Blockchain-Based Evidence Integrity System: A Secure Framework for Digital Forensic Evidence Management

Bhuvanagiri Supreet Sadashiv[1], Monisha C[2], Karan Gowda SY[3], Aditya Ramakrishna Bhat[4], Suhas J[5], N. Vishnu Venkatesh[6]
[1,2,3,4,5,6]*JAIN Deemed to be University*

Abstract—**Digital evidence has become central to modern forensic work, yet keeping that evidence intact, authentic, and legally admissible remains a persistent challenge. Existing evidence management platforms tend to rely on centralised storage, which creates clear vulnerabilities: a single point of failure, exposure to insider manipulation, and an audit trail that can be quietly altered. To close these gaps, this paper introduces a blockchain-based evidence integrity system that keeps a secure, transparent, and tamper-evident record of digital evidence from collection through to courtroom presentation. At its core, the system applies SHA-256 cryptographic hashing to produce a unique fingerprint for every evidence file; Merkle Tree structures then allow groups of files to be verified in a single efficient operation. Hash values and audit events are written to an immutable distributed ledger, so any unauthorised change is caught immediately. Access to the system is governed by JSON Web Token (JWT) authentication paired with role-based controls, ensuring that only personnel with appropriate clearance can interact with sensitive material. A working prototype built on FastAPI demonstrates the full workflow: evidence submission, hash-based verification, real-time monitoring, and automated report generation. Testing showed perfect integrity-verification accuracy, zero missed tampering events, and a complete, court-ready chain of custody outcomes that position this framework as a practical option for law enforcement agencies, forensic practitioners, and legal institutions seeking stronger evidentiary standards.**

Index Terms—**Blockchain; Digital Forensics; Evidence Integrity; Chain of Custody; SHA-256; Merkle Tree; JSON Web Token (JWT); Tamper Detection; Secure Evidence Management**

I. INTRODUCTION

Widespread adoption of cloud platforms, smartphones, and Internet of Things (IoT) devices has flooded modern investigations with digital material. Logs, metadata, communication records, and multimedia content generated by these technologies now feature prominently in criminal prosecutions, cybersecurity incident reviews, and corporate inquiries alike, often providing the most objective and detailed account of events available to investigators(Vishnu Venkatesh & Das, 2026).

Yet digital evidence carries a fundamental weakness: it can be altered, copied, or quietly corrupted without leaving the kind of visible marks that tampered physical evidence typically shows. These characteristics make authenticity difficult to prove and create real risks around admissibility. Protecting the integrity of digital material across its entire lifecycle from first acquisition through final submission to a court has therefore become one of the central concerns of the forensic discipline.

Central to proper evidence handling is the chain of custody a documented record of every person who has handled a piece of evidence and every action taken with it, stretching from initial seizure to courtroom presentation. Current systems typically rely on centralised storage and partly manual logging, a combination that introduces serious weaknesses. A single server failure can wipe records; a disgruntled insider can manipulate logs; and auditing a fragmented paper trail is slow and error prone. Deploying cryptographic hashing helps, but if the resulting hash values are stored in the same centralised database, an attacker who compromises that database can alter both the evidence and its hash defeating the protection entirely(Shukla et al., 2023).

Blockchain technology offers a fundamentally different architecture. Rather than trusting a single authoritative repository, a blockchain distributes records across multiple nodes, linking each new entry cryptographically to everything that came before. Retroactively altering any entry would require re-

computing the chain from that point forward across the majority of participating nodes computationally infeasible in practice. This combination of decentralisation, immutability, and built-in auditability makes blockchain well suited to evidence management, where the stakes of undetected tampering are particularly high.

This paper describes a blockchain-based evidence integrity system built around these properties. Raw evidence files are never written to the chain itself; instead, the system records each file's SHA-256 hash alongside structured metadata, a design that protects evidential privacy while still allowing anyone to verify that a file has not been touched since it was logged. When investigators deal with large batches of files, Merkle Tree construction reduces the verification workload to checking a single root hash rather than every file individually. All user sessions are authenticated with JSON Web Tokens, and access rights are governed by role-based controls that keep investigators, administrators, and auditors within their respective boundaries.

Taken together, these components blockchain immutability, cryptographic hashing, a role-governed web interface, and automated audit logging form a unified framework covering evidence submission, verification, monitoring, and reporting. The overarching goal is to make digital evidence more reliable and legally defensible by giving every stakeholder investigator, legal teams, judges a transparent, independently verifiable record of how that evidence was handled(Venkatesh et al., 2023).

II. LITERATURE REVIEW

Scholarship at the intersection of blockchain and digital forensics has grown steadily since Nakamoto [1] introduced the foundational distributed-ledger concept, with researchers drawn to its three defining properties: immutability, decentralisation, and transparency. These properties translate naturally into forensic requirements, and a substantial body of work now explores how they can be harnessed to secure evidentiary records.

Crosby et al. [6] made an early case for blockchain as an architecture that removes single points of failure through distributed consensus, while Narayanan et al. [2] examined the cryptographic primitives that underpin it hashing functions, digital signatures, and Merkle Trees explaining how each contributes to reliable data verification in practice.

