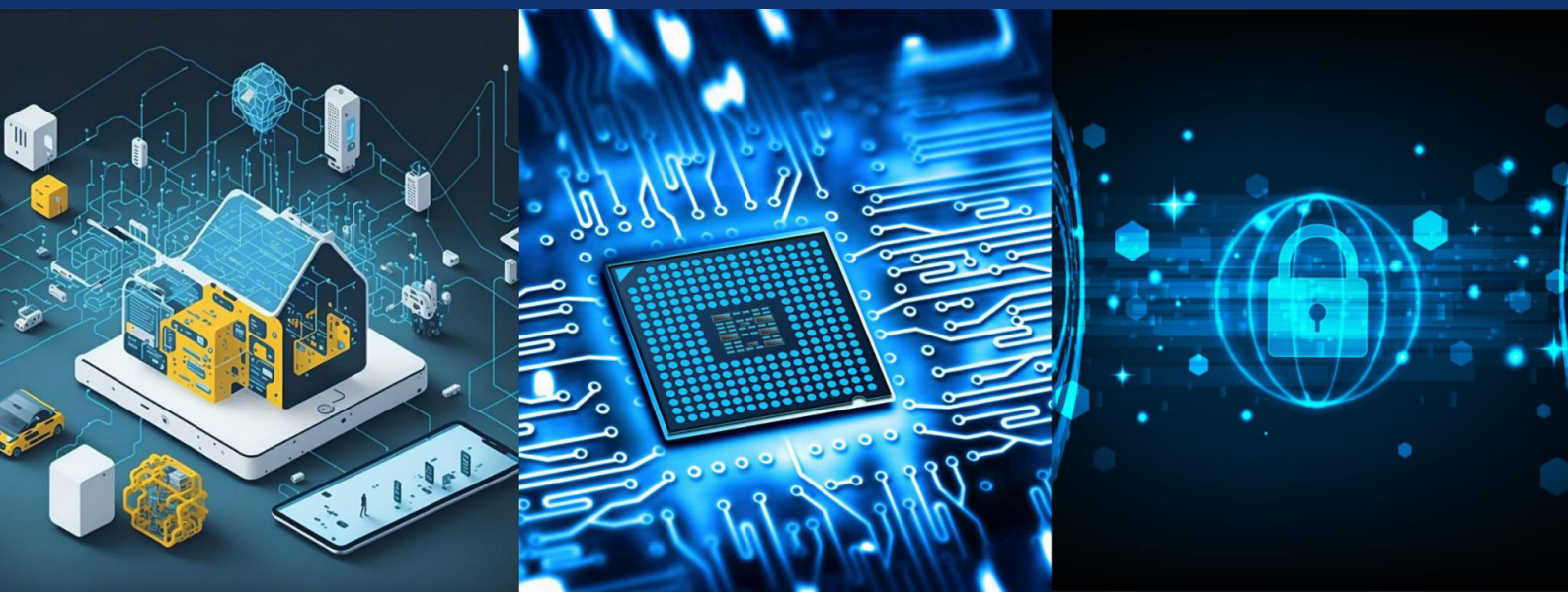
IJIRCCE

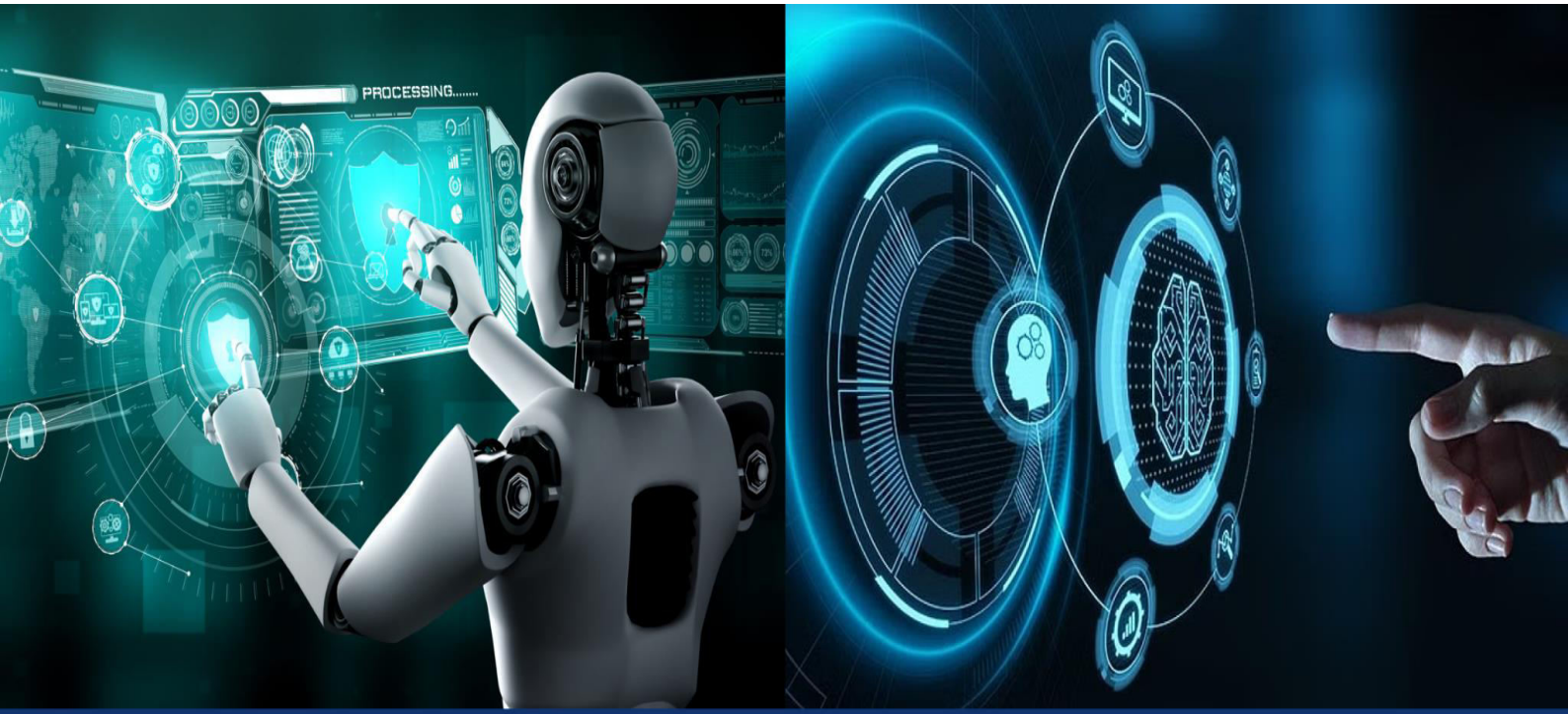
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 10, October 2026

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410023

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Blockchain-Based Tamper Proof Digital Evidence Locker

Dr. Murgesh V Jambigi[1], **Hemanth V Makanur**[2], **Mahabubjilani Sharifsab Ritti**[3], **Prajwal C S**[4], **Sumanth Shivayogeppa Banakar**[5]

Associate Professor, Department of Computer Science and Engineering, Sri Taralabalu Jagadguru Institute of Technology (S.T.J.I.T), Ranebennur, Karnataka, India[1]

U.G. Student, Department of Computer Science and Engineering, Sri Taralabalu Jagadguru Institute of Technology (S.T.J.I.T), Ranebennur, Karnataka, India[2–5]

