivalent). The challenge lies in ensuring a single, unified chain-of-custody ledger is maintained seamlessly despite the evidence being geographically and architecturally distributed.

2. Stronger Legal-Compliance Automation and Adaptability

Further work is critically needed in developing **stronger legal-compliance automation**. This involves empowering the framework with the ability to dynamically adapt its acquisition and analysis procedures based on the **varying international jurisdictional requirements** (e.g., GDPR, CCPA, PIPL). Specifically, the CFF should incorporate modules for **geo-fencing evidence capture** and **automated redaction** or **differential data filtering** based on the legal standing of the subject or the data’s physical location at the time of the incident. This adaptation must be auditable and recorded immutably on the chain-of-custody ledger to uphold admissibility standards in diverse legal venues.

3. Standardized Provider-Forensics Interfaces and Protocols

A crucial, long-term focus is the advocacy for and development of **standardized provider-forensics interfaces** with major CSPs. Currently, forensic investigators rely on general-purpose APIs (like AWS Boto3), which can be rate-limited or lack deep-level forensic fidelity. Collaboration with industry bodies is essential to define **explicit forensic protocols** that allow authorized investigators to perform actions like **low-level virtual disk cloning**, **dedicated memory acquisition calls**, and **guaranteed preservation of highly volatile data** without interrupting the cloud tenant's operations. This standardization would dramatically streamline the acquisition and preservation process, benefiting the entire digital forensics community.

4. Container and Serverless Forensics

Finally, as organizations rapidly shift to microservices, future research must expand the CFF to provide robust support for **container and serverless forensics**. This involves developing novel techniques for capturing transient, state-less evidence (e.g., from **Kubernetes Pods or AWS Lambda functions**) where traditional disk imaging is impossible. This demands a focus on **runtime activity monitoring**, **event-driven preservation triggers**, and the rapid correlation of ephemeral network flows and application logs to reconstruct complex attack narratives.

9. Conclusion

This research presents a practical Cloud Forensics Framework that addresses critical challenges in investigating cloud-based cybercrimes. By leveraging cloud APIs for rapid evidence acquisition, supporting memory and disk analysis, centralizing log correlation, and using an immutable ledger for chain-of-custody, the framework improves response speed, ensures integrity, and scales beyond manual workflows. Prototype evaluation across common incident scenarios demonstrates meaningful gains in acquisition time and operational scalability.

For operational adoption, organizations should establish pre-authorized agreements with cloud providers, deploy 9 readiness measures (logging, snapshot policies), and integrate CFF components into incident response playbooks. Future research should focus on multi-cloud orchestration at enterprise scale, stronger legal-compliance automation, and standardized provider-forensics interfaces.

References

1. M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. Bin 6, Forensics investigation challenges in cloud computing environments, 2012. https://doi.org/10.1109/CyberSec.2012.6246092.

2. K. Sharma, P. K., Kaushik, P. S., Agarwal, P., Jain, P., Agarwal, S., and Dixit, Issues and challenges of data security in a cloud computing environment, in *Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017, pp. 560–566.

3. U. Anwar, H. A. Umair, A. Sikander, and Z. U. Abedin, Government cloud adoption and architecture, 2019. https://doi.org/10.1109/ICOMET.2019.8673457.

4. J. Baldwin, O. M. K. Alhawi, S. Shaughnessy, A. Akinbi, and A. Dehghantanha, Emerging from the cloud: a bibliometric analysis of cloud forensics studies, *Advances in Information Security*, 2018.

5. L. Chen, N.-A. Le-Khac, S. Schlepphorst, and L. Xu, Cloud Forensics, *Security, Privacy, and Digital Forensics in the Cloud*, pp. 201–216, 2019.