On the forensic side, Casey [3] drew attention to persistent weaknesses in conventional chain-of-custody management, arguing that current systems regularly fall short on transparency and auditability. Blockchain's immutable audit trail directly addresses both shortcomings, offering a mechanism that courts can examine and trust.

A range of concrete system proposals has followed. Akinseye et al. [13] built a threat-investigation platform on Hyperledger Fabric, keeping bulky evidence files off-chain while anchoring their metadata securely on the ledger. Regueiro and Urquizu [14] took a complementary approach in cloud settings, designing a blockchain-backed certification layer to strengthen the reliability of evidence auditing in distributed storage environments.

Scalability has driven interest in hybrid designs that pair blockchain integrity guarantees with distributed storage back-ends. Li [15] presented a digital-archival model that uses blockchain for verification and IPFS for cost-effective data storage, while Verma et al. [16] confirmed that such architectures deliver measurable performance and security improvements when handling the large file volumes typical of forensic casework.

More recently, researchers have begun weaving artificial intelligence into these frameworks. Vignesh et al. [17] combined blockchain with anomaly-detection models to flag tampered evidence without human intervention, and Abisha et al. [18] paired a PBFT blockchain with XGBoost classifiers to push tamper-detection accuracy to 98% at scale. Akhtar [19] applied similar machine-learning techniques to the specific challenges of IoT-based forensic environments, where device heterogeneity complicates evidence collection.

Despite this progress, a pattern of limitation runs through the literature: most systems are either conceptual rather than operational, or narrowly scoped to a particular domain such as IoT forensics or cloud auditing. Very few attempts have been made to build a genuinely end-to-end platform that combines secure evidence submission, real-time integrity verification, role-differentiated access control, and automated report generation within a single cohesive tool(Shenoy et al., 2025).

2.1 Research Gap

Examining the literature as a whole reveals several gaps that the present work is positioned to address:

- Few, if any, systems integrate the complete evidence-management workflow submission, verification, access control, logging, and reporting into a single operational platform.
- User-centred design has received relatively little attention, yet usability matters greatly when investigators are working under time pressure in high-stakes environments.
- Real-time integrity verification the ability to detect tampering the moment it occurs rather than during a scheduled audit remains largely unimplemented in current proposals.
- Role-based access controls have not been adequately tailored to the distinct responsibilities of investigators, administrators, and auditors within a forensic workflow.
- Minimal work on automated reporting and audit generation suitable for legal contexts.

The predominance of conceptual or early-prototype work leaves a clear gap for a system designed with real-world operational requirements in mind a gap this study sets out to fill.

III. PROBLEM STATEMENT AND OBJECTIVES

3.1 Problem Statement

As investigations have come to rely more heavily on digital material, the shortcomings of conventional evidence management have become harder to ignore. Systems built around centralised architectures carry structural vulnerabilities that no amount of careful administration can fully eliminate: a breach of the central server compromises everything stored there, and a single corrupted database can unravel an entire case. Cryptographic hashing is now routinely applied to generate evidence fingerprints, but placing those hash values in the same centralised repository as the evidence itself undermines the protection both the file and its verification token can be changed together. Chain-of-custody records suffer from similar fragility: manual or partially automated logging is prone to human error and leaves gaps that defence counsel can exploit. Cloud migration has introduced a further layer of difficulty, bringing jurisdictional ambiguity, uneven access controls, and traceability problems that traditional audit mechanisms were not designed to handle. In court, any credible question about how evidence was handled who accessed it, whether it was altered can be enough to have it excluded, directly affecting the outcome of a prosecution.

What is needed, therefore, is an architecture that removes centralised trust, makes tampering immediately visible, and keeps a continuous, independently verifiable record of every interaction with every piece of evidence from the moment of seizure through to its presentation before a judge.

3.2 Objectives

Working toward this goal, the study pursued the following specific objectives:

- Design a secure web-based system enabling authorized users to upload digital evidence with automatically captured metadata (timestamps, user identity, and case references).
- Implement cryptographic hashing (SHA-256) and store hash values on a blockchain to ensure immutability and detect unauthorized modifications.
- Utilize Merkle Tree structures to enable fast and scalable verification of multiple evidence files simultaneously.
- Develop an automated logging mechanism recording every interaction with evidence to ensure complete chain-of-custody traceability.
- Implement JWT-based authentication and role-based access control (RBAC) restricting system access to authorized users only.
- Design the system to efficiently manage and process large volumes of multi-file digital evidence.
- Generate detailed evidence reports including metadata, verification status, and chain-of-custody logs suitable for legal and audit purposes.
- Ensure all operations are recorded in an immutable and transparent manner, enhancing trust among all stakeholders.

IV. METHODOLOGY

The research follows a design-and-build methodology: rather than modelling or simulating forensic processes in the abstract, the study constructs a working system and then subjects it to empirical testing. Theoretical

constructs drawn from blockchain, cryptographic hashing, and secure authentication are translated into concrete software components that can operate in conditions resembling real forensic practice. The guiding design requirements are secure evidence submission, tamper-proof hash storage, scalable verification, and end-to-end traceability.

