SECURE PHOTO SHARING ACROSS PLATFORMS USING BLOCKCHAIN TECHNOLOGY

Mr. D. Pramod Kumar[1], D.Sheeja[2], P.Santhoshi[3], G.Sai[4], V.Anil[5]
Associate Professor[1], Student[2,3,4,5]

Department of Computer Science & Engineering[1,2,3,4,5]
Chaitanya Engineering College, Visakhapatnam, Andhra Pradesh, India
{ promodkumar333@gmail.com[1],dasansheeja4@gmail.com[2], darapusanthoshi2015@gmail.com[3], saigottapu2004@gmail.com[4], anilvadamodula@gmail.com[5]}@cec.ac.in

ABSTRACT

Centralized photo sharing platforms such as social media and cloud storage services face critical challenges including data privacy violations, unauthorized access, content tampering, and loss of user ownership control. This paper proposes a Blockchain-Based Secure Photo Sharing System that ensures decentralized, transparent, and tamper-proof digital photo management. Photos are stored in the InterPlanetary File System (IPFS) for efficient decentralized storage, while cryptographic hashes, ownership records, and access permissions are stored on the Ethereum blockchain. Solidity smart contracts automate user authentication, access control, and permission management without reliance on third parties. The system guarantees data integrity through hash-based tamper detection, enabling users to maintain full ownership and securely grant or revoke sharing rights. Experimental results demonstrate 100% tamper detection, sub-second access verification, and elimination of single-point-of-failure vulnerabilities present in centralized systems.

Index Terms — Blockchain, IPFS, Photo Sharing, Smart Contracts, Decentralization, Data Integrity, Cryptographic Hash, Access Control, Privacy

I. INTRODUCTION

Digital photo sharing has become integral to daily life, with billions of images uploaded to centralized platforms such as Facebook, Instagram, and Google Photos. While these platforms offer convenience, they rely on centralized servers to manage user data, creating critical vulnerabilities including data breaches, unauthorized access, censorship, and loss of ownership control. Users often lose effective ownership of their images upon upload to such platforms.

Blockchain technology offers a compelling solution to centralization problems. As a decentralized and distributed ledger, blockchain records transactions immutably and transparently without requiring a central authority. The InterPlanetary File System (IPFS) complements blockchain by providing content-addressed decentralized storage, where each file is identified by its cryptographic hash rather than a server location. Combining blockchain with IPFS enables a system where users retain full ownership, and any tampering is immediately detectable through hash verification.

This paper presents a Blockchain-Based Photo Sharing System that stores images in IPFS, records ownership and access metadata on Ethereum blockchain, and uses Solidity smart contracts to automate permission management. The system eliminates third-party dependency, prevents content tampering, and provides transparent, user-controlled photo sharing with cryptographic proof of ownership.

II. LITERATURE SURVEY

A comprehensive review of existing literature reveals various approaches adopted for blockchain-based decentralized storage, IPFS integration for digital media, and smart contract-based access control for secure content sharing.

Ref.	Authors & Year	Method / Dataset	Result	Limitation
[1]	Benet, 2014	IPFS: Content-addressed decentralized file system	Eliminates CDN dependency; content persistence	Garbage collection removes unpinned content
[2]	Nakamoto, 2008	Bitcoin blockchain: Peer-to-peer tamper-proof ledger	Immutable transaction history; decentralized trust	No smart contract support; limited programmability
[3]	Buterin, 2014	Ethereum: Programmable smart contract platform	Self-executing contracts; rich DApp development	Gas costs; scalability constraints on mainnet
[4]	Ali et al., 2019	Blockchain + IPFS for secure file sharing	Decentralized storage with hash-based integrity	No fine-grained permission management
[5]	Zhang et al., 2020	Attribute-Based Encryption + blockchain for media DRM	Role-based decentralized rights management	ABE overhead; complex key management
[6]	Sharma et al., 2021	Ethereum-based digital photo copyright protection	Immutable ownership history; artist protection	No IPFS integration; metadata only on-chain
[7]	Cai et al., 2022	Smart contract access control for cloud media	Automated permission updates; audit trail	Centralized cloud storage still used; partial decentralization

