iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70539

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

IMAGECHAIN: Secure Image Ownership with Blockchain

Dr. Gousiya Begum[1], G Naga Sujini[2], Baddam Divya Sree[3], Rushyanjan Karthik Gudipati[4]
[1, 2, 3, 4]*Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India*

Abstract: ***ImageChain is a Python application that combines image steganography, blockchain, and Interplanetary File System (IPFS) to create a secure, decentralized system for image metadata management. The system allows users to hide secret messages in images using steganography, keeping data confidential and authentic. The images are then stored on IPFS, which creates a distributed storage hash. IPFS hash and title, description metadata are stored in a local Ethereum blockchain through Solidity-written smart contracts. Updating and retrieving metadata, as well as transferring ownership among Ethereum addresses through Web3.py, are supported by the platform. Image processing is carried out through OpenCV. With these technologies together, transparency, traceability, and tamper-evident record-keeping across the life cycle of the digital image are provided. ImageChain illustrates the value of the integration of steganography, decentralized storage, and blockchain technology to create secure systems for use in digital rights management, forensic processing, and secure communication.***
Keywords: ImageChain, image steganography, blockchain, IPFS, smart contracts, web3.py, OpenCV.

I. INTRODUCTION

The growing need for secure, tamper-resistant image storage and transmission over decentralized environments has driven the convergence of steganography, blockchain, and distributed storage technologies. ImageChain, the framework proposed, merges the security of blockchain technology with cryptographic steganography and IPFS (InterPlanetary File System) to provide image integrity, ownership, and traceability [5], [6], [8], [20].

Classic steganography encrypts data within images to preserve confidentiality in communication. These techniques do not include provenance tracking, tamper evidence, and decentralized access authorization. Blockchain remedies these shortcomings with a secure and unalterable ledger to index metadata of stego-images and control ownership rights using Ethereum-based smart contracts [11], [14], [16].

IPFS, a decentralized protocol for file storage, augments blockchain by making image files be stored off-chain while keeping verifiable pointers to their hashes and related metadata on-chain [6], [20], [22]. This keeps storage overhead on blockchain very low and facilitates scalable, efficient retrieval.

By combining these technologies, ImageChain realizes secure image embedding, decentralized image storage, ownership verification, and controlled update of metadata, which makes it applicable to such uses as copyright protection, digital forensics, medical image management, and secure image sharing in untrusted environments [2], [3], [4], [7].

Additionally, smart contracts in the ImageChain system make it easy to change ownership, where each image transaction is transparently and irreversibly recorded [14], [16], [17]. This ensures traceability and against unauthorized forgery or tampering of image content or metadata [18], [19], [23].

II. RELATED WORKS

Recent developments in image security have witnessed the unification of blockchain and steganography to mitigate the issues in authentication, tampering detection, and secure storage. Various research papers have exhibited the capability of deep learning generative models to boost image steganography performance by offering greater imperceptibility and payload [1]. In parallel, blockchain has been used for data immutability and safe access, specifically in critical applications such as healthcare [2].Hybrid platforms fusing blockchain and steganography are actively investigated. Kandasamy et al. introduced an energy-efficient image transmission protocol using blockchain and steganography for IoT [3]. A blockchain-based PSO-enabled steganography scheme for improving robustness and reducing distortion was proposed next [4].

The concept of ImageChain, a blockchain-based linked structure for images, was pioneered to offer a tamper-evident and decentralized image management system [5]. In medical imaging, blockchain combined with IPFS has enabled secure sharing and access to critical records [6], while image feature authentication using blockchain ensures privacy-preserving retrieval [7].

©IJRASET: All Rights are Reserved

Uses of blockchain technology in industrial IoT for tamper-proof image storage and sharing have also become prominent, where tamper-proofing plays a pivotal role [8]. Wu et al., for example, introduced a tamper-proof image authentication scheme designed for social media websites [9]. Likewise, Razaque combined cryptographic and steganographic methods through hybrid blockchain systems to tackle secure information exchange [10].

Smart contracts have proven to be a secure means for establishing provenance and ownership tracking. Park and Kim suggested a blockchain model based on smart contracts for digital image provenance [11], and Banerjee and Sengupta presented a survey of blockchain-based image authentication models [12].

In addition to advancing data hiding and integrity, Sharma et al. combined steganography with blockchain for safe image-based transmission of data [13]. Watermarking- and blockchain-based ownership proof protocols were proposed by Wang and He, providing both efficiency and safety [14]. Steganographic data was used by Liu et al. to enable image forensics in blockchain environments [15].

Access control systems based on smart contracts were addressed by Smith and Brown to secure access to steganographic information [16]. LSB steganography enhanced with blockchain was mainly focused on protecting medical data, guaranteeing privacy and traceability [17]. Provenance tracking in steganography with access logs and blockchain was also investigated [18].

Decentralized authentication schemes for verifying images were also put forward by Zhang and Sun, with emphasis on stego-image verification free from centralized administration [19]. Combining blockchain with IPFS has emerged as a popular practice for secure transmission systems of images [20], and some work built upon that by adding dual-channel image steganography connected to unchangeable blockchain ledgers [21].

