

Decentralized Video Provenance at the Edge: Blockchain-Assisted Integrity for Security Camera Streams

Marco Vela-Koentarjo
The Pennsylvania State University
University Park, Pennsylvania, United States

Madhusudhan Singh
The Pennsylvania State University
University Park, Pennsylvania, United State

Abstract

With the rapid proliferation of generative AI, deepfakes and manipulated media pose increasing risks to political, economic, and social trust. The combination of accelerating developments in artificial intelligence and a lack of digital literacy creates an environment where malicious fake content can thrive. This work proposes a blockchain-based system for tamper evident registration and decentralized storage of security video footage. Unlike full camera-capture provenance, which guarantees end-to-end traceability from sensor to storage, our approach focuses on detectable post-capture tampering, providing verifiable logging of footage while defining clear threat and trust boundaries. By leveraging the Binance Smart Chain (BSC) for cryptographic logging and IPFS for decentralized storage, the system creates a foundation to mitigate the impact of AI manipulation in security and compliance applications. We evaluated the system by developing a functional prototype and conducting stress tests to assess cost efficiency and latency. Our analysis demonstrates that while the system provides robust integrity guarantees, decentralized storage introduces the primary latency bottleneck, with IPFS uploads averaging 2.08 seconds per segment. Furthermore, cost analysis indicates a baseline transaction fee of approximately \$0.57 USD, which is viable for high-value evidence but requires Layer 2 scaling strategies for continuous streaming. Ultimately, this paper demonstrates that tamper-evident, blockchain-backed video registration is a technically feasible alternative to centralized custody logs, provided that scalability and storage considerations, such as decentralized pinning or permanent storage layers, are addressed.

CCS Concepts

• **Applied computing** → *Surveillance mechanisms; Evidence collection, storage and analysis.*

Keywords

Blockchain, Deepfakes, Media Verification, Security, Cryptography, IPFS, Smart Contracts

ACM Reference Format:

Marco Vela-Koentarjo and Madhusudhan Singh. 2026. Decentralized Video Provenance at the Edge: Blockchain-Assisted Integrity for Security Camera Streams. In *Companion Proceedings of the ACM Web Conference 2026 (WWW Companion '26)*, April 13–17, 2026, Dubai, United Arab Emirates. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3774905.3794702>



This work is licensed under a Creative Commons Attribution 4.0 International License. *WWW Companion '26, Dubai, United Arab Emirates.*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2308-7/2026/04

<https://doi.org/10.1145/3774905.3794702>

1 Introduction

Recent advances in generative artificial intelligence have fundamentally altered the world. As detailed in a comprehensive survey by Pei et al. [2], tools capable of creating hyper-realistic video and audio are no longer exclusive to well-funded actors; they are now accessible to the general public via consumer-grade hardware and cloud APIs. This technological shift has ushered in the era of the "deepfake," where synthetic media can mimic human appearance, speech, and mannerisms with visual artifacts that are increasingly imperceptible to the human eye.

The implications of this shift extend far beyond misinformation on social media. In the context of law enforcement and judicial proceedings, the integrity of video evidence is paramount. Chesney and Citron [1] argue that the existence of high-quality deepfakes creates a phenomenon known as the "Liar's Dividend," where malicious actors can plausibly deny the authenticity of genuine incriminating footage by claiming it was AI-generated. Consequently, the burden of proof has shifted to cryptographically proving the origin and history of video evidence.

Current methodologies for establishing this proof are increasingly inadequate. Traditional video surveillance systems rely on centralized architectures such as on-premise Network Video Recorders (NVRs) and cloud-based Video Management Systems (VMS). These architectures introduce critical points of failure. In 2023, the Cybersecurity and Infrastructure Security Agency (CISA) identified remote code execution flaws in widely used security camera gateways [4], demonstrating that a bad actor with administrative privileges can alter footage. Furthermore, the "Chain of Custody" (CoC) in these systems is often maintained via mutable database entries, which Chopade and Khan [3] argue fail to provide the security required in a zero-trust environment.