4.1 System Model Overview

Figure 1 outlines the sequence of operations that moves a piece of evidence through the system. After passing JWT authentication, an investigator uploads a file through the web interface; the system simultaneously captures metadata evidence ID, timestamp, user identity, and file attributes without requiring any manual data entry. SHA-256 hashing then runs on the file to produce its cryptographic fingerprint, which is written to the blockchain ledger together with the associated metadata. Where batches of files are submitted at once, their individual hashes are assembled into a Merkle Tree so that a single root hash can serve as the verification anchor for the entire batch. Integrity checks work by re-hashing the file on demand and comparing the result against the blockchain record; any discrepancy signals tampering. Every upload, verification, and access event is automatically appended to the chain-of-custody log.

Figure 1: System Workflow of the Blockchain-Based Digital Evidence Integrity System

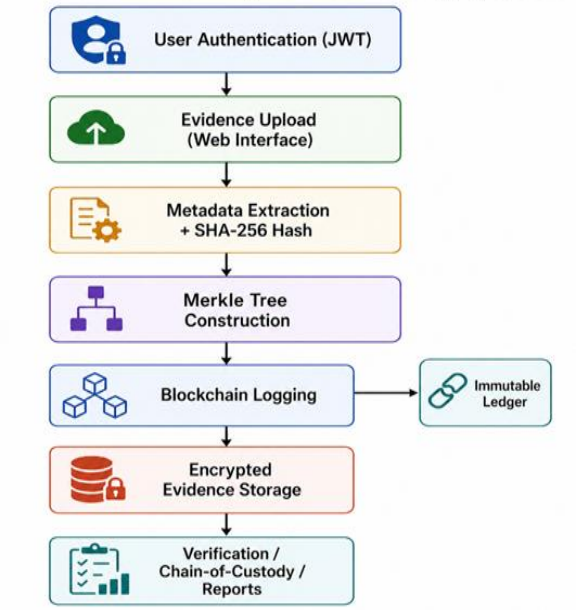


Fig. 1. System workflow of the blockchain-based digital evidence integrity system.

4.2 Secure Evidence Submission and Traceability

Investigators interact with a web interface that accepts evidence files in any common format documents, images, log files, multimedia without requiring specialised client software. The submission process captures a standard set of metadata fields automatically at the moment of upload: a system-generated Evidence ID, the associated case reference number, file name and type, a UTC timestamp, the authenticated user's identity and role, file size and system attributes, and the storage path assigned to the file. Assigning a unique Evidence ID to each item from the outset means there is never any ambiguity about which file is being discussed, verified, or presented in court, and the complete metadata record makes it possible to trace any item from its first appearance in the system through to its final use in legal proceedings.

4.3 Blockchain-Based Tamper Prevention

The blockchain functions as the system's tamper-prevention backbone. Evidence files themselves are never written to the chain only their SHA-256 hashes and associated metadata are recorded, which keeps sensitive material private while still enabling anyone with access to verify the file's integrity. Each block in the chain carries the evidence hash, a reference to the preceding block's hash, a UTC timestamp, the identity of the submitting user, current verification status, and relevant metadata.

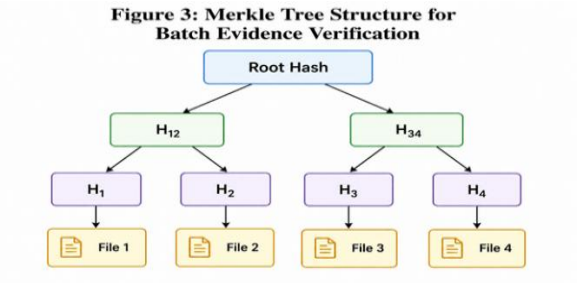
Because SHA-256 is a one-way function, even the smallest change to the original file produces an entirely different hash, so any tampering is surfaced the moment a verification check is run. The chain structure amplifies this protection: altering any historical block would break the hash linkage to every subsequent block, making silent modification computationally infeasible. As a secondary benefit, the blockchain record serves as a tamper-resistant audit log in its own right, capturing every significant action upload, verification, report generation in a form that cannot be retrospectively changed.

4.4 Evidence Integrity Verification Using Hashing and Merkle Tree

At upload, every evidence file receives a SHA-256 hash a 256-bit fingerprint that is unique to that file's exact contents. Verification is straightforward: re-hash the file and compare the result against what the blockchain holds. Matching values confirm that

nothing has changed; a discrepancy, however small, confirms that something has. Because SHA-256 exhibits the avalanche effect, flipping even a single bit in the original file produces a completely different hash output, giving the mechanism exceptional sensitivity to subtle modifications.

When a case involves multiple files, the system constructs a Merkle Tree as shown in Figure 3. Each leaf node holds an individual file's SHA-256 hash; pairs of adjacent hashes are concatenated and re-hashed iteratively, level by level, until only a single Merkle Root remains. That root is the value committed to the blockchain, so the integrity of an entire batch of files can be confirmed or refuted by checking one reference point. Any change to any file in the batch propagates upward through the tree, altering the root and making tampering immediately apparent regardless of which file was touched. This design substantially cuts verification overhead compared with file-by-file checking and keeps the system practical as evidence volumes grow.