ABSTRACT: Digital evidence such as photographs, videos and sensor data now underpins most criminal investigations, yet it is usually stored and shared through channels that cannot prove it has remained unaltered. This paper presents a Blockchain-Based Tamper-Proof Digital Evidence Locker that protects the integrity of crime scene evidence. For every uploaded file the system computes a SHA-256 hash, stores the file in server storage, keeps case metadata in MongoDB, and anchors the hash permanently on an Ethereum blockchain through a Solidity smart contract. During verification the hash of a submitted file is recomputed and compared with the blockchain copy, and the file is declared Original or Tampered. JWT authentication, role-based access control and a complete audit log support the chain of custody. Seventeen functional test cases covering authentication, upload validation, tamper detection, access control and blockchain-offline behaviour were all passed, and a file altered by a single byte was correctly flagged as tampered. The system offers a practical, low-cost alternative to centralized evidence handling.

KEYWORDS: Digital evidence, Blockchain, SHA-256, Smart contract, Chain of custody, Tamper detection, Ethereum.

I. INTRODUCTION

Crime scene investigation depends on the collection, preservation, analysis and presentation of evidence that can decide the outcome of legal proceedings. Most of this evidence is now digital: photographs, videos, Internet of Things (IoT) sensor readings and other data. Digital evidence helps reconstruct events, identify suspects and support arguments in court, but its growing use raises serious questions about security, authenticity and integrity.

In conventional practice, evidence moves between investigators through email, cloud storage or portable media. These channels offer little protection against unauthorized access, modification or loss, and there is often no reliable way to show that a file has not been altered in transit or storage. Even a minor modification can damage the credibility of evidence. A second problem is the chain of custody, the chronological record of how evidence is collected, handled, stored and transferred. If this record cannot be trusted, evidence may be rejected by a court or, worse, lead to a wrongful conviction.

Digital watermarking and basic cryptography have been used to address these issues, but watermarks can be removed or altered, and encryption does not reveal tampering once data has been decrypted. Neither provides a transparent audit trail. Blockchain, a distributed ledger in which time-stamped records are cryptographically linked and cannot be changed without detection [1], [2], is a natural fit for this problem. Combined with a cryptographic hash such as SHA-256 [3], which changes completely when even one bit of a file changes, it allows any alteration to be detected.

This paper presents a web-based Tamper-Proof Crime Scene Evidence Locker in which evidence files are hashed, the hash is recorded on the blockchain, and any later copy of the file can be verified against that record in seconds. The contributions are: (i) a hybrid architecture that keeps bulky files off-chain and only their fingerprints on-chain; (ii) a verification workflow that reports Original or Tampered; (iii) role-based access and audit logging for a traceable chain of custody; and (iv) functional testing of the working prototype.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE SURVEY

Rani et al. [4] propose an evidence preservation system for IoT-enabled smart environments using IPFS, Ethereum blockchain and smart contracts. Smart contracts automate authentication, access control and verification, a two-level process validates both users and uploaded content, and an immutable chain of custody is maintained on the ledger. The decentralized design removes the single point of failure of centralized storage.

Senkyire and Kester [5] validate forensic crime scene images by combining fragile digital watermarking with a cryptographic blockchain. A watermark is embedded at capture, the image hash is stored on the blockchain, and authenticity is later checked by watermark extraction and hash comparison, so that even small edits are detected.

Verma et al. [6] discuss blockchain together with cloud computing for preserving crime scene evidence. Their framework includes participants such as investigators, courts and legal authorities, a front-end, core storage and processing modules, a blockchain network and distributed storage, and emphasizes a digital, immutable chain of custody.

These studies confirm the value of blockchain for evidence integrity, but they are largely conceptual or focus on a single aspect (IoT data, images or cloud storage). The present work complements them with a complete, implemented prototype that includes user management, audit logging and a tested verification workflow.

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing system. Current practice relies on centralized storage or simple communication channels. Its main limitations are weak security and encryption, no reliable detection of post-capture alteration, a single point of failure at the central server, outdated techniques (basic cryptography and watermarking) that can be bypassed, no transparent tracking of data access, and poor handling of large files.