To address systemic vulnerabilities in video evidence integrity, we consider a threat model in which an adversary may attempt to tamper with or falsify security footage after capture. The system assumes that edge devices (cameras or local nodes) are trusted to capture footage correctly, while the network, storage infrastructure, and observers may be untrusted. Within this trust boundary, the project proposes a decentralized architecture for security footage verification. Moving beyond the "arms race" of post-hoc deepfake detection, which often lags behind generation capabilities [2], our system emphasizes proactive, tamper-evident provenance. We leverage the BSC Testnet and the InterPlanetary File System (IPFS) to anchor video data to an immutable public ledger at the point of capture. By replacing centralized trust with cryptographic consensus, we ensure that digital evidence remains verifiably tamper-evident. The proposed methodology is illustrated in Figure 1.

The remainder of this article is organized as follows: Section 2 is a literature review and positions our contributions within the

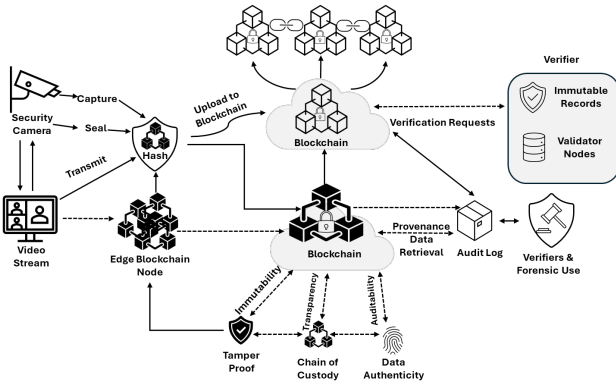


Figure 1: An Overview of the Proposed Idea: Decentralized Video Provenance at the Edge: Blockchain-Assisted Integrity for Security Camera Streams.

current state of the art. Section 3 presents the methodology of proposed Decentralized Video Provenance at the Edge architecture. Section 4 describes the simulation design and experimental results. Section 5 discusses the implications and future research directions.

2 Literature Review

Considerable research has focused on post-hoc AI detection systems. A recent trend is zero-shot detection, which attempts to identify deepfakes without prior exposure to their specific generation method. Sar et al. [11] examine this approach and highlight significant limitations, particularly regarding cross-modal synthesis. Modern deepfake systems can perform highly accurate lip-syncing using AI-generated speech; the resulting video, in which human lip movements align convincingly with synthetic speech, is often indistinguishable by the human eye. Sar et al. note that cross-modal inconsistencies are often poorly captured by zero-shot models relying on semantic embeddings, resulting in frequent misclassification. This highlights the fragility of relying solely on detection models that must constantly race against generative adversaries.

Neekhara et al. [8] introduce FaceSigns, a semi-fragile neural watermarking scheme designed to authenticate images. However, their threat-model analysis identifies key vulnerabilities inherent to watermarking. Attackers may query the decoder network to iteratively perturb an image until the decoded message matches a target value ("oracle attacks"). Furthermore, they examine perturbation-copy attacks, where adversaries extract the watermark noise from a valid asset and transfer it to a deepfake. While FaceSigns employs image-specific perturbations to mitigate this, Qureshi and Megias [9] argue that watermarking approaches generally depend on the computational difficulty of removing the mark, rather than establishing an independent, tamper-evident history of the asset, suggesting that cryptographic provenance provides a stronger guarantee of integrity.

To address the limitations of detection and watermarking, researchers have turned to blockchain technology to establish an

immutable Chain of Custody (CoC). Gipp et al. [6] pioneered "Decentralized Trusted Timestamping," arguing that Proof-of-Work mechanisms act as a decentralized notary.

However, timestamping alone does not prove the origin of high-bandwidth media streams. Hasan et al. [7] established the foundational requirements for secure provenance, proposing the use of iterative cryptographic hashing (hash chains) to detect video tampering. In this model, any modification to a video frame breaks the chain of hashes.