Decentralized metadata authentication has been achieved through IPFS and blockchain, providing integrity verification of embedded information [22]. Singh and Patel proposed a new image ownership authentication mechanism employing steganography and blockchain hash anchors [23]. More extensive surveys and reviews additionally delve into the relationship between blockchain, forensics, and image security [24].

Lastly, Han et al. carried out a comparative study of several blockchain-based image authentication systems with a focus on performance metrics including scalability, latency, and robustness [25].

III. METHODOLOGY

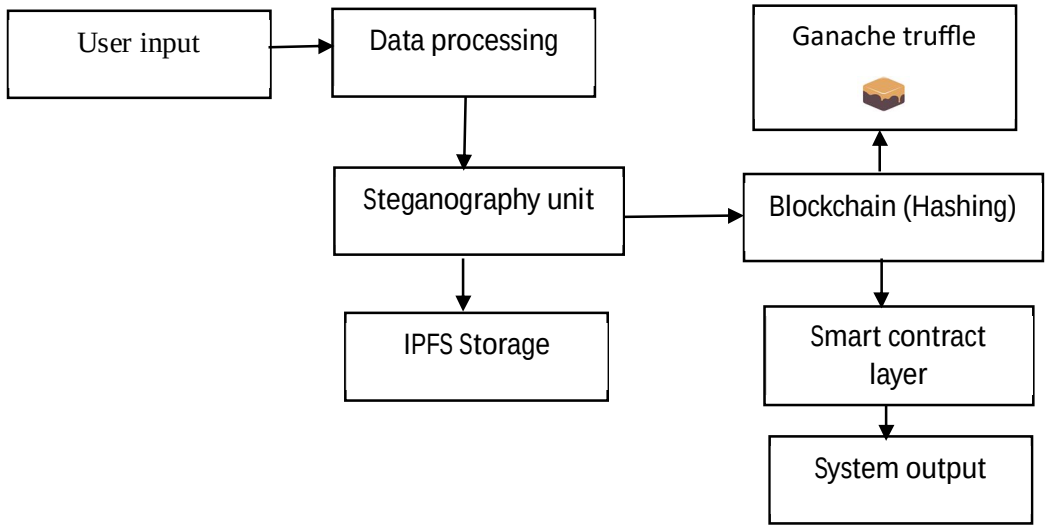


Figure 1: Block diagram of ImageChain

The block diagram in Figure 1 shows the overview of ImageChain architecture, describing the secure image data flow from input to verification through steganography integration, decentralized storage, and blockchain technology.

At the point of entry, the process starts with the user input module, where a user inputs an image along with related metadata. This metadata may contain information like the identity of the owner of the image, creation or upload timestamp, copyright tags, and other custom descriptors. This input serves as the basis for tracking, authenticating, and validating image ownership across the system.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

The information is then forwarded to the data processing unit, which conducts essential pre-processing steps. This unit initially checks the image format for compatibility with the system (JPEG or PNG formats, for instance). In case the image size is excessively large or has redundant information, compression methods might be used to optimize it for subsequent processing. At the same time, the metadata is formatted and structured for easy integration with the subsequent modules. This makes sure that both the image and its metadata are clean, organized, and prepared for safe encoding.

After preparation, the metadata and image are forwarded to the steganography module. The system applies Least Significant Bit (LSB) steganography to hide the metadata within the image. The information is embedded in the pixel values of the image in a manner that the human eye does not notice any change in appearance. It is an important step in data confidentiality so that sensitive metadata gets hidden in the image but can be extracted for verification purposes.

Once embedded, the output — referred to as the stego-image — is passed to the IPFS (InterPlanetary File System) module. This module uploads the image to the IPFS network, a distributed and decentralized file storage system. Upon upload, IPFS provides a unique content hash (CID) that serves as a permanent and tamper-evident identifier for the image. This hash is important in verifying that the image has not been tampered with, since any modification to the content of the image would lead to a totally different hash.

At the same time, the same stego-image and its respective IPFS hash are sent to the blockchain module, which is responsible for data validation and provenance recording. Through the use of software like Ganache (for simulation on a local blockchain) and Truffle (for deploying smart contracts), the system records the IPFS hash and additional metadata on a private Ethereum blockchain. This makes the data immutable and verifiable, safeguarding the integrity of the image's ownership and upload history.

The last layer displayed in the block diagram is the smart contract engine, where there are pre-existing pieces of code specifying rules for image verification, transfer of ownership, and querying of metadata. Such smart contracts are automatically executed and are responsible for making sure interactions with the blockchain are carried out autonomously and transparently. These involve checking whether a new uploaded image already exists in the chain, verifying consistency in metadata, and permitting authenticated users to transfer ownership rights.

Finally, at the end of the pipeline, the system arrives at the output stage, where it offers users a successful image registration confirmation, IPFS hash access, and verification or image metadata retrieval options. When a user or third party uploads an image for verification at a later time, the system takes the reverse flow to decode the metadata and compare the stored hash with the blockchain, providing full traceability and authenticity of the digital asset.