B. Proposed system. The proposed system is a secure platform for managing and sharing crime scene evidence using blockchain and hashing. Evidence such as images and videos is captured, a unique SHA-256 digest is generated for each file, the file is stored in server storage and its metadata in a database, and the digest is recorded on the blockchain, which guarantees immutability and transparency. When an investigator later retrieves or receives a file, the digest is recomputed and compared with the recorded value; any modification, however small, produces a different hash and is detected instantly.

C. Objectives. (1) To store crime scene evidence so that it cannot be altered or deleted without detection; (2) to generate and store a unique cryptographic hash for every evidence file so modifications are easily identified; and (3) to reduce manual effort and human error in storing and verifying evidence.

IV. SYSTEM ARCHITECTURE

The system follows a hybrid on-chain/off-chain design, shown in Fig. 1. A police officer or administrator interacts with an HTML/CSS/JavaScript front-end. Requests are handled by a Node.js (Express) back-end, which computes the SHA-256 hash, stores the evidence file in local/server storage under a random UUID file name, saves metadata (file name, hash, file type, upload date and time, uploader) in MongoDB for quick retrieval, and sends the hash to a Solidity smart contract running on a local Ethereum network (Ganache). Only the hash is placed on the blockchain, which keeps storage costs low while retaining immutability.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410023

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

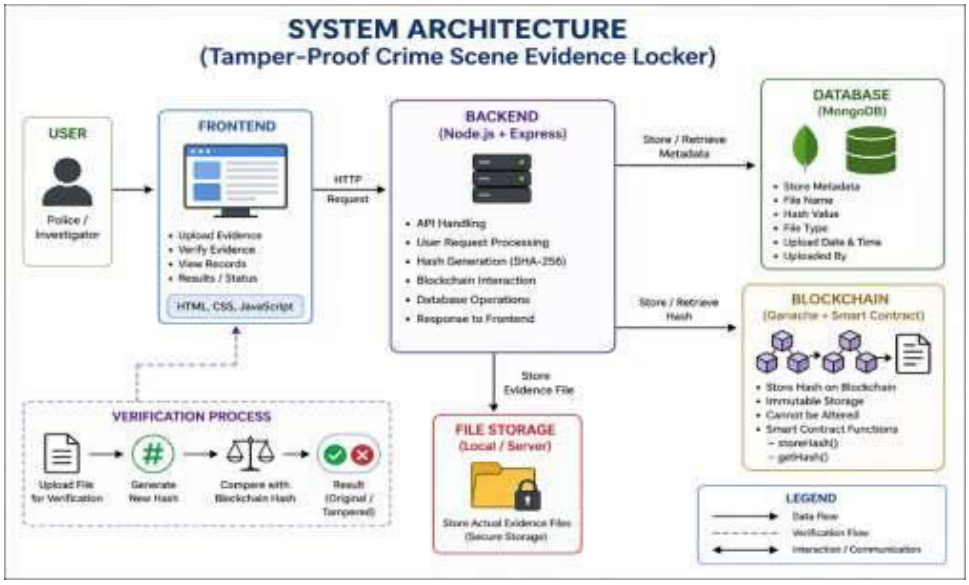


Fig. 1. System architecture of the Tamper-Proof Crime Scene Evidence Locker

The functionality is organized into eight modules: (1) authentication, issuing JWT tokens [7] with role-based access for officers and administrators; (2) evidence upload; (3) hash generation, which streams the file so that large files need little memory; (4) MongoDB metadata management; (5) blockchain service, connecting the back-end to Ganache through Web3.js to store, retrieve and verify hashes; (6) evidence verification; (7) search, history and download with pagination; and (8) administration and audit reporting, including user management, CSV export and audit logs.

Data model and actors. Three main entities are maintained. A User has a username, full name, hashed password, role (admin or officer) and active flag. An Evidence record holds the case number, case title, crime type, officer name, SHA-256 hash, blockchain transaction hash, status (stored, verified or tampered), uploader, upload date and last-verified time. An Audit Log entry records the action, the user who performed it, the case number and details. The two actors are the police officer, who logs in, uploads, searches, downloads and verifies evidence, and the administrator, who additionally manages users, evidence records, blockchain status, audit logs and metadata reports.