Furthermore, provenance requires strict device identity management to prevent 'masquerading' attacks. Cabrera et al. [5] emphasize that software-based security is insufficient for edge nodes in hostile environments. They argue that to prevent key spoofing, valid sensor data must be signed by keys generated and held within Hardware Security Modules (HSMs) or Trusted Execution Environments (TEEs), ensuring that private keys are never exposed to the host operating system.

A critical limitation in blockchain-physical interfaces is the "Oracle Problem", or the risk that a compromised edge device feeds fabricated data to the immutable ledger. Nikouhei et al. [?] note that as AI surveillance moves to the Edge, compromised models could generate false flags. To mitigate software-level tampering, Sabt et al. [10] propose the use of TEEs such as the ARM TrustZone.

Finally, even perfectly secured hardware faces "analog replay" or "presentation attacks," where an attacker presents a screen displaying fake footage to the camera sensor. Wang et al. [12] demonstrate that standard 2D analysis is insufficient to detect high-resolution screen spoofing. They propose the integration of depth information (via kinetic sensors or structured light) to analyze the 3D structure of the scene. By correlating depth maps with visual data, the system can distinguish between a flat screen (2D) and a real environment (3D), a necessary hardware requirement for certifying the "liveness" of security footage.

3 Methodology

This paper evaluates the feasibility of a decentralized Chain of Custody for video streams. The literature review identifies that a production deployment requires specialized edge hardware, such as Trusted Execution Environments and depth sensors, to mitigate physical tampering. However, acquiring and programming proprietary camera firmware is outside the scope of this study.

Consequently, this methodology adopts a software simulation approach. We implemented a browser-based recording interface that acts as a functional proxy for a smart security camera. This abstraction allows us to prove the efficacy of the security model against tampering and masquerading attacks. Physical tamper resistance remains a hardware-dependent variable for future industrial iterations.

The prototype architecture follows a three-tier design pattern as shown in Figure 2:

- **Presentation Layer (Client):** Built with Next.js, this layer interfaces with the user's webcam. It handles video capture, local SHA-256 hashing, and interactions with the Web3 provider. We implement an EIP-712 signing flow, allowing the camera hardware to sign evidence without holding gas tokens.

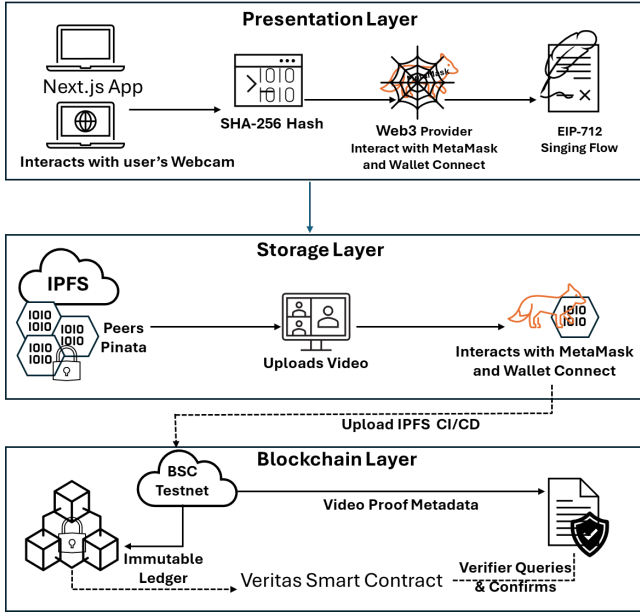


Figure 2: System architecture of the proposed prototype layers.

- **Storage Layer (IPFS):** Storing raw video data on a blockchain is prohibitively expensive due to gas costs. We utilize IPFS, a content-addressable peer-to-peer storage network. For this experimental prototype, we employ **Pinata** as a pinning service to ensure data persistence and high availability during testing.
- **Verification Layer (Blockchain):** The BSC Testnet serves as the immutable ledger. The smart contract stores the link between the video's content hash, its IPFS CID, the uploader's address, and the block timestamp.