Root hash stored on blockchain. Any file change propagates upward, altering the root hash immediately.

Fig. 3. Merkle Tree construction for efficient batch verification of digital evidence files.

4.5 Chain of Custody Management

Every interaction between a user and an evidence item triggers an automatic log entry no manual recording is required. The system logs evidence uploads, file accesses, verification requests, and report-generation events, capturing the user identity, action type, UTC timestamp, and relevant evidence ID for each. These entries are written concurrently to the local SQLite database and to the blockchain ledger. The dual-storage approach provides redundancy against localised data loss and, because the blockchain copy is immutable, ensures that no one can quietly alter or delete a custody record after the fact. The result is a chronological audit trail that meets the exacting documentation standards expected in forensic and legal proceedings.

4.6 Access Control and Authentication

Authentication relies on JSON Web Tokens. When a user logs in with valid credentials, the system issues a signed JWT encoding that user's ID, role assignment, session validity window, and expiration time. Every subsequent request to the system must carry a valid, unexpired token; requests without one are rejected outright, irrespective of the endpoint they target.

Two roles cover the main operational needs. Investigators may upload evidence, run integrity checks, and access reports for their own cases. Administrators inherit all investigator capabilities and additionally can monitor evidence across all active cases, manage user accounts, review full audit logs, and adjust system configuration. Both successful logins and blocked access attempts are recorded in the audit log; entries can be anchored to the blockchain to give them the same permanence and tamper-resistance as evidence records.

By binding permissions tightly to verified identities and auditing every access event, the architecture guards against both external intrusion and the more subtle risk of insider misuse.

4.7 Algorithms Used in the System

4.7.1 SHA-256 Hash Generation Algorithm

When a file is submitted, the system reads it as a binary stream and feeds it through the SHA-256 engine. Internally, the algorithm splits the data into 512-bit blocks and subjects each block to a series of compression operations, bitwise transformations, and modular additions. Intermediate state values carry forward from one block to the next, and when the final block has been processed the accumulated state collapses into a fixed 256-bit digest. That digest is written to the blockchain together with the file's metadata. Later, when a verification request arrives, the system re-runs the same hashing procedure on the file as it currently exists and compares the new digest against the one on the chain agreement means the file is intact; divergence means something has changed.

4.7.2 Blockchain Logging Algorithm

Logging an evidence submission involves packaging the computed hash (H) and its associated metadata (M) as a single blockchain transaction. The system

retrieves the hash of the most recent block (P) and constructs a new block containing H, M, P, and the current UTC timestamp (T). After passing consensus validation, the block is appended to the chain and a transaction ID is returned for use in future audit queries. Because every block carries a reference to its predecessor, any attempt to alter a historical record breaks the chain linkage and is immediately apparent on inspection.

4.7.3 Merkle Tree Construction Algorithm

For batch submissions, the algorithm begins by assembling the individual file hashes [H1, H2, …, Hn] as leaf nodes. Working up the tree, it concatenates adjacent pairs, re-hashes them, and repeats until a single Merkle Root emerges. Only this root is committed to the blockchain. To verify any individual file later, the system recomputes that file's hash and traverses the Merkle path upward, requiring far fewer hash computations than re-verifying every file in the batch independently an important efficiency gain when a case involves hundreds or thousands of files.

4.7.4 Evidence Verification Algorithm

Verification begins by accepting the file under inspection and generating a fresh SHA-256 hash. The system then retrieves the original hash recorded at upload time from the blockchain. If the two values agree, the file is marked Valid and the successful verification is added to the custody log. If they differ, the file is flagged as Tampered, an alert notifies the responsible administrator, and the mismatch event is permanently recorded on the chain. Either way, every verification outcome pass or fail becomes part of the immutable custody trail.

4.7.5 Chain of Custody Logging Algorithm

Logging is event-driven: any interaction with an evidence item upload, access, verification, report generation immediately triggers a log write. The captured fields are the user ID, action type, UTC timestamp, and evidence reference, assembled into a structured record that is written to the backend database. Where stronger guarantees are needed, a cryptographic hash of the log entry itself can be anchored on the blockchain, preventing anyone from quietly altering or deleting it after the fact. The cumulative result is a timestamped, tamper-resistant record of every interaction in chronological order the authoritative chain-of-custody document for forensic and legal review.

V. SYSTEM DESIGN AND IMPLEMENTATION

The system takes the form of a web application that brings together blockchain integrity controls, cryptographic hashing, and secure session management within a single coherent platform. Its internal structure follows a five-layer architecture illustrated in Figure 2 is chosen because separating concerns across layers makes each component easier to test, maintain, and scale independently.