Smart contract. The Solidity contract exposes three functions: storeHash(caseNumber, hash) records the digest of a case and returns a transaction hash, getHash(caseNumber) returns the stored digest, and verifyHash(caseNumber, newHash) compares a recomputed digest with the stored one and emits a HashVerified event. Since a recorded value cannot be edited, an attacker who alters a file in storage cannot make the altered file match the on-chain digest.

V. METHODOLOGY AND IMPLEMENTATION

A. Upload algorithm. The officer selects a file and enters the case details, after which the system performs the following steps:

1. Validate the file type (jpg, png, pdf, mp4, zip) and size limit, and store the file temporarily.
2. Compute the 64-character SHA-256 hash of the file, a fingerprint unique to that exact content.
3. Generate a UUID, move the file into the evidence folder and discard the temporary copy.
4. Call storeHash(caseNumber, hash) on the smart contract and receive the transaction hash.
5. Save the case details, hash and transaction hash in MongoDB, write an audit-log entry and show a confirmation.

B. Verification algorithm. To verify a file, the officer submits the file and its case number:

1. Recompute the SHA-256 hash of the submitted file.
2. Retrieve the original hash for that case from the blockchain through verifyHash(caseNumber, newHash), which compares the values and emits a HashVerified event.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410023

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. If both hashes are identical the result is Original; otherwise it is Tampered. If the blockchain is unreachable, the back-end falls back to the hash held in MongoDB.
4. Update the record status, write an audit-log entry and display a green (Original) or red (Tampered) result card.

C. Security and audit. Access is protected by JWT-based login. Officers can upload, verify, search and download evidence, whereas administrators can additionally manage users, delete records, export metadata as CSV and read audit logs. Every login, upload, verification, download and deletion is logged with user, case number and time, which supports the chain of custody. The functions are exposed through a REST API (for example POST /api/login, POST /api/upload, POST /api/verify, GET /api/evidence and GET /api/download/:id, with administrator-only endpoints for deletion, export, user management and audit logs).

D. Environment. The prototype runs on Windows 10/11 on an Intel Core i3/i5 machine with 4 GB RAM (8 GB recommended) and 10 GB free storage. It uses Visual Studio Code, HTML/CSS/JavaScript for the front-end, Node.js for the back-end, MongoDB for metadata, and Ganache with Truffle and Solidity for the blockchain layer.

VI. RESULTS AND DISCUSSION

The prototype provides a landing page, login, dashboard, upload page, verification page, evidence history and details views, and an admin panel with user management, evidence management, audit logs and blockchain status. When evidence is uploaded, its SHA-256 hash and the blockchain transaction hash are displayed in the evidence details view, and the record is marked Verified after a successful check.

Fig. 2 shows the two verification outcomes. When the original image is re-submitted, the newly computed hash equals the stored hash and the system displays “Original – Evidence Verified”. When a different file is submitted for the same case, the computed hash differs from the stored hash and the system displays “Tampered – Hash Mismatch Detected”. The audit log records each of these verifications with the user, case number and time.