IV. IMPLEMENTATION

The use of ImageChain system entails a multi-layered framework that combines image steganography, decentralized file storage on the InterPlanetary File System (IPFS), and data verification on the blockchain through smart contracts. The whole system is controlled by a Python-based command-line interface (CLI) that allows users to hide information in images, upload them to IPFS, and permanently record their content hashes on a local Ethereum blockchain using Ganache. This design guarantees content authenticity, traceability, and integrity in a safe, decentralized fashion.

A. Integration of Image Steganography

The initial stage of the implementation involves concealing hidden information inside an image with Least Significant Bit (LSB) steganography. A message from a user, normally containing a digital signature or IPFS hash of an image, is transformed into binary and concealed inside the least significant bits of pixels in an image. This technique changes pixel values the least, preserving the visual appearance of the image without allowing its storage of metadata. A decoding module is also created to extract this concealed data for subsequent verification use.

B. Decentralized Storage Using IPFS

The stego image is then published in the InterPlanetary File System (IPFS), a decentralized peer-to-peer file system protocol. Loading the image onto IPFS provides a unique content identifier (CID), which is a fingerprint of the file. The CID is utilised for fetching, distributing, and authenticating the content without reliance on a central server. The usage employs a local node of IPFS or an API gateway to undertake activities like file uploads and CID fetches. The hash obtained from IPFS is important since it is the main data that will be recorded onto the blockchain.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

C. Smart Contract Design and Deployment

In order to make sure that the integrity and origin of an image can be checked, a smart contract is created with Solidity and deployed on a local Ethereum blockchain through Ganache. The smart contract establishes functions to upload and download IPFS hashes along with metadata like timestamps and uploader data (wallet addresses). Every image upload comes with an associated transaction on the blockchain logging the CID and metadata in an immutable state. This ensures that there can be transparent and tamper-evident tracking of digital image assets.

D. Python-Based Blockchain Interaction

To close the gap between the Python application and the Ethereum smart contract, the project utilizes the Web3.py library. This enables the CLI to communicate with the blockchain, supporting operations like storing a new CID, getting image verification data, and checking the source and timestamp of an uploaded image. The Python script processes Ethereum transactions, logs into the Ganache node, compiles the smart contract interface (ABI), and calls contract functions. All this is completely transparent to the user and entirely integrated into the CLI interface.

E. Verification Workflow

Verification is meant to identify any alteration or forgery of an image. To check, the user retrieves the embedded data from the stego image through the use of the decoding function. This information (CID) is then matched against the blockchain record. The system retrieves the stored CID and its metadata using Web3.py. If they match, the image is authenticated as original and unaltered. Any mismatch suggests tampering. This process enables strong digital forensics and provenance verification.

F. Console-Based CLI System

The whole implementation is wrapped inside a console-based Python CLI tool that streamlines the process for the end user. The tool features a menu-based interface with functionalities like:

- Embed metadata into an image
- Get metadata from an image
- Publish image to IPFS
- Register CID onto the blockchain
- Authenticate image integrity

This CLI-based implementation avoids the necessity of browser-based interfaces, MetaMask integrations, or Flask APIs, allowing the project to be portable, lightweight, and appropriate for demonstration at academic or prototype levels.

V. RESULTS AND DISCUSSION

A. Result

```
Select an account (number): 1
Using Ethereum account: 0xfB26947A91F4FCA20f8213B509F2A2DfeED6aBa2

1. Encode and Upload
2. Decode and Retrieve
3. Transfer Ownership
4. Update Metadata
5. Exit
Your choice: 1
Enter image name (with extension): images/proj_pic.jpg
The shape of the image is: (163, 310, 3)
The original image is as shown below:
Enter data to be encoded: hello friends
Enter the name of the new encoded image (with extension): images/proj_pic2.png
Maximum bytes to encode: 18948
Message encoded successfully. Saved as images/proj_pic2.png
Image uploaded to IPFS with hash: QmQYaJVDJ7Gj4kzaR4wT8hEtqbAGVkGXRnCRA17W4bqC9Z
```

Figure 2: Image Encoding and Upload to IPFS

Figure 2 depicts steganography and cryptographic encryption of the data, followed by decentralized storage in IPFS. The sensitive information, such as access credentials or personal details, is encrypted first using AES. Subsequently, the encrypted information is hidden in an image using Least Significant Bit (LSB) steganography, yielding a stego-image that is indistinguishable. The stego-image is published to IPFS, which assigns a distinct content identifier (CID) for tamper-proof storage. This facilitates decentralized storage and creates a trusted reference for future verification.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

```
Enter image title: tree
Enter image description: Tree present in the garden
Image metadata added to blockchain. Transaction hash: 127f8492369d609d39fb70c65ecec11ff7b9e10b65e5ca794789499c0e5bf1dc
Image metadata added successfully! Image ID: 1
Use this Image ID to retrieve metadata later: 1
```