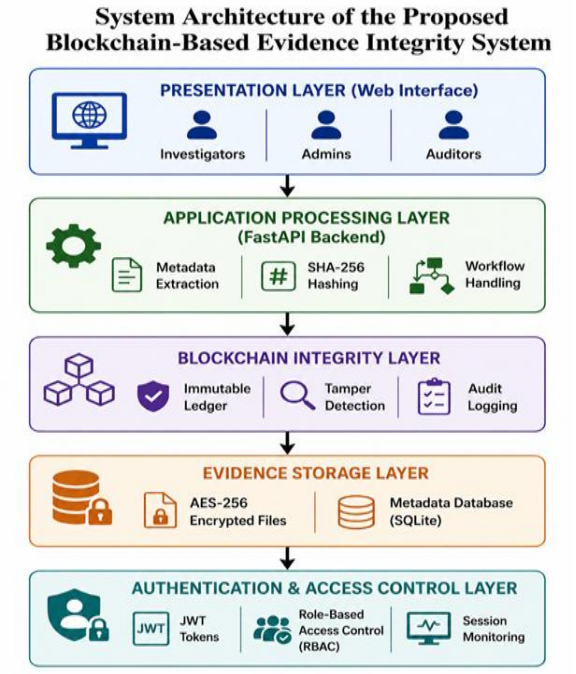


Fig. 2. Layered system architecture of the proposed blockchain-based evidence integrity system.

5.1 System Architecture

5.1.1 Presentation Layer

The Presentation Layer is the browser-based interface through which all three user roles interact with the system. Investigators and administrators log in through a secure portal, submit evidence via forms that accept a broad range of file formats (JPEG, PNG, PDF, DOCX, TXT, LOG, CSV, MP4, audio), and monitor case progress through dashboards that show real-time verification status. The interface surfaces alert notifications immediately when a hash mismatch or blocked access attempt is detected. Usability was treated as a first-class design requirement throughout, recognising that forensic work is conducted under time pressure and that an unintuitive interface increases the risk of procedural errors.

5.1.2 Application Processing Layer

FastAPI serves as the backbone of the Application Processing Layer, handling all core business logic and inter-layer communication. When evidence arrives, this layer extracts metadata, dispatches the file for SHA-256 hashing, and coordinates the resulting data across layers. A simple workflow engine moves each item through four status states are Uploaded, Under Review, Verified, and Archived, providing unambiguous lifecycle tracking that any authorised user can inspect at any time. This layer also generates audit log entries and dispatches them to the Blockchain Integrity Layer.

5.1.3 Blockchain Integrity Layer

The Blockchain Integrity Layer is where the system's trust guarantees are grounded. Every evidence upload and every significant system event produce a blockchain transaction; each new block references its predecessor through a cryptographic link, creating the chain structure that makes historical records tamper-evident. Raw evidence files are not stored on the chain, only hashes and metadata keeping the ledger compact while preserving the ability to verify any file at any time. Precise UTC timestamps embedded in each block support the construction of legally coherent chronological timelines. Smart contracts can additionally be configured to enforce access policies and trigger automatic alerts if a policy boundary is breached.

5.1.4 Evidence Storage Layer

Uploaded evidence files are held in an encrypted repository protected by AES-256, so even if an attacker gained physical or network access to the storage medium, the files would remain unreadable. Alongside each file the layer maintains a comprehensive metadata record which has name, type, size, case ID, investigator identity, SHA-256 hash, and storage path. Backup and recovery mechanisms guard against hardware failures and ransomware attacks. Version control prevents any original file from being overwritten; if processing requires a modified copy, that copy is kept separately and the original is preserved in its submitted state.

5.1.5 Authentication and Access Control Layer

The Authentication and Access Control Layer sit across all system entry points. Login triggers credential validation against the user database; success yields a signed JWT encoding the user ID, role, and token expiry. From that point, every API request must present a valid, unexpired token or it is rejected before any business logic executes. Within authenticated sessions, role assignments constrain what each user can see and do investigators are limited to their own evidence and reports, administrators have system-wide visibility and management capabilities. Every access event, including failed login attempts and blocked privilege-escalation attempts, is logged; these records can be anchored to the blockchain for the same immutability guarantees that apply to evidence records.

5.2 Implementation Technologies

The implementation is built entirely in Python, chosen for its mature cryptographic and web libraries. FastAPI provides the high-performance API layer and handles communication between the application logic and the blockchain module. SQLite serves as the lightweight relational store for metadata and logs during the prototype phase; migrating to a more robust database engine such as PostgreSQL would be straightforward for a production deployment. SHA-256 hashing is performed through Python's standard cryptographic library, and JWT handling is managed by a dedicated Python JWT package. All traffic between clients and the server is encrypted over HTTPS/TLS. The system is platform-independent and tested on Windows, Linux, and macOS and requires only a modern web browser on the client side.

5.3 System Event Log Structure

Table 1 describes the fields recorded in every system event log entry, along with the source from which each field value is drawn.

Table 1. Structure of System Event Log Fields and Data Sources

Field	Description	Source
event_id	UUIDv4 identifier	System Generated
user	JWT sub claim	JWT Payload
action	CRUD operation type	API Endpoint
evidence_id	SHA-256 of affected file	Blockchain
timestamp	Coordinated Universal Time (UTC)	NTP Server

5.4 Access Control and Role Management

Figure 5 shows the role-based access architecture in schematic form. Every user, regardless of role, passes through the JWT authentication gateway at login.