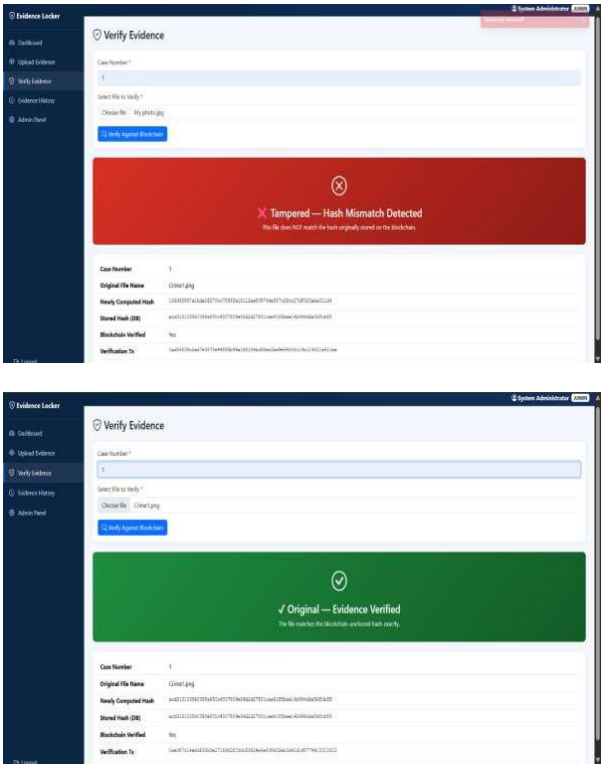


Fig. 2. Verification result: tampered (top) and original (bottom)

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410023

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Seventeen test cases (TC01–TC17) were executed; their grouped outcomes are summarized in Table I. All cases produced the expected result. In particular, a file with a single byte changed was reported as Tampered, unauthorized and tokenless requests were rejected with 403 and 401 responses, disallowed or oversized files were refused, and when Ganache was stopped the system still saved the file and metadata, showed a warning and left the transaction field empty so that the record could be identified as not yet anchored.

TABLE I. SUMMARY OF FUNCTIONAL TEST RESULTS

Group	Test cases	Verified behaviour	Result
Authentication	TC01–TC03	JWT issued for valid login; wrong password rejected; call without token returns 401	3/3 Pass
Upload validation	TC04–TC07	Valid upload returns file hash and transaction hash; .exe, oversize and missing-field uploads rejected	4/4 Pass
Tamper detection	TC08–TC10	Unchanged file → Original; file with one byte changed → Tampered; unknown case → 404	3/3 Pass
Search and download	TC11–TC13	Partial case-number search, pagination (>10 records), original file download	3/3 Pass
Access control and admin	TC14–TC16	Officer blocked from admin API (403); admin delete with audit entry; CSV export	3/3 Pass
Fault tolerance	TC17	Ganache stopped: file and metadata saved, warning shown, transaction field empty	1/1 Pass

The results show that hash anchoring gives a simple and dependable integrity check: the cost of verification is one hash computation and one contract call, so the result appears within seconds for the tested files. The off-chain/on-chain split keeps the blockchain small, and the audit log gives investigators a traceable record of every action. Because the evaluation is functional, performance with very large files and large numbers of concurrent users has not been benchmarked and is left for future work.

The upload page (Fig. 3) collects the case number, title, crime type, officer, police station, date and description together with the file, and the audit log (Fig. 4) lists each login, upload and verification result in time order. Table II contrasts the proposed locker with conventional evidence handling.

