



CipherFlow Storage: Encrypted File Exchange across Distributed Content Networks

P. Subramanya Sai¹, Bhukya Niranjana², Akula Tejaswini², A Varshitha², Rangu Prashanth²

¹Assistant Professor, ²UG Student, ^{1,2}Department of Computer Science and Engineering

^{1,2}Vaagdevi Engineering College, Bollikunta, Warangal, 506005, Telangana, India.

ABSTRACT

The exponential increase in digital data exchange and online communication has intensified the need for secure, transparent, and dependable file-sharing systems. Critical information such as financial records, healthcare data, confidential documents, and research outputs is frequently transmitted across distributed networks, where conventional centralized storage models introduce significant vulnerabilities. These traditional systems rely on single-point control, making them prone to data breaches, unauthorized access, service disruptions, and integrity violations. Furthermore, they lack transparency and robust audit mechanisms, raising concerns about data reliability and trustworthiness during storage and transmission. To overcome these limitations, this work proposes a decentralized and secure file-sharing framework that integrates blockchain technology, the Inter-Planetary File System (IPFS), and cryptographic techniques. The system ensures data confidentiality by encrypting files using the Elliptic Curve Integrated Encryption Scheme (ECIES) before storing them in the distributed IPFS network. Instead of placing the actual data on-chain, only essential metadata including file hash, ownership details, timestamps, and access permissions is maintained within a smart contract on the Ethereum blockchain. This design guarantees immutability, traceability, and protection against tampering, while enabling fine-grained access control. In addition to secure storage, the framework incorporates ChaCha20-based symmetric encryption to evaluate and compare computational performance with asymmetric methods. The combination of decentralized storage, cryptographic security, and immutable ledger technology eliminates reliance on centralized authorities, thereby reducing single points of failure and enhancing system resilience. The proposed approach ensures that only authorized entities can access and decrypt shared content while maintaining transparency of file transactions.

Keywords: Blockchain, IPFS, ECIES, ChaCha20, Decentralized File Sharing, Smart Contracts, Data Security.

1. INTRODUCTION

In the modern data-driven environment, the challenges associated with secure storage and efficient access to digital information have become increasingly significant. Managing and sharing time-sensitive files through conventional methods, such as physical storage devices or third-party platforms, is often inefficient and unreliable. As shown as figure 1 these approaches introduce concerns related to data authenticity, security, and confidentiality, as different platforms follow inconsistent management standards. Issues such as duplicate or low-quality data, hidden costs, and the presence of malicious or untraceable files further complicate the ecosystem [1].

Additionally, high infrastructure costs and data isolation during distribution create further obstacles, emphasizing the need for a secure, transparent, and efficient solution that ensures data integrity and trustworthy file handling, especially for sensitive and copyrighted content. In recent years, blockchain-based applications have gained substantial traction across domains such as governance, financial services, electronic documentation, and supply chain systems. Blockchain functions as a decentralized ledger maintained collectively by network nodes, where transactions are securely recorded and



validated. Its core properties, including immutability, transparency, traceability, and resistance to tampering, make it a strong candidate for building reliable peer-to-peer communication systems [2]. The integration of cryptographic techniques, distributed storage, consensus mechanisms, and smart contracts further enhances security and automation.