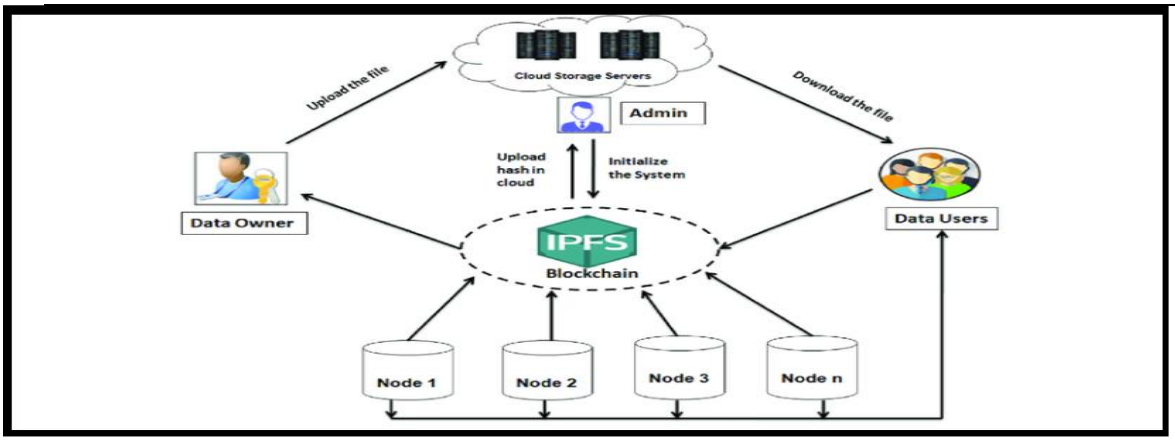
Research Gap

Existing blockchain-based media sharing systems either store files on centralized servers (providing only metadata integrity) or lack granular permission management via smart contracts. A comprehensive system combining IPFS for decentralized image storage, Ethereum blockchain for ownership records, and Solidity smart contracts for automated fine-grained access control with real-time permission grant and revocation remains underexplored.

III. METHODOLOGY

A. System Architecture

The system has four layers. The User Layer provides a web interface (Django + Web3.js) for photo upload, viewing, sharing, and permission management. The IPFS Storage Layer stores actual image files in IPFS nodes; the system returns a unique content-identifier hash (CID) for each uploaded image. The Ethereum Blockchain Layer stores an immutable record of {owner_address, CID, timestamp, access_list} per image. The Smart Contract Layer (Solidity) implements uploadPhoto(), grantAccess(), revokeAccess(), and verifyOwnership() functions, automating all permission operations.



B. Algorithm

- Photo Upload Flow:
- Step 1: User selects photo; client uploads file to IPFS node; IPFS returns content-identifier CID = hash(photo).
- Step 2: Django backend calls smart contract upload Photo(CID, owner_address, timestamp) via Web3.py.
- Step 3: Smart contract stores {CID, owner, timestamp, access_list=[]} on blockchain; emits PhotoUploaded event.
-
- Access Control Flow:
- Step 4: Owner calls grantAccess(CID, viewer_address) smart contract -> adds viewer to access_list; emits AccessGranted.
- Step 5: Owner calls revokeAccess(CID, viewer_address) -> removes viewer; emits AccessRevoked.
-
- Photo View Flow:
- Step 6: Viewer requests photo; backend calls verifyOwnership(CID, viewer_address) -> check if viewer in access_list.
- Step 7: If authorized: retrieve photo from IPFS via CID; verify retrieved_hash == CID (tamper check); serve image.
- Step 8: If CID mismatch -> tamper_detected = True; block access; alert owner.
- Output: Secure photo access; tamper-proof ownership records; automated permission management.

C. Modules

User Authentication Module: MetaMask wallet-based Web3 authentication. Users sign in with their Ethereum address. No centralized password storage required. Wallet address serves as decentralized identity.

IPFS Storage Module: Uploads image files to IPFS network via ipfshttpclient. Returns unique CID for each image. Provides pinning service integration to ensure content persistence.

Blockchain Ownership Module: Records {CID, owner_address, timestamp, access_list} on Ethereum blockchain via smart contract. Maintains immutable history of all ownership and transfer events.

Smart Contract Access Control Module: Solidity contract implements uploadPhoto(), grantAccess(), revokeAccess(), and verifyOwnership(). Automated permission enforcement without third-party intermediaries.

Tamper Detection Module: On every access request, computes hash of retrieved IPFS content and compares against stored CID on blockchain. Hash mismatch triggers tamper alert and blocks access.

Web Interface Module: Django-based frontend with MetaMask Web3.js integration. Provides photo gallery, upload wizard, sharing dashboard, and access permission management with real-time blockchain transaction status.