Once authenticated, the system routes them to the capability set that matches their assigned role: investigators can upload evidence, run verification checks, and pull reports on their own cases; administrators can do all of that and additionally monitor evidence across all cases, manage user accounts, and review system-wide audit logs; auditors receive read-only access to logs and chain-of-custody records, enough to carry out oversight functions without the ability to alter any record.

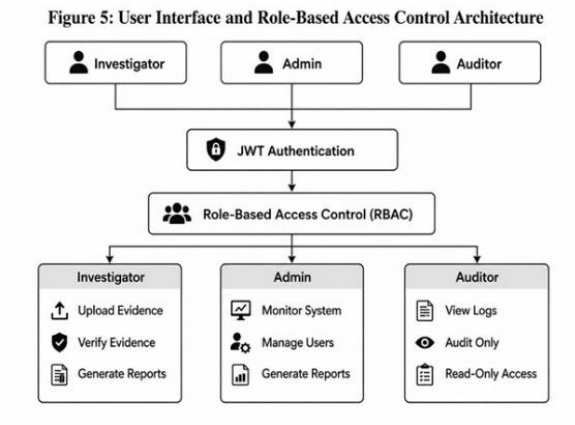


Fig. 5. User interface layer depicting role-based access control for investigators, administrators, and auditors.

Figure 4 traces the verification algorithm from file submission through to a final Valid or Tampered determination, making explicit the decision logic at each step.

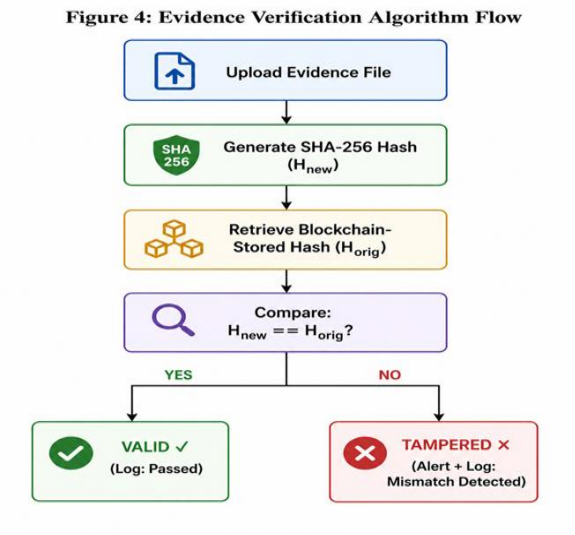


Fig. 4. Evidence verification algorithm flow comparing recomputed SHA-256 hash against blockchain-stored value.

VI. RESULTS AND ANALYSIS

The system was deployed in a controlled laboratory setting and put through a structured evaluation covering its principal functional areas. Test scenarios were designed to probe evidence upload reliability, hash generation consistency, deliberate tampering detection, multi-file batch handling, chain-of-custody completeness, access control robustness, and the quality of generated reports. Together, these scenarios exercise every major pathway through the system and provide a realistic picture of how it would behave under genuine forensic workloads.

6.1 Evidence Upload and Storage

Upload testing covered a representative cross-section of evidence formats: documents (PDF, DOCX, TXT), images (JPEG, PNG), log files (LOG, CSV), and screenshots. Every file was accepted and correctly indexed, with the full metadata set that is evidence ID, case reference, uploader identity, UTC timestamp, file size, and storage path captured automatically without any manual entry from the investigator. Both single-file submissions and multi-file batches completed without errors or data loss across repeated trials. The upload module showed the kind of format flexibility and consistency that real casework demands.

6.2 Hash Generation and Verification

Every uploaded file was hashed with SHA-256 and the result written to the blockchain. When verification runs were performed across multiple repetitions and file types such as files that had not been touched since upload were correctly identified as intact with perfect accuracy. Hash generation proved fully deterministic: the same file always produced the same digest, regardless of when or how many times the operation was repeated. This determinism is a prerequisite for reliable forensic use, and the results confirm it holds in practice.

6.3 Tamper Detection

To stress-test detection sensitivity, several evidence files were deliberately modified after upload using a range of techniques: straightforward content editing, metadata-only changes, combined file renaming and content modification, and timestamp manipulation. These scenarios were chosen to cover both obvious tampering and the more subtle alterations that a

sophisticated adversary might attempt. Every modified file produced a hash mismatch on re-verification, and none evaded detection. Table 2 records the outcomes in detail.

Table 2. Tamper Detection Results for Modified Evidence Files

Test File	Modification Type	Hash Status	Detection Result
File 1	Content Edit	Mismatch	Tampered – Detected
File 2	Metadata Change	Mismatch	Tampered – Detected
File 3	Rename + Content Edit	Mismatch	Tampered – Detected
File 4 (Control)	No Modification	Match	Valid – Integrity Confirmed

The sensitivity of SHA-256's avalanche effect was evident throughout: even the most minimal change to a file altering a single bit produced a completely different digest, making the modification immediately visible. This level of sensitivity is exactly what forensic work requires; defence challenges to evidentiary integrity are much harder to sustain when tampering, however subtle, is automatically surfaced.