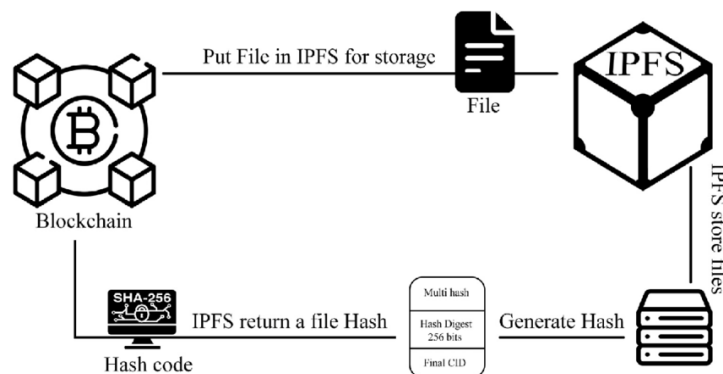


Figure 1: Blockchain and IPFS-based file storage workflow

However, distributed networks still face challenges related to communication efficiency and data dissemination. Content-centric networking approaches like Named Data Networking (NDN) address these issues by enabling data retrieval based on content names rather than host locations, improving efficiency and reducing redundancy. Furthermore, in private blockchain environments, trusted node participation strengthens privacy and controlled access. The concept of distributed ledgers originated with Bitcoin, which introduced a decentralized mechanism for recording transactions without relying on a central authority [3]. Since then, blockchain technology has been widely adopted across multiple industries, including healthcare, energy management, and supply chain systems, due to its ability to ensure secure and transparent data handling [4].

Despite these advantages, directly storing large and frequently updated files on the blockchain is impractical due to scalability limitations. Continuous data updates lead to rapid growth in blockchain size, increasing storage and computational overhead. To address this, distributed storage solutions such as IPFS are utilized, allowing multiple nodes to participate in a unified storage network. By leveraging content-based addressing, IPFS enables efficient data retrieval while minimizing redundancy across the system [5]. This approach significantly reduces the storage burden on the blockchain by offloading large files while retaining only essential metadata on-chain, thereby achieving a balanced and efficient storage architecture.

2.LITERATURE SURVEY

Sangeeta et al. [6] proposed a decentralized IPFS and blockchain-based application for distributed file storage. It is possible to upload various types of files into their decentralized application (DApp), and hashes of the uploaded files are permanently saved on the Ethereum blockchain with the help of smart contracts. Because it cannot be removed, it is immutable. By clicking on the file description, they can also view the file. DApp also includes a keyword search feature to assist us in quickly locating sensitive information. They used Ethers.js' smart contract event listener and contract. query Filter to filter and read data from the blockchain. The smart contract events are then written to a text file for their DApp's keyword search functionality. Sultana et al. [7] utilized a blockchain network, where each academic certificate is stored as a digital asset on the blockchain. These digital certificates are cryptographically secured, timestamped, and associated with unique identifiers, such as hashes or public keys, ensuring their integrity and immutability. Anyone with access to the blockchain network can verify a certificate's authenticity, using the MetaMask extension and Ethereum network, eliminating the need for intermediaries and reducing the risk of fraudulent credentials.



Shahzad et al. [8] proposed a decentralized medical image-sharing framework to address these issues by integrating blockchain technology, the Interplanetary File System (IPFS), and edge computing. Blockchain technology enforced secure patient-centric access control through smart contracts that enable patients to directly manage their data-sharing permissions. The IPFS provides decentralized and scalable storage for medical images and effectively resolves the storage limitations associated with blockchain. Edge computing enhances system responsiveness by significantly reducing latency through local data processing to ensure timely medical image access. Robust security is ensured by using elliptic curve cryptography (ECC) for secure key management and the Advanced Encryption Standard (AES) for encrypting medical images to protect against unauthorized access and data breaches. Fernández-Blanco et al. [9] integrated the IPFS protocol with Distributed Ledger Technologies (DLTs) can enable pure decentralized storage subsystems for dApps, avoiding single points of failure and ensuring data integrity and security. At the same time, as an external source of information, oracles are required to ensure data correctness while managing IPFS data. Despite the potential use of such components in real use cases, the current literature lacks detailed oracle implementations designed to interact with the IPFS protocol. To tackle such an issue, this article presents two open-source use cases that integrate smart contracts, an oracle and an IPFS-based storage subsystem that will allow future professors, students, researchers and developers to learn and experiment with advanced dApps and DLTs.

Bin Saif et al. [10] required generally enough to guarantee immutability and traceability, but misses to support another important property, that is, data availability. This is particularly true when a traditional, centralized database is chosen for off-chain storage. For this reason, many proposals tried to properly combined blockchain with decentralized IPFS storage. However, the storage of data on IPFS could pose some privacy problems. Belfqih et al. [11] presented a decentralized authentication protocol leveraging blockchain technology and the IPFS data management framework to provide secure and real-time communication between IoT devices. Using the Ethereum blockchain, smart contracts, elliptic curve cryptography, and ASCON encryption, the proposed protocol ensures the confidentiality, integrity, and availability of sensitive IoT data. The mutual authentication process involves the use of asymmetric key pairs, public key registration on the blockchain, and the Diffie–Hellman key exchange algorithm to establish a shared secret that, combined with a unique identifier, enables secure device verification.