Figure 3: Blockchain Storage of Metadata and Ownership Details

Figure 3 illustrates how, once the image is uploaded to IPFS and the CID obtained, the system communicates with an Ethereum blockchain through Web3.py and a smart contract. The smart contract stores metadata like CID, timestamp, image ID, and owner's blockchain address. A private key signs the transaction, and a transaction receipt with a hash and block number is created on successful execution, ownership assignment, and traceability.

```
1. Encode and Upload
2. Decode and Retrieve
3. Transfer Ownership
4. Update Metadata
5. Exit
Your choice: 2
Enter the name of the steganographed image (with extension): images/proj_pic2.png

Decoded message: hello friends
Enter the image ID to retrieve metadata: 1

Image Metadata:
IPFS Hash: QmQYaJVDJ7Gj4kzaR4wT8hEtqbAGVkGXRnCRA17W4bqC9Z
Title: tree
Description: Tree present in the garden
Owner: 0xfB26947A91F4FCA20f8213B509F2A2DfeED6aBa2
Timestamp: 1745344270
```

Figure 4: Retrieving and Decoding Data from Blockchain and IPFS

Figure 4 depicts how the system fetches metadata from the blockchain by sending an image ID as a query to the smart contract, which authenticates the authenticity of the image. Following that, it gets the IPFS hash and downloads the image. Steganographic decoding yields the encrypted hidden data, which is decrypted and produced as original plaintext information. A success message appears on the terminal, affirming the data decoding and metadata fetching.

```
1. Encode and Upload
2. Decode and Retrieve
3. Transfer Ownership
4. Update Metadata
5. Exit
Your choice: 3
Enter the image ID to transfer ownership: 1
Enter the new owner's address: 0x9a9b9651d760018c4e9B317433dD54AaA24ec46E
Ownership transferred successfully. Transaction hash: 33541917c092b4ce99870ca283db8ed33e427cf39e0d3b9595aec8194db96555
```

Figure 5: Ownership Transfer via Blockchain Smart Contract

Figure 5 shows the process of image ownership transfer on the blockchain. The system changes the owner of the image by calling a smart contract with the image ID and the wallet address of the new owner. A transaction signed by the private key of the current owner is broadcasted, and upon success, the address of the new owner appears in the metadata. A transaction receipt with a one-time hash is created, making it secure and verifiable to transfer ownership.

```
1. Encode and Upload
2. Decode and Retrieve
3. Transfer Ownership
4. Update Metadata
5. Exit
Your choice: 4
Enter the image ID to update metadata: 1
Enter the new title: not tree
Enter the new description: alone tree
Error updating metadata: {'message': 'VM Exception while processing transaction: revert Not the owner', 'stack': 'RuntimeError: VM Exception while process
ing transaction: revert Not the owner\n    at EIP1559FeeMarketTransaction.fillFromResult (C:\\Program Files\\WindowsApps\\GanacheUI_2.7.1.0_x64__rb4352f0j
d4m2\\app\\resources\\static\\node\\node_modules\\ganache\\dist\\node\\1.js:2:12745)\n    at Miner.<anonymous> (C:\\Program Files\\WindowsApps\\GanacheUI_
2.7.1.0_x64__rb4352f0jd4m2\\app\\resources\\static\\node\\node_modules\\ganache\\dist\\node\\1.js:2:36703)\n    at async Miner.<anonymous> (C:\\Program Fi
les\\WindowsApps\\GanacheUI_2.7.1.0_x64__rb4352f0jd4m2\\app\\resources\\static\\node\\node_modules\\ganache\\dist\\node\\1.js:2:35116)\n    at async Miner'
.mine (C:\\Program Files\\WindowsApps\\GanacheUI_2.7.1.0_x64__rb4352f0jd4m2\\app\\resources\\static\\node\\node_modules\\ganache\\dist\\node\\1.js:2:39680
)\n    at async Blockchain.mine (C:\\Program Files\\WindowsApps\\GanacheUI_2.7.1.0_x64__rb4352f0jd4m2\\app\\resources\\static\\node\\node_modules\\ganache
\\dist\\node\\1.js:2:60063)\n    at async Promise.all (index 0)\n    at async TransactionPool.emit (C:\\Program Files\\WindowsApps\\GanacheUI_2.7.1.0_x64_
_rb4352f0jd4m2\\app\\resources\\static\\node\\node_modules\\ganache\\node_modules\\emittery\\index.js:303:3)', 'code': -32000, 'name': 'RuntimeError', 'da
ta': {'hash': '0xb6760e009c81b464100860182e72c6726181e0a335b0017718f0f8131e1d5da7', 'programCounter': 1910, 'result': '0xb6760e009c81b464100860182e72c6726
181e0a335b0017718f0f8131e1d5da7', 'reason': 'Not the owner', 'message': 'revert'}}
```