6.4 Multi-File Handling and Merkle Tree Efficiency

Multi-file batch testing involved simultaneous submissions of mixed-format files belonging to different cases. Each file was processed independently its own Evidence ID, hash, log entry, and custody record with no conflicts or indexing errors arising across any batch. Merkle Tree construction reduced the verification workload for each batch to a single root-hash comparison, rather than a hash check per file. The performance difference compared with sequential file-by-file verification grew with batch size, confirming that the approach will scale effectively as case evidence volumes increase.

6.5 Chain of Custody and Audit Log Results

Across the full test cycle, every user action was captured in the audit log without gaps or formatting inconsistencies. The logged event types such as uploads, file access, verification requests, report generation were each represented with the expected fields: user identity, action type, UTC timestamp, and evidence reference, structured identically from entry to entry. Log synchronisation to the blockchain proceeded without errors, and once written, entries were confirmed to be immutable. The resulting custody record was complete, chronological, and consistent throughout, satisfying the documentation standards that forensic practitioners and courts expect.

6.6 Access Control and Authentication Results

Access control testing simulated all three user roles such as investigators, administrators, and completely unauthorised external parties, as well as edge cases such as expired tokens, absent tokens, and deliberate privilege-escalation attempts. In every case, authorised users received exactly the capabilities their role specified, no more and no less. Every unauthorised access attempt was blocked before reaching any protected resource, and the blocked attempt was immediately appended to the audit log. The module showed no susceptibility to session manipulation in any test variant, giving confidence that sensitive evidence data is adequately protected against both external attacks and insider overreach.

6.7 Report Generation

Report generation was tested across different evidence types and user roles. In every case, the system produced a complete report evidence metadata, SHA-256 hash value, verification status, and full custody history accurately and without delay. Formatting was consistent across all generated documents, producing output that reads as professionally prepared documentation rather than a raw data dump. These reports are structured to serve multiple audiences: internal review, court submission, and external compliance auditing, without requiring post-processing or reformatting by the investigator.

6.8 Overall Performance Evaluation

Table 3 summarises system performance across all evaluated dimensions.

Table 3. Overall System Performance Evaluation Summary

Evaluation Parameter	Result	Status
Integrity Verification	100% Accuracy	Pass
Tamper Detection	All Cases Detected	Pass
Multi-File Processing	Efficient (Merkle Tree)	Pass
Access Control (RBAC + JWT)	Fully Enforced	Pass

Chain-of-Custody Logging	Complete & Consistent	Pass
Report Generation	Accurate & Instant	Pass
Unauthorized Access Attempts	All Blocked & Logged	Pass

Taken together, the evaluation results paint a consistent picture: every tested dimension returned a pass, and the margin of success in the quantitative measures 100% verification accuracy, zero missed tamper events, zero successful unauthorised access attempts leave little ambiguity about the system's fitness for purpose. The combination of blockchain immutability, SHA-256 hashing, Merkle Tree batch verification, JWT-based authentication, and automated custody logging delivers a coherent framework that addresses the core operational and legal requirements of digital forensic evidence management.

VII. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This study has presented a blockchain-based evidence integrity system developed to tackle documented weaknesses in how digital evidence is currently managed. Centralised architectures the dominant approach in practice that leave evidence repositories vulnerable to tampering, offer limited transparency to outside auditors, and produce chain-of-custody records that can be quietly altered. These weaknesses collectively undermine the reliability and legal admissibility of the evidence they are supposed to protect. The system addresses these weaknesses through a combination of four mutually reinforcing components: blockchain-based immutable logging, SHA-256 fingerprinting, Merkle Tree batch verification, and JWT-governed role-based access. Critically, raw evidence files are never written to the chain only their hashes and metadata are, so the ledger carries no sensitive content while still enabling anyone with appropriate access to verify that a file has not changed since it was submitted.

Experimental evaluation confirmed that the system achieves perfect integrity-verification accuracy, detects all deliberate tampering attempts regardless of the modification strategy used, and maintains a complete, chronological chain of custody with no gaps. Merkle Tree construction keeps verification efficient as evidence volumes grow, and RBAC prevents both external intrusion and internal privilege misuse. Automated reporting translates raw logs and blockchain records into formatted documentation ready for court submission or compliance review without manual reformatting.

Overall, the framework represents a practical, scalable step forward for organisations such as law enforcement agencies, forensic service providers, or corporate security teams that need a more trustworthy and legally defensible approach to digital evidence management.

7.2 Limitations

These results should be read in light of some important constraints. Testing used a controlled dataset of moderate scale; performance with the large, heterogeneous file sets common in real investigations such as full disk images, raw network-traffic captures, mobile device extractions has not been measured. The prototype was not subjected to enterprise-scale deployment conditions such as sustained high concurrency or distributed multi-node operation, so bottlenecks that only emerge under production load remain uncharacterised. The blockchain component was evaluated on a private controlled network, which means enterprise-grade performance aspects, consensus latency under failure conditions, node synchronisation overhead, and transaction throughput at scale have yet to be tested. The system has not been piloted in a genuine law enforcement or court environment, so its practical admissibility in different legal jurisdictions is an open question. Finally, formal penetration testing and threat modelling have not been conducted; the access-control and cryptographic components perform well under the scenarios tested here, but a professional red-team exercise would be required before a production deployment.