6. S. Biggs and S. Vidalis, Cloud computing: the impact on digital forensic investigations,*Conference: Internet Technology and Secured Transactions, 2009. ICITST*. 2009. https://doi.org/10.1109/ICITST.2009.5402561

7. Zafarullah, F. Anwar, and Z. Anwar, Digital forensics for Eucalyptus, in *Proceedings - 2011 9th International Conference on Frontiers of Information Technology, FIT 2011*, pp. 110–116, 2011. https://doi.org/10.1109/FIT.2011.28.

8. S. B. S. Farid Daryabar, A. Dehghantanha, N. I. Udzir and N. Fazlida Binti Mohd Sani, A survey about impacts of cloud computing on digital forensics, *International Journal of Cyber-Security and Digital Forensics*, Vol. 2, No. 2, pp. 77–94, 2013.

9. D. Reilly, C. Wren, and T. Berry, Cloud computing: Forensic challenges for law enforcement, *Internet Technol. Secur. Trans. (ICITST), 2010 Int. Conf.*, 2010.

10. B. Martini and K. K. R. Choo, An integrated conceptual digital forensic framework for cloud computing, *Digital Investigation*, Vol. 9, No. 2, pp. 71–80, 2012. https://doi.org/10.1016/j.diin.2012.07.001.

11. J. Plunkett, N.-A. Le-Khac, and T. Kechadi, Digital Forensic Investigations in the Cloud: A Proposed Approach for Irish Law Enforcement, *11th Annual IFIP WG 11.9 International Conference on Digital Forensics (IFIP119 2015), Orlando, Florida, United States,*, 2015.

12. W. Yassin, M. Faizal Abdollah, R. Ahmad, Z. Yunos and A. Ariffin, Cloud forensic challenges and recommendations: a review, *Journal Cyber Security*, Vol. 2, No. 1, pp. 19–29, 2020.

13. B. Manral, G. Somani, K. K. R. Choo, M. Conti and M. S. Gaur, A systematic survey on cloud forensics challenges, solutions, and future directions, *ACM Computing Survey*, 2019. https://doi.org/10.1145/3361216.

14. A. Pichan, M. Lazarescu and S. T. Soh, Cloud forensics: technical challenges, solutions and comparative analysis, *Digital Investigation*, 2015. https://doi.org/10.1016/j.diin.2015.03.002.

15. B. Martini and K. K. R. Choo, Cloud forensic technical challenges and solutions: a snapshot, *IEEE Cloud Computing*, 2014. https://doi.org/10.1109/MCC.2014.69.

16. P. Dixit, R. Kohli, A. Acevedo-Duque, R. R. Gonzalez-Diaz and R. H. Jhaveri, Comparing and analyzing applications of intelligent techniques in cyberattack detection, *Security and Communication Networks*, 2021. https://doi.org/10.1155/2021/5561816.

17. V. Subramaniyaswamy, et al., Somewhat homomorphic encryption: ring learning with error algorithm for faster encryption of IoT sensor signal-based edge devices, *Security and Communiction Networks*, 2022. https://doi.org/10.1155/2022/2793998.

18. V. Prakash, A. Williams, L. Garg, C. Savaglio and S. Bawa, Cloud and edge computing-based computer forensics: challenges and open problems, *Electronics*, Vol. 10, No. 11, pp. 1229, 2021. https://doi.org/10.3390/electronics10111229.

19. J. Han, J. Kim, and S. Lee, 5W1H-based expression for the effective sharing of information in digital forensic investigations, *arXiv Prepr. arXiv2010.15711*, 2020.

20. R. Mckemmish, What is forensic computing ?, *Change*, Vol. 118, No. 118, pp. 1–6, 1999.

21. L. Le-Khac, N. A., Plunkett, J., Kechadi, M. T., and Chen, Digital forensic process and model in the cloud, *Security, Privacy, and Digital Forensics in the Cloud*, p. 239, 2019.

22. M. Khanafseh, M. Qatawneh and W. Almobaideen, A survey of various frameworks and solutions in all branches of digital forensics with a focus on cloud forensics, *International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 8, pp. 610–629, 2019. https://doi.org/10.14569/ijacsa.2019.0100880.