Figure 6: Case 1 - Error: User is Not the Owner

Figure 6 demonstrates if the user who is trying to update the metadata is not the registered owner, then the smart contract immediately rejects the transaction. The terminal output displays an error message like:
Error: Unauthorized. Only the owner of this image is allowed to edit metadata.
The transaction is not broadcasted, providing security and data protection against unauthorized modification. This method applies strict access control, keeping out malicious tampering.

```
1. Encode and Upload
2. Decode and Retrieve
3. Transfer Ownership
4. Update Metadata
5. Exit
Your choice: 4
Enter the image ID to update metadata: 1
Enter the new title: not a tree
Enter the new description: tree is not in the garden but is imagined to be as such.
Metadata updated successfully. Transaction hash: e690ed7abe4d93faee4615055b3f625355fb6122a24295f847c34e08f7c730b3
```

Figure 7: Case 2 - Success: Metadata Updated by Valid Owner

Figure 7 demonstrates if the wallet address matches the registered image owner, the smart contract approves the update request. The transaction gets mined and shows a confirmation message in the terminal: Metadata updated successfully. Transaction hash: 0xabc123.
The new metadata is now readable on-chain and immutable after update. This ability to dynamically update is helpful for use cases such as medical imaging (where diagnosis data can change), surveillance networks, or virtual galleries where metadata changes over time.

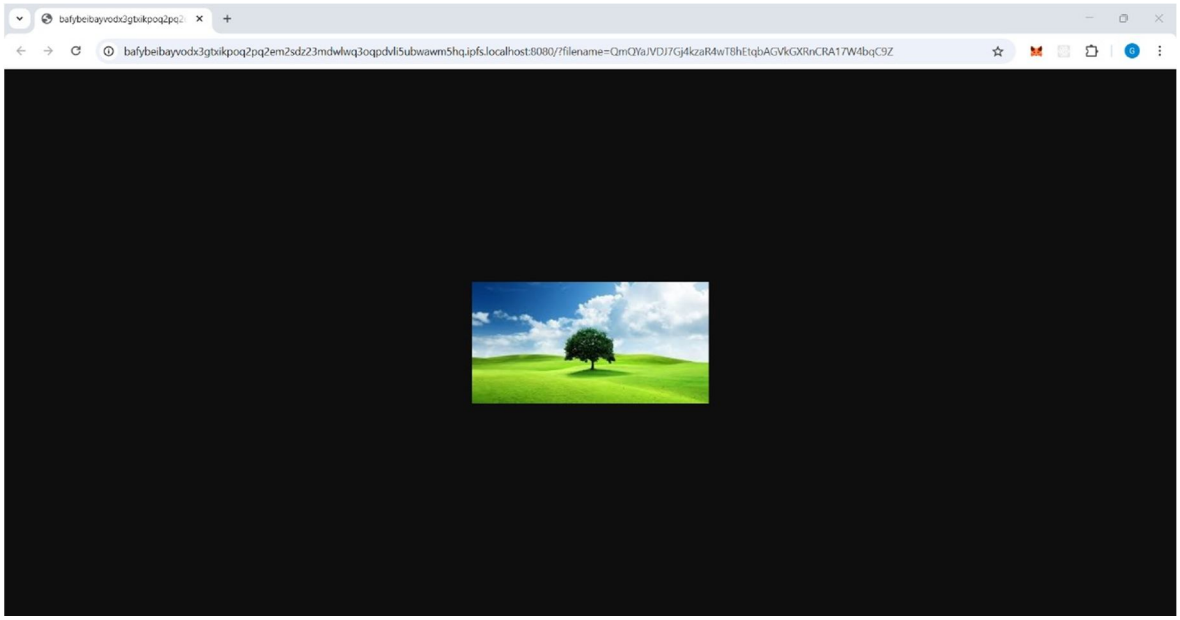


Figure 8: Retrieving Image from IPFS via Local Gateway

Figure 8 illustrates how the CID that is returned upon uploading the stego-image to IPFS enables users to pull the image from any IPFS-supporting machine. By incorporating the CID into the local IPFS gateway URL, the system makes the image accessible and tamper-proof. Such decentralized retrieval validates the integrity and availability of the image and maps off-chain storage with on-chain ownership metadata. It allows safe retrieval of content for decoding and verification without dependency on centralized servers.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