3.1 Smart Contract Design

A key innovation in this architecture is the use of EIP-712 typed data signatures. Security cameras are often IoT devices that lack the resources to manage cryptocurrency wallets or pay for gas fees. To solve this, the contract implements a *Relayer pattern* via the `registerVideoSigned` function.

The Message Schema (Definition). Before a camera can sign data, the contract must define a Type Hash. This acts as a structured template (or schema). It is important to distinguish this definition from the signature itself as `VIDEO_TYPEHASH` simply tells the blockchain how to read the data the camera has signed:

```
bytes32 public constant VIDEO_TYPEHASH = keccak256(
    Video(bytes32 contentHash,
        uint256 originalTimestamp,
        uint64 sequence,
        string cid)
);
```

In this workflow, the camera signs a hash of the video metadata off-chain. This signature is sent to a Relayer (a backend service), which submits the transaction and pays the gas fees. To ensure

the evidence is attributed to the camera and not the relayer, the contract must "recover" the signer's identity. This is handled by the `_recoverSigner` function, which splits the 65-byte signature into its cryptographic components (r , s , and v) and uses the EVM's `ecrecover` opcode:

```
function _recoverSigner(bytes32 digest,
    bytes memory signature)
    internal pure returns (address) {
    if (signature.length != 65) return address(0);
    bytes32 r; bytes32 s; uint8 v;
    assembly {
        r := mload(add(signature, 32))
        s := mload(add(signature, 64))
        v := byte(0, mload(add(signature, 96)))
    }
    return ecrecover(digest, v, r, s);
}
```

3.2 Cryptographic Workflow and Data Flow

The proposed system ensures data integrity and tamper-evident provenance through the workflow depicted in Figure 3. Video is captured in 30-second segments at 360p (640×360) using the WebM format. These parameters were selected for prototype testing to approximate edge device conditions. Importantly, only the fixed-length hash and metadata are submitted on-chain, so blockchain transaction cost is independent of video size, duration, or resolution. Each segment is hashed using SHA-256, and an EIP-712 signature is generated over the hash. To prevent replay attacks, a monotonically increasing sequence number is included; a timestamp is also recorded to support retrieval and auditing, but alone it cannot prevent replay, as an attacker could reuse an old transaction with the same timestamp. The video file is then uploaded to an IPFS pinning service (Pinata), while encrypted data is temporarily stored in a local queue if connectivity is disrupted. In the current demo, this queue resides in the browser, but in a full deployment it could be stored locally on an edge device, limited only by available storage. Finally, the Relayer submits the signature and metadata to the smart contract, which validates the sequence number and signature before committing the record to the ledger, ensuring end-to-end cryptographic attestation.

4 Implementation and Results Details

To validate the architectural viability of the proposed system, we conducted a series of performance tests focusing on latency, throughput, and cost-efficiency. The primary goal was to determine if a decentralized architecture could meet the near-real-time requirements of security surveillance.

4.1 Experimental Setup

All performance benchmarks were conducted on a MacBook Pro (Apple M4 Pro Chip, 24GB Unified Memory) connected via a low-latency broadband connection (81 Mbps Upload).

Video segments were captured at 640×360 resolution (360p, 15fps) in 30-second intervals using the VP9/WebM codec. The segment size averaged 900 KB. Blockchain interactions utilized the BSC Testnet with a gas price of 3 Gwei. IPFS storage persistence was handled through via Pinata.

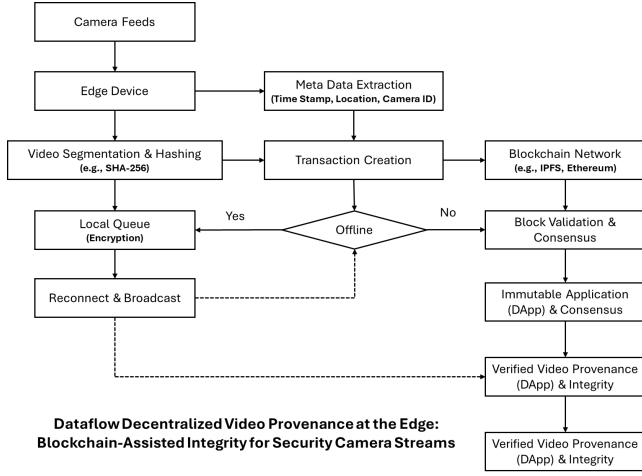


Figure 3: Data flow of proposed methodology.