Naz et al. [12] proposed a blockchain-based secure data sharing platform by leveraging the benefits of IPFS. A meta data is uploaded to IPFS server by owner and then divided into n secret shares. The proposed scheme achieved security and access control by executing the access roles written in smart contract by the owner. Users are first authenticated through RSA signatures and then submit the requested amount as a price of digital content. After the successful delivery of data, the user is encouraged to register the reviews about data. Sayed et al. [13] examined practical methods for safely accessing patient data from any location at any time, with a special focus on IPFS servers and blockchain technology in addition to group signature encryption. Essential processes like maintaining the confidentiality of medical records and safe data transmission could be made easier by these technologies. Their proposed framework enables secure, remote access to patient data while preserving accessibility, integrity, and confidentiality using Ethereum blockchain, IPFS, and group signature encryption, demonstrating hospital-scale scalability and efficiency. Experiments show predictable throughput reduction with file size ($200 \rightarrow 90$ tps), controlled latency growth ($90 \rightarrow 200$ ms), and moderate gas increase ($85k \rightarrow 98k$), confirming scalability and efficiency under varying healthcare workloads.



Merlec et al. [14] presented a systematic comparative analysis of decentralized storage systems, emphasizing their potential to enhance sustainable DSS. By highlighting the integral role of blockchain technology, this study critically examined various decentralized storage platforms, including Arweave, BitTorrent, Dat Protocol, File coin, Hyper core Protocol, IPFS, MaidSafe, Sia, Storj, and Swarm. The analysis covered the key architectural features of these systems, their performance metrics, and their contribution to user data sovereignty. Khan et al. [15] attempted to review the security of cloud systems when using the blockchain Ethereum, and cloud computing was briefly discussed with its advantages and disadvantages. The idea of a decentralized cloud was briefly demonstrated with blockchain technology. Furthermore, previous papers were reviewed and presented in tabular form. This dictated that there are still research gaps in the field of blockchain-based cloud computing systems.

3. PROPOSED SYSTEM

The proposed system architecture integrates a web-based interface, backend processing, decentralized storage, and blockchain technology to enable secure file sharing and access control. The user interacts with the system through a Django-based web application, where files are uploaded, viewed, or requested for access. The backend handles encryption using ChaCha20 for data security and ECIES for key management, along with serialization and performance tracking. Encrypted files are stored in IPFS, which generates a unique content identifier for retrieval. The blockchain layer stores metadata such as hash, encrypted key, nonce, permissions, and timestamps using smart contracts. Access control mechanisms ensure that only authorized users can read or download files based on permissions. During retrieval, encrypted data is fetched from IPFS and decrypted using stored keys. The system also includes performance visualization comparing encryption techniques. The overall design ensures confidentiality, integrity, decentralization, and controlled access to shared data as illustrated in Figure 2.