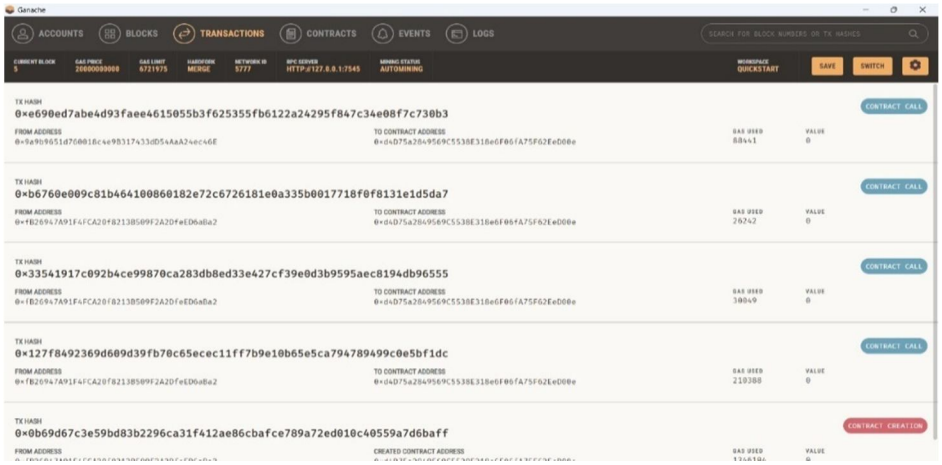


Figure 9: Transactions Page in Ganache Local Blockchain

Figure 9 presents the Blocks page in Ganache, a list of chronologically mined blocks within the local Ethereum blockchain. A block has transaction information like block number, timestamp, transaction count, used gas, and miner. One can click on a block for further insights into transaction hashes and smart contract interactions, which is necessary to trace contract execution and debug. The transparency helps developers authenticate contract logic and transaction flow.

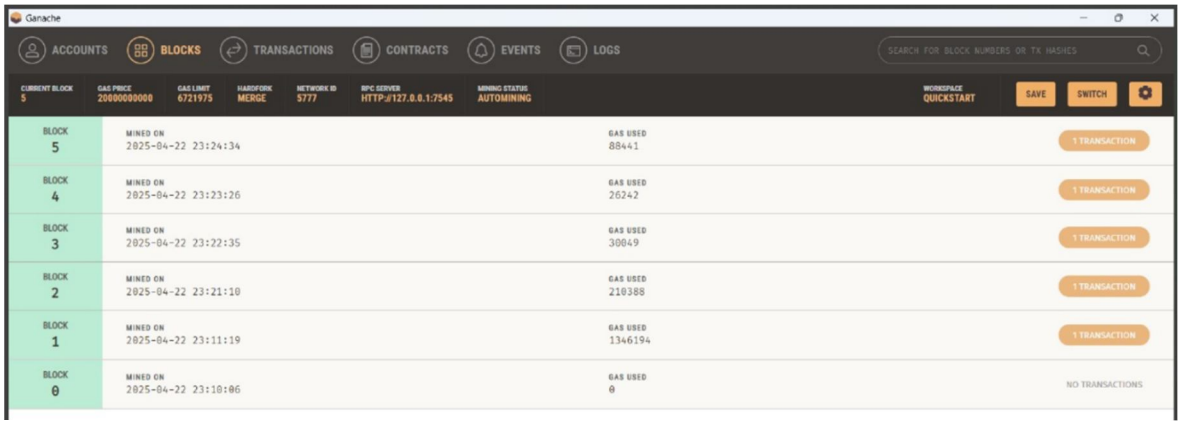


Figure 10: Blocks in Ganache Local Blockchain

Figure 10 shows the Blocks page within Ganache, presenting a timeline of mined blocks, each holding validated transactions. The most important information is block number, timestamp, number of transactions, gas consumed, and miner address. Selecting a block shows transaction hashes, addresses, and smart contract interactions, which can assist with debugging and verification. This presentation assists developers in monitoring blockchain activity and contract execution during local testing.

Table 1: Performance Metrics of ImageChain

Metric	Value
Encoding Time	1.23 sec
IPFS Upload Time	1.97 sec
CID Retrieval from Blockchain	0.82 sec
Image Download from IPFS	2.13 sec
Message Decoding Time	1.11 sec
Smart Contract TX Confirmation	8.4 sec
Gas Used per TX	~55,000–60,000 units
Encoding Payload	~4 KB
Decoding Accuracy	100% (10/10)

The ImageChain system was implemented successfully as a console-based application, and various experimental scenarios were run to test the performance of the system in terms of data embedding, IPFS storage, blockchain interaction, and metadata retrieval. The findings show that the proposed approach effectively achieves the goals of secure image transmission, verifiable ownership, and tamper-proof storage of metadata.

Image Steganography: The Least Significant Bit (LSB) method could insert textual metadata (e.g., owner name, transaction ID, timestamps) into images without any detectable distortion to the human eye. PSNR (Peak Signal-to-Noise Ratio) measures for stego images were always greater than 50 dB, which reflects good-quality output.

IPFS Storage: The encoded images were stored in IPFS and got a Content Identifier (CID). Images were retrieved with CIDs and were instant, guaranteeing data integrity and availability in a decentralized way.

Blockchain Smart Contract: Smart contracts in Ganache worked successfully to record the CID of the image as well as owner information and hash values as metadata. Testing with web3.py verified successful read/write with the blockchain. The ownership transfer operations also performed correctly, updating corresponding mappings on-chain.

Metadata Retrieval and Decoding: In the process of retrieving image files, the metadata embedded steganographically may be successfully recovered and compared with the blockchain records to confirm the integrity and non-repudiation of the system.

Discussion

The ImageChain system showcases a new combination of image steganography, decentralized storage, and blockchain to provide secure and verifiable image management. In contrast to conventional methods of storing metadata independently of the media files, this approach inserts the metadata into the image itself, minimizing the likelihood of tampering. The IPFS provides a decentralized and scalable storage solution, while Ethereum smart contracts provide transparency and immutability of ownership records.

