IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155216

SafeDox: A Review on Secure Document Sharing Using Blockchain, IPFS, and End-to-End Encryption

Neeharika Shakthivelan[1], Badareesh P[2], C S Jeevan Setty[3], Poorvi P[4], Mrs. Poornima H N[5]

Dept. of CSE (ICB), K. S. Institute of Technology, Karnataka, India[1-5]

Abstract: The need for the development of more efficient document-sharing mechanisms has been brought about by an increased demand for secure data sharing due to developments in cloud computing and digital communication systems. Traditional centralized platforms for data sharing have several limitations such as data leaks, insecurity, single point of failure, and non-transparency. Hence, the need for developing advanced technology like blockchain, IPFS, and encryption is paramount. In the current review article, the discussion has been focused on various methods of secure document-sharing platforms, including blockchain, IPFS, and encryption algorithms. Through blockchain, it is possible to achieve data integrity, data tampering, transparency, and data authentication. Besides, IPFS enables users to employ decentralized storage services that can store huge amounts of data. Similarly, various algorithms for secure encryption like AES-256 and ECC have been considered as they ensure encryption keys and data transfers. Further, different studies and approaches have been compared in terms of methods, merits, drawbacks, and security considerations. Finally, some areas that have not been explored adequately like privacy protection, access control, scalability and complexity of the system have been highlighted as gaps. Consequently, a SafeDox framework has been proposed.

Keywords: Blockchain, IPFS, Secure Document Sharing, AES-256, ECC, Smart Contracts, End-to-End Encryption, Decentralized Storage.

I. INTRODUCTION

Due to the rapid development of cloud storage and communication platforms, there is an increasing need for secure methods to exchange documents among users [1]. Traditional centralized systems face several challenges such as data leak-age, weak access control, single-point failure, and lack of transparency [2, 3]. These issues can be addressed through the integration of Blockchain, Inter-Planetary File System (IPFS), and advanced encryption techniques in modern document-sharing systems [4].

This review paper presents a comprehensive study of blockchain-based document-sharing systems and their security using IPFS and end-to-end encryption. Blockchain technology ensures data integrity, transparency, authentication, and protection against data tampering [5], whereas IPFS enables de-centralized and efficient file storage [6]. Cryptography methods that can be used to guarantee efficient data transfer and key management include AES-256 and ECC [7, 8]. Besides this, the comparative analysis of recent studies is conducted on the basis of research methods, results, limitations, and security aspects [9]. Research gaps related to privacy, access control, scalability, and implementation challenges have been found in the literature review. Moreover, SafeDox framework employing blockchain and IPFS technologies will be presented.

II. LITERATURE SURVEY

Steichen et al [1] proposed a blockchain-based decentralized access control mechanism integrated with IPFS using Ethereum smart contracts. The system managed an on-chain access control list and enforced file permissions at the node level. Although the framework improved transparency and re-moved the need for a central authority, it lacked file-level encryption and did not support dynamic access revocation. Metadata leakage through DHT traffic was also a concern.

Zhao [2] proposed privacy-preserving federated learning framework that made use of blockchain and IPFS in order to store the models in decentralized way and aggregate them respectively. This architecture achieved anonymity amongst participants but it did not include any feature related to document sharing and had no concern with persistence of storage.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

Chen et al. [3] proposed threshold proxy re-encryption technique by exploiting the idea of blockchain and distributing the proxy nodes for secure data exchange between different entities in the system. In this technique, key handover took place without revealing the decryption key of the sender. It had high latency because of proxy node interaction. Proposed by Athanere and Thakur [4], semi-decentralized hierarchical data exchange systems use the CP-ABE encryption scheme together with IPFS. In this case, the attribute authorities are multiple and not one. It ensures fine-grained access control of files.

Goh [5] introduced the blockchain-based federated learning framework where the Ethereum smart contract and IPFS coordination mechanism were used. Its weaknesses were a high gas consumption due to a high number of interactions and scalability issues associated with the public Ethereum network.

Sangeeta and Nam [6] implemented a decentralized vehicular network based on the integration of Ethereum, IPFS, and Meta-Mask. Efficiency regarding performance and security of off-chain storage and on-chain hashing was achieved. Nevertheless, a weakness of this implementation was that any user who had a Content Identifier could access the file because there were no access restrictions apart from a hash value.

Mathwale and Ramisetty [7] created a file sharing mechanism based on the Hyperledger Fabric, IPFS, and public-key cryptography for transferring documents between organizations. Permissioned blockchains provided faster consensus time compared to other systems. However, the disadvantage of the proposed mechanism was that it required a full consortium including the Certificate Authority and lacked mobile clients' support.