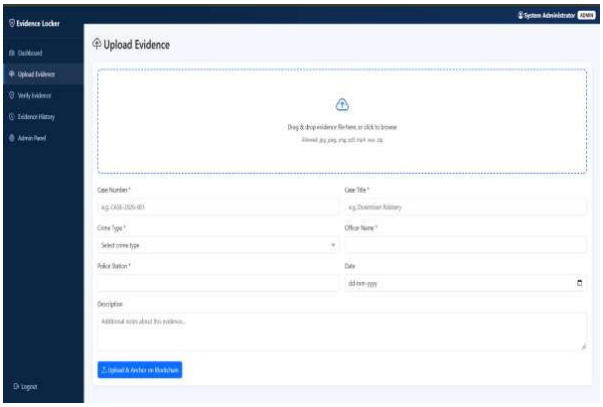


Fig. 3. Evidence upload page

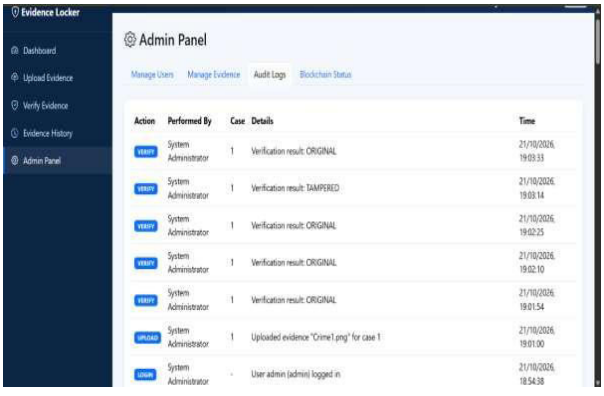


Fig. 4. Audit log of administrator actions

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE II. CONVENTIONAL EVIDENCE HANDLING VERSUS THE PROPOSED LOCKER

Aspect	Conventional system	Proposed locker
Integrity check	None or manual; alteration may go unnoticed	SHA-256 hash compared with blockchain copy; any change detected
Record keeping	Mutable centralized database	Immutable ledger entry per evidence file
Single point of failure	Central server	Hash anchored on distributed ledger
Chain of custody	Paper or ad-hoc logs	Audit log of every login, upload, verification, download, deletion
Access control	Shared folders or email	JWT login with officer and admin roles

VII. CONCLUSION AND FUTURE SCOPE

This paper presented a working Tamper-Proof Crime Scene Evidence Locker that combines a web application, a database, a blockchain and secure file storage. SHA-256 hashing gives every file a unique fingerprint, the smart contract preserves that fingerprint permanently, and the verification process shows within seconds whether a file is original or tampered. Role-based access and a complete audit log add accountability and support a reliable chain of custody, so the system meets the stated objectives and offers a practical alternative to insecure, centralized evidence handling.

Future enhancements include storing evidence files in IPFS [8] with the content identifier recorded on-chain, deploying the contract on a public or consortium Ethereum network, keying the contract by a unique evidence ID so that several files can belong to one case, encrypting files at rest and adding two-factor authentication, recording chain-of-custody transfers between officers as on-chain events, adding digital watermarking and more evidence formats, and developing a mobile application for evidence capture at the scene.

VIII. ACKNOWLEDGEMENT

The authors thank the Department of Computer Science and Engineering, Sri Taralabalu Jagadguru Institute of Technology, Ranebennur, for the facilities and support provided during this work.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[2] V. Buterin, “A next-generation smart contract and decentralized application platform (Ethereum white paper),” 2014.
[3] National Institute of Standards and Technology, “Secure Hash Standard (SHS),” FIPS PUB 180-4, Aug. 2015.
[4] D. Rani, N. S. Gill, P. Gulia, M. Yahya, T. A. Ahanger, M. M. Hassan, F. B. Abdallah and P. K. Shukla, “A secure digital evidence preservation system for an IoT-enabled smart environment using IPFS, blockchain, and smart contracts,” Peer-to-Peer Networking and Applications, vol. 18, no. 2, p. 5, Dec. 2024. doi: 10.1007/s12083-024-01855-z.
[5] I. B. Senkyire and Q.-A. Kester, “Validation of forensic crime scene images using watermarking and cryptographic blockchain,” in Proc. 2019 Int. Conf. Computer, Data Science and Applications (ICDSA), 2019, pp. 1–4.
[6] S. Verma, A. Kumar, S. Pandey and P. Negi, “Blockchain and cloud computing used in preservation of crime scene evidences,” in Proc. 2023 2nd Int. Conf. Edge Computing and Applications (ICECAA), 2023, pp. 7–11.
[7] M. Jones, J. Bradley and N. Sakimura, “JSON Web Token (JWT),” RFC 7519, Internet Engineering Task Force, May 2015.
[8] J. Benet, “IPFS – Content addressed, versioned, P2P file system,” arXiv:1407.3561, 2014.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details