Table 1: System Latency Metrics ($N = 10$) on M4 Pro

Metric	Mean (ms)	Std Dev (σ)	Min (ms)	Max (ms)
Hashing (SHA-256)	162	± 28	136	220
IPFS Upload	2,088	± 145	1,950	2,773
EIP-712 Signing	1,940	± 210	1,768	2,316
Relayer Dispatch	615	± 115	469	740
Block Consensus*	< 3,000	—	< 1,000	3,000

*Verified via blockchain timestamps; excludes RPC indexing lag.

While benchmarks were collected on an Apple M4 Pro, deployment on edge devices like a Raspberry Pi 4 or Jetson Nano would incur higher latency due to limited CPU performance and memory bandwidth. Compute-intensive tasks such as hashing and signing would take noticeably longer, and IPFS uploads could be slowed by constrained networking. Despite this, the architecture remains feasible on embedded hardware, though practical deployments may require lightweight optimizations or offloading to capable relayers.

4.2 Results

The system captures footage in 30-second segments and completes full automated processing in around 4.8 seconds each segment. This encompasses relayer execution, IPFS storage, and cryptographic hashing. This indicates that merely 16% of each time interval is utilized, providing ample capacity to accommodate network jitter or brief bandwidth fluctuations. The additional capacity enables the system to manage elevated video resolutions, such as 720p, without inducing delays or backlogs. This demonstrates that the pipeline remains reliable and responsive in practical scenarios. We also examined the costs associated with registering video evidence on the blockchain to assess its long-term viability. A single video registration used an average of 228,466 gas units on the BSC Testnet. This expense is applicable to mainnet configurations due to the predictability of gas consumption in EVM-based systems. On December 26, 2025, the gas price is 3 Gwei, and the BNB price is \$835. The expense per segment is approximately \$0.57. This is reasonable for occasional submission of high-value evidence; but, it becomes

costly for continuous streaming, with anchoring every 30 seconds amounting to around \$68.40 per hour.

$$\text{Cost} = 228,466 \text{ (Gas)} \times 3 \cdot 10^{-9} \text{ (BNB)} \times \$835 \approx \$0.57 \text{ USD} \quad (1)$$

There are two primary methods to enhance the system for large-scale and continuous deployments. Initially, several video segment hashes can be amalgamated into a singular Merkle tree, with solely the Merkle root recorded on-chain. The cost per segment decreases to approximately \$0.029 when there are twenty segments in a batch. Secondly, utilizing the registry contract on Ethereum Layer-2 rollups such as Arbitrum or Optimism can reduce transaction expenses by one to two orders of magnitude. Real-world rollup deployments demonstrate that good data management can reduce expenses to less than one cent, making continuous video evidence anchoring both scalable and economical. Table 2 summarizes the projected cost reductions for continuous streaming scenarios under these proposed architectures.

Table 2 summarizes the projected cost reductions for continuous streaming scenarios under these proposed architectures.

Table 2: Projected Cost Reduction Strategies

Architecture Strategy	Factor	Cost / Hour (USD)
Current (Baseline)	1.00	68.40
Merkle Batching ($N=20$)	0.05	3.42
Layer 2 Migration	0.001–0.01	0.07–0.70

5 Conclusion

This project demonstrates a functional proof-of-concept for a blockchain-backed system for video provenance. By integrating a Next.js frontend, IPFS storage, and a BSC smart contract, the prototype establishes that video content can be verified in a tamper-evident manner, ensuring that the Chain of Custody remains unbroken from upload to verification.

