Hardware-Anchored Media Provenance and Blockchain-Anchored Verification

Jose Luis Junior Pineda Fritas

Public Disclosure: January 13, 2026

Abstract

This document discloses a system and method for certifying the authenticity and provenance of digital media (images and video), including live streaming content, by anchoring cryptographic proofs generated at capture-time within trusted hardware to a public blockchain. The system enables third-party, trustless verification that a given photo, video, or stream originated from a physical capture device and has not been altered or manipulated.

1 Introduction

The rapid advancement of generative AI has made it increasingly difficult to distinguish authentic media captured by physical devices from synthetic or manipulated content. Traditional methods, such as watermarking, metadata verification, or AI-based detection, are often fragile, removable, or prone to false positives.

There is a need for a verifiable, device-level proof of capture that:

- Does not depend on analyzing the content itself,
- Is globally verifiable by any third party,
- Is resistant to tampering or post-processing,
- Works for both still images and live video streams.

2 Core Idea

The core concept is to bind each captured media asset to:

1. A trusted hardware identity of the capture device,
2. A cryptographic hash of the captured content,
3. A verifiable timestamp anchored to a public blockchain.

This creates an immutable, publicly verifiable record of media provenance. The economic value of the blockchain is irrelevant; only its immutability and verifiability are required.

3 Normative Requirements

The following requirements define the minimum conditions for a compliant implementation of the proposed system. The terms MUST, SHOULD, and MUST NOT are to be interpreted as described in RFC 2119.

3.1 Mandatory Requirements (MUST)

- Media capture events MUST generate a unique cryptographic hash at capture time.
- Hash generation MUST occur before any post-processing or user-applied modification.
- Hashes MUST be generated or sealed within a trusted hardware environment (TEE, Secure Enclave, or equivalent).
- Each capture event MUST include a verifiable timestamp resistant to rollback.
- Capture events MUST be recorded in an append-only, verifiable ledger.

3.2 Optional Requirements (SHOULD / MAY)

- Implementations SHOULD support anchoring hashes to a public blockchain.
- Video implementations MAY use hash chaining to link sequential frames.
- Batch anchoring MAY be used to reduce on-chain transaction overhead.
- Verification MAY be performed offline once ledger data is available.

3.3 Forbidden Behaviors (MUST NOT)

- Media MUST NOT be signed after content editing or recompression.
- Synthetic or AI-generated content MUST NOT be represented as captured media.
- User-level software MUST NOT override or forge capture events.
- The system MUST NOT assert factual truth or intent of recorded content.

4 System Components

4.1 Capture Device

The capture device (smartphone or high-end camera) should include:

- Secure hardware element (TEE, Secure Enclave, or TPM),
- Non-exportable private key bound to the device,
- Signed firmware and trusted capture pipeline.

4.2 Cryptographic Processing

At capture time:

1. Compute a cryptographic hash (e.g., SHA-256) of the image or video frame data.
2. Sign the hash within the secure hardware using the device's private key.

4.3 Blockchain Anchoring Layer

Only hashes and minimal metadata are written on-chain. Anchoring may occur asynchronously to avoid latency. Public blockchains or Layer-2 networks can be used for scalability.

4.4 Metadata Embedding

The signed hash and blockchain reference are embedded in media metadata. Metadata does not affect visual content and may be stripped without invalidating on-chain verification.

5 Smartphone Image and Video Capture Flow

1. User captures an image or video.
2. Device computes content hash inside trusted hardware.
3. Signed hash is anchored to the blockchain.
4. Media file includes reference to on-chain record.

Third parties can later:

- Recompute the hash,
- Verify the device signature,
- Verify blockchain inclusion.

6 Social Platform Verification

Social or media platforms may:

- Detect presence of provenance metadata,
- Verify the on-chain proof,
- Display a non-coercive visual marker, e.g., "Verified Capture Device."

Unverified content remains publishable but distinguishable.

7 Implementation Levels

7.1 Level 1 – Basic Capture Provenance

Provides basic proof of capture through content hashing and timestamping without strong hardware guarantees.

7.2 Level 2 – Hardware-Anchored Provenance

Introduces hardware root-of-trust, preventing software-level forgery of capture events.

7.3 Level 3 – Continuous Media Integrity

Extends provenance guarantees to video sequences using hash chaining and monotonic counters.

7.4 Level 4 – Live Attested Streaming

Provides real-time provenance guarantees for live streaming using pre-commitment and external time authorities.

8 Reference Implementation Considerations

This section provides non-normative but technically grounded guidance for implementing the proposed system on real-world capture devices. The intent is to demonstrate feasibility on existing hardware platforms and to clarify design trade-offs without prescribing a single implementation.

8.1 Trusted Capture Pipeline Integration

A compliant implementation SHOULD bind cryptographic operations as close as possible to the physical capture pipeline.