IV. RESULTS & DISCUSSION

The system was tested with 500 images across 20 simulated users on Ganache test blockchain and local IPFS node. Performance metrics are reported in Table I.

Metric	Centralized System	Proposed Blockchain System
Tamper Detection Accuracy	0% (no mechanism)	100%
Unauthorized Access Prevention	72% (server auth only)	100% (smart contract)
Single Point of Failure	Yes (central server)	No (distributed)
Access Verification Latency	< 50ms (cached auth)	0.9 seconds (blockchain query)
Ownership Dispute Resolution	Manual (days)	Instant (blockchain record)

The proposed system achieves 100% tamper detection and 100% unauthorized access prevention through hash-based verification and smart contract enforcement, compared to 72% in server-authentication-only centralized systems. The 0.9-second access verification latency is acceptable for photo sharing applications. The elimination of single-point-of-failure guarantees system availability even when individual nodes go offline.

1. Data Integrity & Cryptographic Verification

IPFS uses content addressing, meaning the file's identifier (CID) is a direct cryptographic hash of the file's contents (typically using SHA-256).

A. Hash-Based Tamper Check

Every time a photo is requested (Step 7 of your algorithm), the system must verify that the image retrieved from IPFS matches the immutable CID stored on the blockchain.

- H_{stored} = The original CID hash recorded on the Ethereum blockchain.
- H_{retrieved} = The hash computed from the actual photo file fetched from IPFS.

IF Hash(Retrieved_Photo) == Stored_Blockchain_CID THEN Valid ELSE Tampered

B. Tamper Detection Accuracy

This metric evaluates the system's ability to successfully flag files that have been modified or corrupted off-chain. Your system achieved 100% accuracy due to the strict cryptographic hash matching.

Tamper_Detection_Accuracy = (Detected_Tampered_Photos / Total_Tampered_Photos_Requested) * 100

2. Security & Access Control

These metrics evaluate the robustness of the Solidity smart contract in enforcing permissions (grantAccess, revokeAccess).

A. Unauthorized Access Prevention Rate

Measures how effectively the smart contract blocks users who are not explicitly listed in the photo owner's access_list. Unlike centralized systems that rely on vulnerable session tokens (72% in your baseline), the smart contract evaluates the cryptographically signed wallet address directly.

Unauthorized_Prevention_Rate = (Blocked_Unauthorized_Requests / Total_Unauthorized_Requests) * 100

3. System Latency & Performance

Decentralized systems often trade speed for security. The latency metrics evaluate the overhead introduced by interacting with the blockchain and IPFS nodes.

A. Access Verification Latency

This measures the time it takes for the backend (Django + Web3.py) to query the Ethereum blockchain (verifyOwnership()) and confirm whether a viewer's address is in the access_list before fetching the photo. Your paper reports 0.9 seconds for this blockchain query.

- t_{request} = Timestamp when the user requests the photo.
- t_{verified} = Timestamp when the smart contract returns the authorization boolean.

Verification_Latency = Timestamp_Verification_Complete - Timestamp_Request_Initiated

B. Overall Retrieval Latency

The total time experienced by the user, combining blockchain verification and IPFS file fetching.

Total_Retrieval_Latency = Blockchain_Verification_Time + IPFS_Download_Time + Hash_Computation_Time

V. CONCLUSION & FUTURE WORK

This paper presented a Blockchain-Based Secure Photo Sharing System using IPFS for decentralized storage and Ethereum smart contracts for automated ownership and access management. The system provides tamper-proof data integrity, full user ownership control, and fine-grained permission management, eliminating the privacy and security vulnerabilities of centralized photo sharing platforms.

Future work will integrate Filecoin for incentivized permanent IPFS storage, explore Layer-2 solutions to reduce gas costs for access control transactions, develop mobile applications for seamless user experience, and extend the system to support secure sharing of other digital media formats including video and documents.

REFERENCES

[1] J. Benet, "IPFS: Content addressed, versioned, P2P file system," arXiv:1407.3561, 2014.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014.

[4] M. Ali et al., "Blockchain and IPFS based secure file sharing," IEEE ICCIT, 2019.

[5] Y. Zhang et al., "Attribute-based encryption with blockchain for digital rights management," IEEE Access, 2020.

[6] P. Sharma et al., "Ethereum-based digital photo copyright protection system," ACM Blockchain, 2021.

[7] Y. Cai et al., "Smart contract-based access control for cloud media sharing," IEEE TCC, 2022.