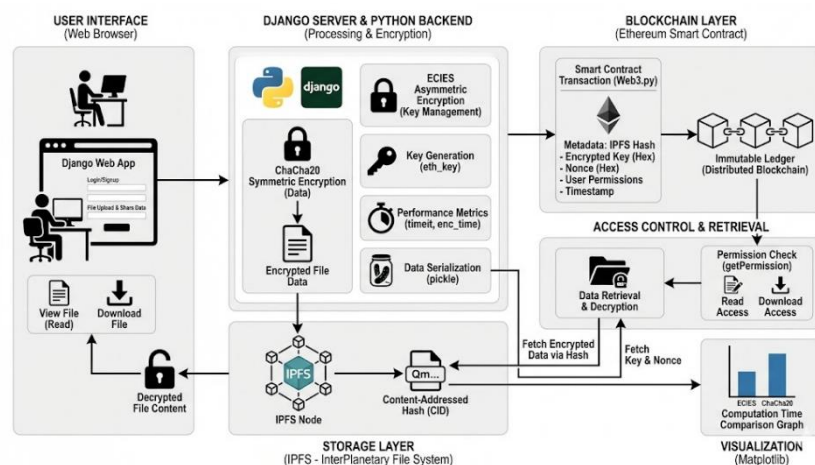


Figure 2: Proposed System architecture

User Interaction and File Submission: The process begins with the user accessing the Django web interface through a browser, where they can upload files, share data, or request access. User authentication ensures that only registered users can interact with the system. The interface also provides options to view or download files based on granted permissions. This layer acts as the entry point for all operations in the system.

Data Processing and Encryption: Once a file is uploaded, the backend processes the data by converting it into a serialized format for efficient handling. The system applies ChaCha20 symmetric encryption to secure the file content before storage. Additionally, ECIES is used for managing cryptographic keys, ensuring secure key exchange. Performance metrics such as encryption time are also recorded during this stage.



Storage in IPFS Network: After encryption, the file is stored in the IPFS distributed storage system. IPFS generates a unique content identifier (CID) based on the file content, which ensures integrity and eliminates duplication. This decentralized storage approach reduces dependency on centralized servers and enhances data availability. Only the encrypted version of the file is stored, ensuring security even in distributed environments.

Blockchain Metadata Management: The system records essential metadata on the blockchain using smart contracts. This includes the IPFS hash, encrypted key, nonce, ownership details, access permissions, and timestamps. Since blockchain data is immutable, it prevents tampering and ensures transparency. This layer acts as a secure ledger for tracking all file-sharing activities and access rights.

Access Control and Permission Handling: When a user requests access to a file, the system verifies permissions through the blockchain. If the file is private, the owner must approve the request, assigning either read or download rights. The permission mechanism ensures controlled access and prevents unauthorized usage. This step enforces security policies and maintains data privacy.

Data Retrieval and Decryption: Upon successful authorization, the system retrieves the encrypted file from IPFS using the stored hash. The corresponding key and nonce are fetched from the blockchain metadata to decrypt the file. The decrypted content is then presented to the user either for viewing or downloading. This ensures that only authorized users can access the original data securely.

3.1 Proposed workflow

The proposed workflow as shown in Figure 3 integrates encryption, blockchain, and distributed storage to create a secure and tamperproof file-sharing environment. When a user interacts with the system, every action from uploading a file to retrieving it is processed through layers of encryption, permission validation, and decentralized storage. This ensures that files remain confidential, access is fully controlled, and all activities are transparently recorded on the blockchain for auditability and trust.

User Login and Authentication: The user begins by logging into the Django application, which verifies credentials stored securely on the blockchain. This ensures that identity validation is decentralized and tamper resistant. Once authenticated, the user gains access to the main dashboard of the system.

File Upload and Serialization: When the user uploads a file, the system reads and converts it into serialized byte data to prepare it for encryption. Serialization ensures compatibility with both encryption modules and distributed storage. This creates a uniform structure for further processing.

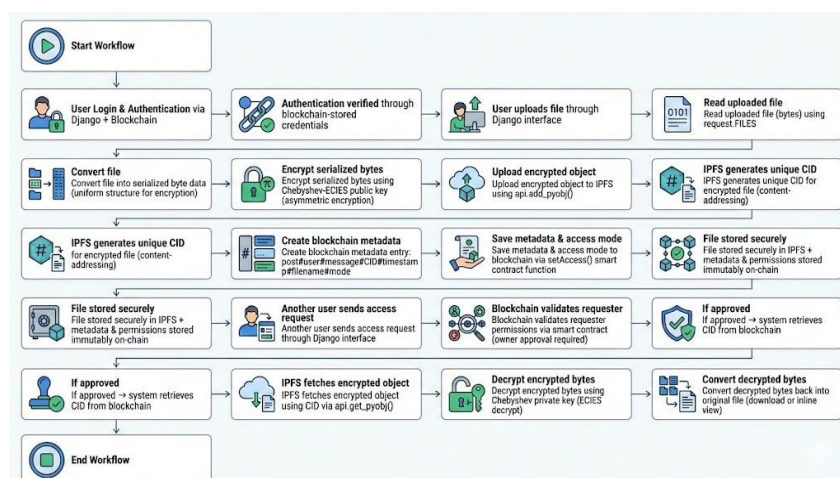


Figure 3: proposed internal workflow



ECIES Encryption: The serialized file is encrypted using ECIES, which uses public-key cryptography to secure the content. Only the private key holder can decrypt the data, guaranteeing confidentiality even after storage. This encryption protects the file before it leaves the user's environment.