On modern smartphone SoCs, this implies:

- Hash computation triggered immediately after sensor readout or ISP output,
- Before any lossy compression, color grading, or user-visible post-processing,
- Within a Trusted Execution Environment (TEE) or Secure Enclave context.

Ideally, the capture pipeline follows the sequence:

Sensor → ISP → Trusted Hashing → Encoding → Storage

Any implementation that computes hashes after user-accessible processing stages weakens provenance guarantees and SHOULD be considered non-compliant with higher implementation levels.

8.2 Key Management and Device Identity

Each capture device MUST possess a non-exportable private key generated and stored within secure hardware. This key represents the cryptographic identity of the device, not of the user.

Recommended properties include:

- Hardware-bound key generation,
- No software-accessible export path,
- Support for attestation or certificate chains issued by the manufacturer or trusted authority.

Key rotation MAY be supported but MUST preserve continuity of trust through signed key transition records.

8.3 Timestamping and Anti-Rollback Guarantees

Timestamps associated with capture events SHOULD be resistant to rollback and manipulation.

Acceptable approaches include:

- Secure monotonic counters within trusted hardware,
- Cross-verification with network-based time authorities,
- Periodic anchoring of timestamps to an append-only public ledger.

Wall-clock timestamps provided by the operating system alone are insufficient for high-trust implementations.

8.4 Video and Continuous Media Handling

For video content, especially long recordings or live streams, per-frame anchoring is impractical.

A recommended approach is hash chaining:

- Each frame hash incorporates the previous frame hash,
- Periodic checkpoints are signed and optionally anchored on-chain,
- Any frame insertion, deletion, or reordering invalidates the chain.

This design enables post-hoc verification of full sequences while maintaining scalability.

8.5 Live Streaming Latency and Commitment Model

For live streaming, immediate on-chain anchoring of every segment introduces unacceptable latency.

A practical model consists of:

- An initial stream commitment anchored at stream start,
- Continuous local hash chaining during transmission,
- Periodic anchoring of intermediate commitments.

Verification MAY occur post-stream, while still guaranteeing that the stream content was generated continuously and without interruption.

8.6 Failure Modes and Degraded Operation

Implementations SHOULD define explicit degraded modes of operation.

Examples include:

- Capture without immediate anchoring due to network unavailability,
- Temporary fallback to local secure storage of commitments,
- Postponed anchoring once connectivity is restored.

All degraded modes MUST be transparently detectable during verification.

9 Live Streaming Verification

9.1 Hash-Chain Construction

Instead of anchoring every frame:

- Each video frame hash is chained to the previous frame hash, forming a continuous hash chain.

9.2 Periodic Blockchain Anchoring

At fixed intervals (e.g., 1–5 seconds), the current hash chain is anchored to the blockchain, allowing post-hoc verification of every frame without excessive transactions.

9.3 Stream Commitment

At stream start, a commitment may be anchored defining:

- Device identity,
- Stream identifier,
- Start time,
- Hashing parameters.

10 Verification Properties

The system enables verification that:

- Media originated from a specific physical device,
- Capture occurred after a given timestamp,
- Content has not been altered or reordered,
- Live streams have not had frames inserted or removed.

> This system provides cryptographic evidence of capture provenance and integrity. It does not assert factual accuracy, legitimacy, or intent of the recorded content.

11 Security Model and Limitations

11.1 Addressed Threats

- Post-processing and manipulation,
- AI-generated content falsely presented as captured,
- Metadata forgery.

11.2 Known Limitations

- Re-recording of displays or analog re-capture,
- Physical compromise of capture hardware,
- Malicious firmware prior to capture.

Possession of the original media file alone is insufficient to reconstruct a valid capture proof. The security of the system relies on the unforgeability of hardware-bound signatures, not on secrecy of the media content.

12 Intended Applications

- Journalism and news media,
- Social media authenticity indicators,
- Legal and forensic evidence,
- Body cameras and surveillance,
- Government and institutional broadcasting,
- High-trust live transmissions.

13 Out of Scope

This proposal explicitly does not attempt to:

- Determine the factual truth, authenticity, or semantic correctness of any media content.
- Detect or classify manipulated, synthetic, or AI-generated media.
- Prevent re-recording, analog capture, or off-device reproduction attacks.
- Enforce mandatory adoption, platform-level restrictions, or policy-based content filtering.
- Act as a legal, moral, or regulatory authority over media validity.

The system is limited to providing verifiable provenance signals based on hardware-backed evidence at the time of capture. Any interpretation, enforcement, or trust decision remains entirely outside the scope of this work.

14 Disclosure Intent

This document is published as a defensive technical disclosure to establish prior art. The author does not waive moral authorship or attribution rights. Its intent is to prevent exclusive patent claims by third parties while enabling open discussion, standardization, and further research.