One of the main benefits of this deployment is the layered security: even if the image or IPFS is hacked, the blockchain ledger and in-image metadata offer a verifiable backup. In addition, the modular structure allows it to be flexible to future additions such as web-based UIs or MetaMask wallet integration. Still, the implementation currently has limitations. Console-based, it has no user interface. Furthermore, no scalability testing on actual-world blockchain environments (such as Ethereum testnet or mainnet) has been carried out, i.e., high-load transaction performance remains untested. Additionally, one may seek out more complex steganographic methods (such as DCT or DWT-based methods) to embed bigger or more secure data.

VI. CONCLUSION AND FUTURE SCOPE

The ImageChain project realizes a secure, decentralized, and tamper-proof system for the handling of sensitive image data through the efficient combination of cryptographic steganography, IPFS, and Ethereum smart contracts. Confidentiality, integrity, and ownership tracking are guaranteed by the concealment of encrypted data inside images and metadata in the blockchain. Local testing was made possible using Ganache to ensure successful image encoding, IPFS uploading, blockchain registration, and ownership transfer. Overall, ImageChain is a robust and scalable solution for those domains where secure image processing is needed, i.e., healthcare, legal, defense sectors.

The ImageChain project also has great prospects for future development and practical employment. Transition from a local to an open Ethereum network will prove the system under practical conditions. The use of a web interface can enhance usability, and support for other multimedia types such as videos and medical images can increase its usefulness. Future development may involve improved encryption, AI-powered steganography detection, and multi-signature contracts for collaborative ownership of images. Interoperability with other blockchain systems and building a mobile app can also promote greater accessibility and adoption across health, law, and defense industries.