Upload Encrypted File to IPFS: The encrypted data is sent to IPFS, which splits it into chunks and distributes them across multiple network nodes. IPFS generates a unique content identifier (CID) that represents the encrypted file immutably. This CID acts as the retrieval key for future access.

Blockchain Storage of File Metadata: The system stores file metadata—such as filename, CID, timestamp, owner, and access mode on a blockchain smart contract. This ensures permanent, tamperproof tracking of file operations without involving centralized servers. All permission logic is also handled through blockchain records.

Access Request and Permission Validation: If another user wants to view or download the file, they initiate an access request through the application. The blockchain verifies whether the requester has been granted permission by the owner. This validation prevents unauthorized file retrieval or misuse.

Retrieve Encrypted File from IPFS: Once access is approved, the system uses the stored CID to fetch encrypted chunks from IPFS nodes. IPFS automatically reassembles the data using its Merkle DAG structure. The encrypted file is then delivered to the requester for decryption.

Decryption Using Private Key: The requester decrypts the received encrypted file using their private key, restoring the original content safely. This ensures that only legitimate users can access readable data, even though the file was stored publicly in a distributed environment. The confidentiality of the content remains fully preserved.

ChaCha20 Encryption for Performance Comparison: Parallel to ECIES, the system also encrypts the serialized file with the ChaCha20 algorithm. This enables benchmarking between asymmetric and symmetric encryption. The recorded encryption times help evaluate communication overhead and efficiency.

Result Logging and Transparency: Every action upload, permission changes, access requests, and retrievals is logged immutably on the blockchain. This creates a transparent audit trail that prevents disputes or unauthorized tampering. The system thus maintains accountability and trust for all interactions.

4. RESULTS ANALYSIS

The results demonstrate the successful implementation of a secure and decentralized file-sharing system that integrates blockchain, IPFS, and cryptographic techniques. Files are encrypted before storage, ensuring confidentiality, while IPFS provides efficient and reliable distributed storage using content-based addressing. Blockchain maintains immutable metadata and access control, enabling transparent and tamper-proof tracking of file operations. The system effectively enforces permission-based access, allowing only authorized users to read or download files. Performance evaluation shows that ChaCha20 offers faster encryption compared to ECIES, highlighting a trade-off between speed and security. The proposed framework achieves improved data integrity, security, and system reliability compared to traditional centralized approaches.

Figure 4 illustrates the viewing of shared messages along with metadata such as IPFS hash, timestamps, and file names. It presents approximately 4–5 attributes associated with each shared file. The figure highlights how users can track and verify stored data efficiently. It demonstrates the use of hash values as unique identifiers in decentralized storage. The system maintains a record of shared files for easy retrieval. It reflects transparency and traceability in file sharing operations.



© 2026 All rights are reserved

Data Owner	Shared Message	IPFS File Address	Shared Date Time	Shared File Name	Access Control	Download File
prasad	hello	QmeA1NvmsDQcgrT1mW9EjRDvEqzmQNUwayf3PxeXGVpfe	2026-03-25 14:42:30	A5. Marine Secure Share vault.docx	Public	Download File
prasad	bye	Qmcap7z5woCG9yREYMHNN4AzzGmnVw7AoApiNoXmimzE63	2026-03-25 14:43:07	TICKET.pdf	Private	Download File
prasad	okay bbye	QmW1VAVSGuK4epjp4F8qBAQjLUTg8cuzWkz3K4HRLTZf	2026-03-25 14:43:27	A3. BLOCKCHAIN-BACKED ZERO TRUST NETWORKING FOR SECURE EVENT LOGGING AND POLICY-GUIDED DATA ACCESS.docx	Public	Download File