7.3 Future Scope

Future development work will pursue several directions. Connecting the system to a distributed storage back-end IPFS or a cloud object store would overcome the scalability ceiling that comes with local encrypted storage, while also enabling multi-agency remote access. Migrating from the prototype's private blockchain to an enterprise network such as Hyperledger Fabric or a consortium Ethereum deployment would open the system to national and cross-border forensic applications. Embedding AI-

driven anomaly detection would shift the system from reactive to proactive, flagging unusual access patterns or potential insider threats before they cause damage. A companion mobile application would allow field investigators to upload evidence at the collection point eliminating the handling steps that currently occur between seizure and entry into a formal system. Stronger authentication mechanisms, including biometric verification and hardware security modules, would raise the bar against credential-based attacks. Finally, and perhaps most importantly, formal pilots conducted in partnership with law enforcement agencies and legal institutions would generate the evidence needed to answer questions of court admissibility and align the system with jurisdiction-specific evidentiary standards.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, Bitcoin and Cryptocurrency Technologies. Princeton, NJ, USA: Princeton Univ. Press, 2016.

[3] E. Casey, Digital Evidence and Computer Crime, 3rd ed. Amsterdam, Netherlands: Academic Press, 2011.

[4] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. London, U.K.: Pearson, 2017.

[5] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in Proc. IEEE Security and Privacy Workshops, 2015, doi: 10.1109/SPW.2015.27.

[6] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond Bitcoin," Applied Innovation Review, no. 2, pp. 6–19, 2016.

[7] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of blockchain technology," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3416–3452, 2018, doi: 10.1109/COMST.2018.2842460.

[8] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in Proc. IEEE Int. Congr. Big Data, pp. 557–564, 2017, doi: 10.1109/BigDataCongress.2017.85.

[9] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications," Telematics and Informatics, vol. 36, pp. 55–81, 2019, doi: 10.1016/j.tele.2018.11.006.

[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and solutions," in Proc. IEEE Conf. Internet Things, 2017, doi: 10.1109/iThings.2016.7873607.

[11] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.

[12] A. Akinbi, Á. MacDermott, and A. M. Ismael, "A systematic literature review of blockchain-based IoT forensic investigation process models," Forensic Science International: Digital Investigation, vol. 42, p. 301470, 2022.

[13] Akinseye et al., "Digital threat investigation system using Hyperledger Fabric for secure forensic evidence management," IEEE Transactions on Information Forensics and Security, 2023.

[14] Regueiro and B. Urquizu, "Blockchain-based evidence trustworthiness system in certification," Journal of Cybersecurity and Privacy, vol. 5, no. 1, p. 1, 2024.

[15] R. Li, "BDHAPI: A distributed digital archival model based on blockchain and IPFS," Information Sciences, 2025.

[16] V. V. Natarajan, P. Singhal, D. Pandey, M. Sharma, R. Rautdesai, D. Khubalkar, and A. Gupta, "Crime forecasting using historical crime location using CNN-based images classification mechanism," 2023, doi: 10.4018/978-1-6684-8618-4.ch013.

[17] T. Vignesh et al., "AI-integrated blockchain system for automated tamper detection in digital evidence," IEEE Transactions on Forensic Science, 2026.

[18] S. S. Shenoy and N. V. Venkatesh, "A predictive framework for real-time courtroom assistance using AI-based mock legal advisor," International Journal of Research and Analytical Reviews (IJRAR), vol. 12, no. 2, pp. 440–444, May 2025.

[19] M. S. Akhtar and T. Feng, “Using blockchain to ensure the integrity of digital forensic evidence in an IoT environment,” EAI Endorsed Transactions on Creative Technologies, vol. 9, no. 31, p. e2, 2022.

[20] L. Ahmad, S. Khanji, F. Iqbal, and F. Kamoun, “Blockchain-based chain of custody: Towards real-time tamper-proof evidence management,” in Proc. 15th Int. Conf. Availability, Reliability and Security, 2020.

[21] M. Shukla, V. Srivastav, M. D. Khare, and N. V. Venkatesh, “IoT-driven solutions for VANET trustworthiness: Examining misconduct and position security challenges,” Multidisciplinary Reviews, vol. 6, p. 2023ss059, 2024, doi: 10.31893/multirev.2023ss059.

[22] V. V. Natarajan, P. Das, and A. Rajiv, “A robust detect and avoid system for autonomous drone navigation,” NexusTech, vol. 1, p. 2026004, 2026, doi: 10.31893/tech.2026004.

[23] R. Mishra et al., “Blockchain chain-of-custody system for secure inter-agency evidence sharing,” Digital Investigation Journal, 2025.

[24] M. N. V. Venkatesh, D. A. Rajiv, M. P. Das, and M. S. Warrier, “Vantage point recreation: A novel approach in endpoint security for smart homes,” International Journal of Innovative Research in Technology (IJIRT), 2026, doi: 10.64643/IJIRTV12I8-191180-459.

[25] N. Shaji and N. V. Venkatesh, “Smart evidence management using blockchain,” International Journal of Innovative Research in Technology (IJIRT), 2026, doi: 10.64643/IJIRTV12I11-199202-459.