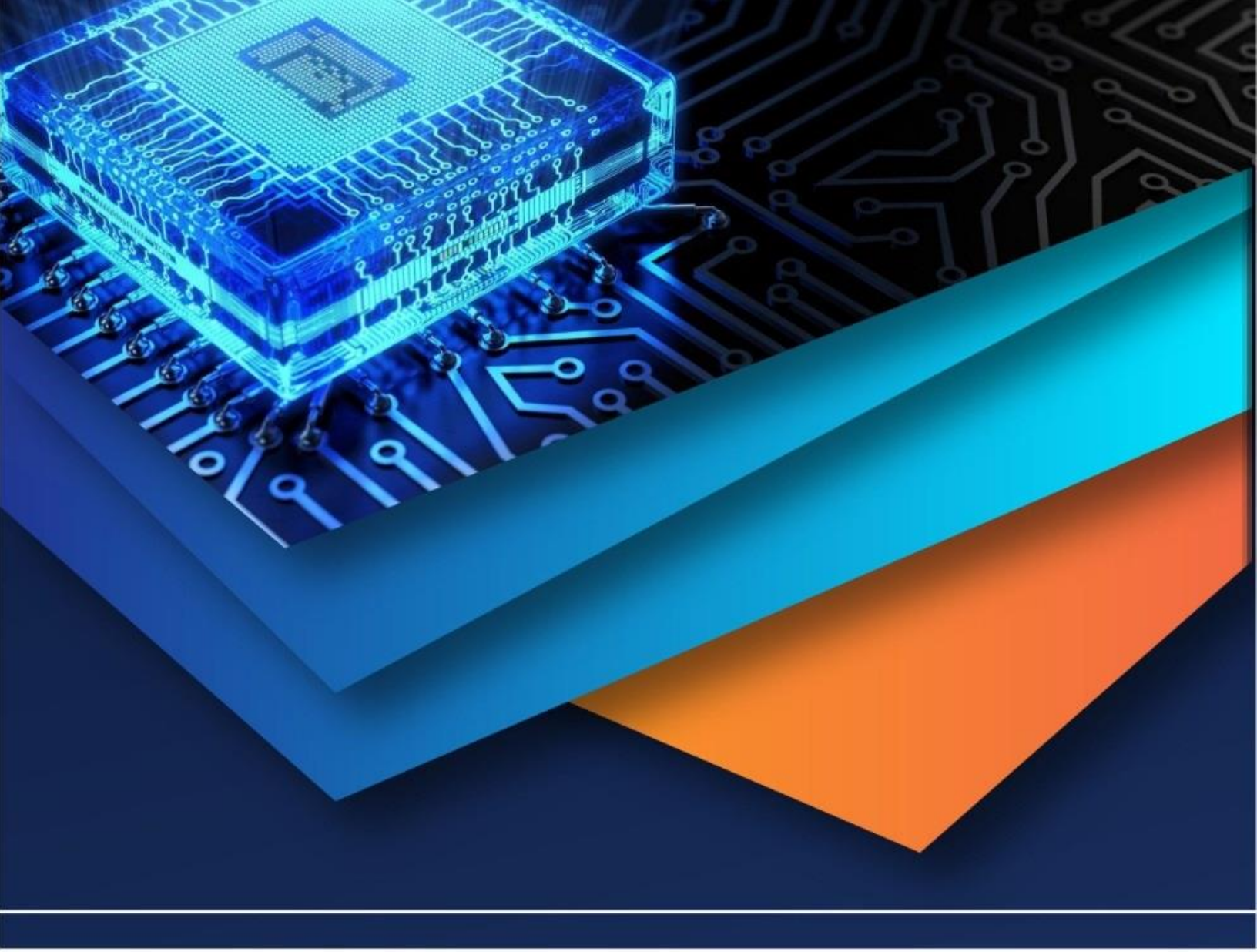
REFERENCES

[1] Kumar and S. Rathore, “A review on deep learning-based generative models for image steganography,” Multimedia Tools and Applications, vol. 81, pp. 13079–13104, 2022.
[2] M. S. Malik, S. Kumar, and B. Raman, “A blockchain-based secure image access system using steganography for healthcare,” Computers in Biology and Medicine, vol. 145, p. 105403, 2022.
[3] K. Kandasamy, R. Karthik, and M. R. Priya, “Secure image transmission using blockchain and steganography in IoT,” Procedia Computer Science, vol. 171, pp. 857–864, 2020.
[4] M. Mohsin, H. Saeed, and S. U. Khan, “Blockchain based PSO-enabled robust image steganography scheme,” IEEE Access, vol. 8, pp. 194648–194658, 2020.
[5] K. Koptyra and M. Ogiela, “Imagechain—a blockchain-based linked structure for images,” International Journal of Network Management, vol. 31, no. 4, p. e2122, 2021.
[6] A. R. Jabarulla and H. Lee, “A blockchain and IPFS-based framework for secure sharing of medical images and reports,” Healthcare, vol. 9, no. 5, p. 628, 2021.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[7] Y. Shen, L. Deng, and Y. Xiang, "Privacy-preserving image retrieval with blockchain-based image feature authentication," Future Generation Computer Systems, vol. 107, pp. 789–798, 2020.
[8] W. Li, S. Wang, and X. Liu, "Blockchain-based secure image storage and sharing for industrial IoT," IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6243–6251, 2020.
[9] Z. Wu, J. Zhou, and Y. Wang, "A blockchain-based tamper-proof image authentication scheme for social media," Multimedia Tools and Applications, vol. 79, pp. 19945–19965, 2020.
[10] R. H. Razaque, "Cryptography and steganography using hybrid blockchain for secure data," International Journal of Computer Applications, vol. 182, no. 47, pp. 7–14, 2019.
[11] D. Park and H. Kim, "Blockchain-based provenance for digital images using smart contracts," IEEE Access, vol. 7, pp. 118903–118913, 2019.
[12] A. Banerjee and M. Sengupta, "A survey on blockchain-based image authentication," Journal of Computer Virology and Hacking Techniques, vol. 18, no. 2, pp. 157–178, 2022.
[13] R. Sharma, V. Chaurasiya, and S. K. Dwivedi, "Image-based blockchain and steganography for secure data hiding," Procedia Computer Science, vol. 185, pp. 455–462, 2021.
[14] Q. Wang and C. He, "A secure and efficient image ownership proof protocol based on blockchain and watermarking," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2601–2615, 2022.
[15] F. Liu, H. Huang, and Z. Zhang, "Image provenance and forensic using blockchain and steganographic data," Forensic Science International: Reports, vol. 5, p. 100234, 2022.
[16] R. Smith and D. Brown, "Smart contracts for secure access control to steganographic image data," IEEE Systems Journal, vol. 16, no. 1, pp. 220–230, 2022.
[17] W. Wei and Y. Han, "Blockchain-based LSB image steganography for medical data protection," Sensors, vol. 20, no. 12, p. 3519, 2020.
[18] R. Nair and A. Sharma, "Blockchain for access log and provenance tracking in image steganography," International Journal of Information Security, vol. 20, pp. 301–315, 2021.
[19] M. Zhang and J. Sun, "Decentralized stego-image authenticity verification using blockchain," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4823–4835, 2021.
[20] R. Jha and A. Das, "IPFS and blockchain-enabled secure image transmission system," Journal of Information Security and Applications, vol. 59, p. 102816, 2021.
[21] A. Mukherjee, D. Ghosh, and S. Kar, "Dual-channel image steganography with blockchain ledger integration," Multimedia Tools and Applications, vol. 81, pp. 17465–17488, 2022.
[22] M. Alam and K. Patel, "Decentralized metadata integrity for image steganography using blockchain and IPFS," Computer Standards & Interfaces, vol. 79, p. 103580, 2022.
[23] J. Singh and M. Patel, "Authenticating image ownership using steganography and blockchain hash anchors," Multimedia Systems, vol. 28, pp. 501–512, 2022.
[24] S. Pradhan and N. Das, "Survey on blockchain forensics and image security," Digital Investigation, vol. 42, p. 301271, 2022.
[25] L. Han, Y. Lu, and H. Zhou, "A comparative analysis of blockchain-based image authentication models," Future Generation Computer Systems, vol. 113, pp. 223–234, 2020.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)