Figure 4: Viewing Shared Messages Screen

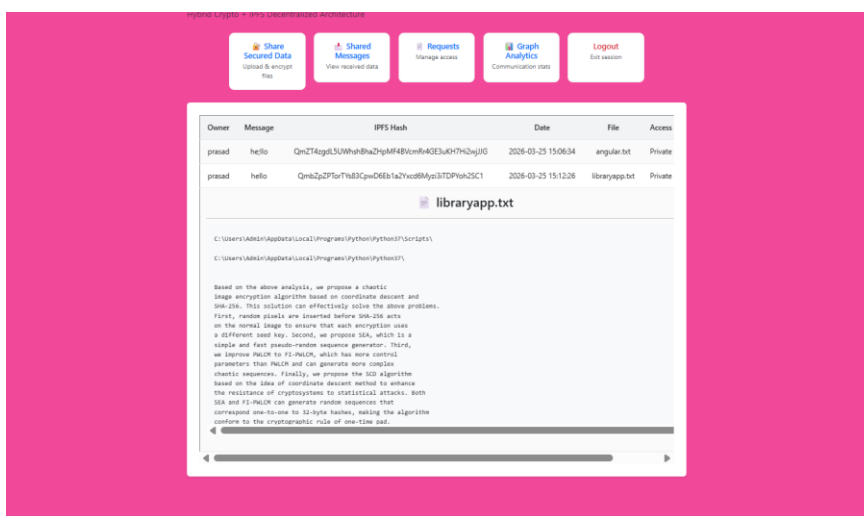


Figure 5: Reading File Screen

Figure 5 depicts the file reading interface where authorized users can access and view decrypted content. It illustrates the retrieval of files using a unique IPFS hash value. The figure shows how encrypted data is converted back into readable format after validation. It reflects the completion of the secure data transmission cycle. The system performs around 2–3 operations including verification, decryption, and display. It ensures that only authorized users can view the content.

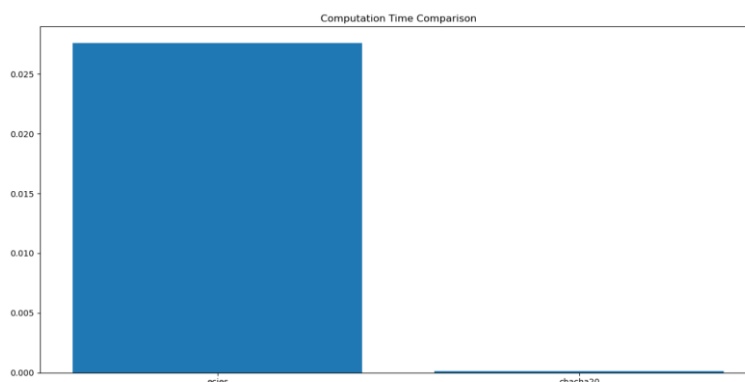


Figure 6: Computation Time Graph Screen

Figure 6 illustrates the computation time comparison graph between different cryptographic algorithms used in the system. It demonstrates performance differences where one algorithm shows significantly



lower execution time than the other. The figure highlights that computation time is measured in milliseconds for accuracy. It reflects how lightweight algorithms can reduce processing overhead by a noticeable margin between the ecies and Chacha20. It shows how performance impacts real-time system efficiency.

5. CONCLUSION

The proposed system effectively establishes a secure and decentralized framework for data sharing by integrating Django, blockchain, IPFS, and cryptographic techniques. By maintaining metadata on the blockchain while storing encrypted files in IPFS, the system ensures data integrity, immutability, and optimized storage utilization. The adoption of ECIES strengthens data confidentiality, whereas blockchain-based mechanisms enable transparent and reliable access control. Performance analysis indicates that the implemented encryption approach provides strong security with acceptable execution time, and the comparison with ChaCha20 reveals opportunities for further improvement in computational efficiency. The overall design minimizes reliance on centralized systems and enables controlled, secure data access within a distributed environment.