Future work will focus on strengthening the infrastructure for production scalability and decentralization. To address the economic and persistence challenges of storing petabytes of footage, the architecture will evolve from centralized IPFS pinning services to incentivized permanent storage layers. Additionally, while the current system emphasizes integrity, future iterations will investigate privacy-preserving techniques, such as zero-knowledge proofs, to enable compliance with regulations like GDPR, particularly when handling public surveillance data.

Finally, a major objective is to transition from browser-based capture to dedicated edge hardware. This includes the integration of Hardware Security Modules (HSMs) to manage private keys within a Trusted Execution Environment. To mitigate "analog gap" vulnerabilities—such as screen spoofing—the system will incorporate depth sensing (LiDAR) and motion analysis components directly into the camera firmware. Together, these extensions will transform the current prototype into a robust, enterprise-grade infrastructure for authenticated video surveillance.

References

- [1] Chesney, R., & Citron, D. (2019). Deep fakes: A looming challenge for privacy, democracy, and national security. *California Law Review*, 107, 1753.
- [2] Pei, G., Zhang, J., et al. (2024). Deepfake Generation and Detection: A Benchmark and Survey. *arXiv preprint arXiv:2403.17881*.
- [3] Chopade, R., & Khan, R. A. (2024). Digital Forensics: Maintaining Chain of Custody Using Blockchain. *Proceedings of the 2024 IEEE International Conference on Trust, Privacy and Security in Intelligent Systems (TPS-ISA)*, 11-19.
- [4] Cybersecurity and Infrastructure Security Agency (CISA). (2023). *Top Routinely Exploited Vulnerabilities of 2023*. U.S. Department of Homeland Security. Retrieved from cisa.gov.
- [5] Cabrera-Gutierrez, A. J., Castillo, E., Escobar-Molero, A., & Cruz-Cozar, J. (2023). Secure Sensor Prototype Using Hardware Security Modules and Trusted Execution Environments in a Blockchain Application: Wine Logistic Use Case. *Electronics*, 12(13), 2987.
- [6] Gipp, B., Meuschke, N., & Gernandt, A. (2015). Decentralized Trusted Timestamping using the Bitcoin Blockchain. *Proceedings of the iConference 2015*.
- [7] Hasan, R., Sion, R., & Winslett, M. (2009). The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance. *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST '09)*, 1-14.
- [8] Neekhara, P., Hussain, S., Zhang, X., Huang, K., McAuley, J., & Koushanfar, F. (2022). FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes. *arXiv preprint arXiv:2204.01960*.
- [9] Qureshi, A., & Megias, D. (2021). Blockchain-based multimedia content protection: Review and open challenges. *Applied Sciences*, 11(1), 1-22.
- [10] Sabt, M., Achemlal, M., & Bouabdallah, A. (2015). Trusted Execution Environment: What It Is, and What It Is Not. *Proceedings of the 2015 IEEE Trust-com/BigDataSE/ISPA*, 1, 57-64.
- [11] Sar, A., Roy, S., Choudhury, T., & Abraham, A. (2025). Zero-Shot Visual Deepfake Detection: Can AI Predict and Prevent Fake Content Before it is Created? *Foundations and Trends in Signal Processing*, 19(3), 212–370.
- [12] Wang, Z., Zhao, C., Qin, Y., Zhou, Q., Qi, G., Wan, J., & Lei, Z. (2019). Exploiting temporal and depth information for multi-frame face anti-spoofing. *arXiv preprint arXiv:1811.05118*.
- [13] Nikouei, S. Y., Chen, Y., Song, S., Xu, R., Choi, B.-Y., & Faughnan, T. R. (2018). Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN. *arXiv preprint arXiv:1805.00330*.
- [14] Chervinski, J. O., Kreutz, D., Xu, X., & Yu, J. (2023). Analyzing the Performance of the Inter-Blockchain Communication Protocol. *arXiv preprint arXiv:2303.10844*.