Ensuring Image Integrity Using Steganography and Blockchain Technology

Baltag Ștefan-Bogdan
Technical Military Academy
"Ferdinand I"
Bucharest, Romania
baltagstefanbogdan@gmail.com

Aloman Alexandru
Special Telecommunication Service
Bucharest, Romania
alexandru.aloman@stsnet.ro

Bratu Alexandru
National University of Science
and Technology Politehnica Bucharest
Bucharest, Romania
ing.bratu.alexandru@gmail.com

Goga Nicolae
National University of Science
and Technology Politehnica Bucharest
Bucharest, Romania
n.goga@rug.nl

***Abstract*—This paper proposes a decentralized solution for digital image integrity based on a hybrid architecture that combines LSB steganography and blockchain smart contracts. The system allows users to digitally sign and verify images without centralized storage or disclosure of content. The solution ensures tamper resistance, user authentication, and traceable verification using a local Ethereum blockchain and a secure web platform.**

***Index Terms*—Steganography, Blockchain, Ethereum, Digital Signature, Image Integrity, Smart Contracts, MetaMask, Flask, React.**

I. Introduction

Digital images serve as vital evidence in legal, media, and defense domains. However, due to advancements in editing software and AI (e.g., deepfakes), traditional protection methods such as watermarking are insufficient. This work addresses image integrity through a robust system that embeds cryptographic data within images and manages verification using blockchain technology.

II. System Design and Implementation

The system is built as a full-stack web platform that integrates cryptographic operations, steganographic processing, and decentralized user authentication. Its architecture consists of several interconnected components:

A. Frontend: React Interface

The frontend is implemented using React.js. It offers an interactive user interface for uploading images, managing MetaMask wallet connections, initiating signing or verification, and displaying verification results. State and effect hooks are used to track MetaMask account changes and coordinate backend interactions.

B. Blockchain: Local Ethereum Network (Ganache)

A local Ethereum blockchain, deployed using Ganache, serves as the foundation for decentralized identity and access control. Smart contracts written in Solidity handle:

- User role management (admin, signer/verifier, verifier-only),
- Account registration and approval,
- Logging of verification events.

Each user is identified by their Ethereum public address and authenticated via MetaMask.

C. Authentication and Signing: MetaMask + ECDSA

MetaMask is used for digital identity management and cryptographic signing. Users sign image hashes generated by the backend using their private key. The ECDSA (Elliptic Curve Digital Signature Algorithm, curve secp256k1) is employed to generate secure, verifiable signatures.

D. Backend: Flask Image Processing

The Flask backend performs the following tasks:

1) Accepts images from the user interface.
2) Computes the SHA-256 hash of the uploaded image.
3) Receives the signature from MetaMask and embeds it into the image using LSB steganography.
4) Extracts and verifies embedded signatures during verification.

LSB steganography modifies the least significant bits of RGB channels in image pixels to hide the cryptographic data, ensuring invisibility and minimal distortion.

E. Data Storage: SQLite

All non-sensitive data—such as user access logs, signing/verifying durations, and statistics—is stored locally in an SQLite database. This enables fast, lightweight analytics while preserving the image's confidentiality (no image content is stored).

III. Experimental Results

Tests were conducted under various scenarios:

- **Authentic Images**: Correct validation.
- **Modified Images**: Detected by hash mismatch.

- **Unsigned Images**: Flagged as unverifiable.

Performance:

- Average signing time: $\sim$1.0 seconds
- Average verification time: $\sim$0.6 seconds
- 100% detection for tampered and unsigned files

Admin Dashboard: Includes user role management, activity logs, metrics tracking (image count, signing times), and real-time statistics visualization.

IV. Conclusions and Future Work

This paper introduced a decentralized system for securing the authenticity and integrity of digital images using a combination of digital signatures, steganography, and blockchain technology. The proposed platform allows users to sign and verify images locally, without relying on centralized storage or exposing the content of the media. Identity and access control are handled via a smart contract deployed on a local Ethereum blockchain, ensuring transparent and tamper-resistant user management.

Experimental results demonstrated the system's ability to correctly validate signed images and detect any unauthorized modifications. The platform includes features for user access control, image verification, and real-time monitoring, making it suitable for deployment in environments that require controlled media distribution and verifiable image integrity.

Future Work

Future work will focus on:

- implementing an audit mechanism based on a public blockchain,
- optimizing the embedding and extraction process for high-resolution images,
- extending support for additional digital file types beyond images.

These enhancements aim to improve system scalability and extend its applicability to domains such as legal evidence management.