REFERENCES

- [1] Khalid, U.; Asim, M.; Baker, T.; Hung, P.C.; Tariq, M.A.; Rafferty, L. A decentralized lightweight block-chain-based authentication mechanism for IoT systems. *Clust. Comput.* 2020, 23, 2067–2087.
- [2] Yang, W.; Aghasian, E.; Garg, S.; Herbert, D.; Disiuta, L.; Kang, B. A survey on blockchain-based internet service architecture: Requirements, challenges, trends, and future. *IEEE Access* 2019, 7, 75845–75872.
- [3] Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, Rome, Italy, 1–4 December 2009.
- [4] Afanasyev, A.; Shi, J.; Zhang, B.; Zhang, L.; Moiseenko, I.; Yu, Y.; Shang, W.; Huang, Y.; Abraham, J.P.; DiBenedetto, S.; et al. *NFD Developer's Guide; NDN Project Technical*; NDN: Los Angeles, CA, USA, 2014.
- [5] Purmani, S. S. R. (2025). Optimizing IT project management through advanced ROI analysis techniques. *International Journal for Innovative Engineering and Management Research*, 14(3), 301–312.
- [6] N. Sangeeta; Nam, S.Y. Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability. *Electronics* 2023, 12, 1545. <https://doi.org/10.3390/electronics12071545>
- [7] Sultana, S.A.; Rupa, C.; Malleswari, R.P.; Gadekallu, T.R. IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field. *Information* 2023, 14, 446. <https://doi.org/10.3390/info14080446>
- [8] Shahzad, A.; Chen, W.; Zhang, Y.; Kumar, R. Zero-Trust Medical Image Sharing: A Secure and Decentralized Approach Using Blockchain and the IPFS. *Symmetry* 2025, 17, 551. <https://doi.org/10.3390/sym17040551>
- [9] Fernández-Blanco, G.; García-Cereijo, P.; Fernández-Caramés, T.M.; Fraga-Lamas, P. Hands-On Blockchain Teaching and Learning: Integrating IPFS and Oracles Through Open-Source Practical Use Cases. *Educ. Sci.* 2025, 15, 1229. <https://doi.org/10.3390/educsci15091229>



- [10] Bin Saif, M.; Migliorini, S.; Spoto, F. Efficient and Secure Distributed Data Storage and Retrieval Using Interplanetary File System and Blockchain. *Future Internet* 2024, 16, 98. <https://doi.org/10.3390/fi16030098>
- [11] Belfqih, H.; Abdellaoui, A. Decentralized Blockchain-Based Authentication and Interplanetary File System-Based Data Management Protocol for Internet of Things Using Ascon. *J. Cybersecur. Priv.* 2025, 5, 16. <https://doi.org/10.3390/jcp5020016>
- [12] Naz, M.; Al-zahrani, F.A.; Khalid, R.; Javaid, N.; Qamar, A.M.; Afzal, M.K.; Shafiq, M. A Secure Data Sharing Platform Using Blockchain and Interplanetary File System. *Sustainability* 2019, 11, 7054. <https://doi.org/10.3390/su11247054>
- [13] Sayed, S.; Famous, M.S.; Mazumder, R.; Khan, R.T.; Kaiser, M.S.; Hossain, M.S.; Andersson, K.; Khondoker, R. Blockchain and InterPlanetary Framework for Decentralized and Secure Electronic Health Record Management. *Blockchains* 2025, 3, 12. <https://doi.org/10.3390/blockchains3040012>
- [14] Merlec, M.M.; In, H.P. Blockchain-Based Decentralized Storage Systems for Sustainable Data Self-Sovereignty: A Comparative Study. *Sustainability* 2024, 16, 7671. <https://doi.org/10.3390/su16177671>
- [15] Khan, N.; Aljoaey, H.; Tabassum, M.; Farzamnia, A.; Sharma, T.; Tung, Y.H. Proposed Model for Secured Data Storage in Decentralized Cloud by Blockchain Ethereum. *Electronics* 2022, 11, 3686. <https://doi.org/10.3390/electronics11223686>