Sonkamble [8] provided a blockchain-based and IPFS-based solution for EHRs which allowed patients to manage the access control. It was ensured that sensitive data would be accessible only by authorized users. Scalability and interoperability with other systems stayed unsolved issues.

Jadhav et al. [9] offered the decentralized storage of documents with help of hybrid encryption (AES and ECC), Ethereum, and IPFS technologies. They demonstrated that AES-ECC encryption guarantees a very high level of confidentiality, while computation is still efficient. View counting problem was beyond their interests, and mobile solution was not presented.

Sameera [10] reviewed various privacy preservation strategies applied in blockchain based federated learning, i.e., HE, SMPC, DP, etc. It was purely theoretical work without any real-world implementations and tests.Cai et al. [11] suggested applying CP-ABE and zero-knowledge proofs allowing to check an access policy while hiding users' attributes. It was noted that their approach offered advanced encryption and privacy-preserving policy validation. At the same time, high complexity made mobile implementation impossible.

Proxy re-encryption, attribute-based encryption, and blockchains were considered by Jabri et al. [12] in their works for protecting data sharing in IoT scenarios. Moving the process of re-encryption to blockchain proxies allowed achieving better theoretical security because of limited resources of those devices. However, it remained to be just a theoretical idea which was rather expensive in terms of overheads and had no implementation.

In his survey [13], Nguyen reviewed the literature on blockchain-based data sharing systems and classified them according to the type of consensus mechanism, storage approach, and access control technique applied. The survey confirmed that scalability, consistency of access control techniques, and device mobility remain three major problems to solve. No attempt was made to solve those problems.

Rangwala et al. [14] provided a survey taxonomy of blockchain-based federated learning systems that addressed the compromise between decentralization, storage efficiency, and transaction throughput. Moreover, the paper identified the size of transactions and on-chain storage as two barriers to applying this technology. The paper discussed federated learning frameworks but not document sharing systems.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155216

TABLE I. LITERATURE REVIEW – SUMMARY TABLE

Sl. No.	Paper	Authors	Year	Methodology	Findings	Limitations
[1]	Blockchain-based, Decentralized Access Control for IPFS	M. Steichen, B. Fiz, R. Norvill, W. Shbair, R. State	2018	Blockchain-based decentralized access control integrated with IPFS using Ethereum smart contracts; maintained on-chain access control list and enforced file permissions at the node level.	Improved transparency and removed the need for a central authority in managing IPFS file access.	No file-level encryption; no dynamic access revocation; metadata leakage through DHT traffic.
[2]	Privacy-Preserving Blockchain-Based Federated Learning	Y. Zhao	2021	Privacy-preserving federated learning framework using blockchain and IPFS for decentralized model storage and aggregation.	Achieved anonymity among participants in the federated learning process.	No document sharing feature; no concern with storage persistence.
]3]	Threshold Proxy Re-Encryption for Secure IoT Data Sharing via Blockchain	Y. Chen, B. Hu, H. Yu, Z. Duan, J. Huang	2021	Threshold proxy re-encryption using blockchain and distributed proxy nodes for secure data exchange; key handover without revealing the sender's decryption key.	Enabled secure data exchange between entities without exposing decryption keys.	High latency due to proxy node interaction.
[4]	Hierarchical Semi-Decentralized Blockchain and IPFS Approach for Secure Data Sharing	S. Athanere, R. Thakur	2022	Semi-decentralized hierarchical data exchange using CP-ABE encryption with IPFS and multiple attribute authorities.	Achieved fine-grained access control of files with distributed attribute authorities.	Not explicitly detailed in the reviewed text; reliance on multiple attribute authorities adds complexity.
[5]	Blockchain-Enabled Federated Learning Architecture	E. Goh	2023	Blockchain-based federated learning framework using Ethereum smart contracts and IPFS as a coordination mechanism.	Demonstrated coordination of federated learning via blockchain and IPFS.	High gas consumption due to a large number of on-chain interactions; scalability issues on the public Ethereum network.
[6]	Blockchain and IPFS Storage for Vehicular Networks	N. Sangeeta, S. Y. Nam	2023	Decentralized vehicular network using Ethereum, IPFS, and MetaMask; off-chain storage with on-chain hashing.	Achieved performance efficiency and security through off-chain storage and on-chain hashing.	Any user with a Content Identifier (CID) can access the file; no access restrictions beyond the hash value.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

Sl. No.	Paper	Authors	Year	Methodology	Findings	Limitations
[7]	Secure File Sharing Using Blockchain	R. Mathwale, R. Ramisetty	2023	File sharing mechanism using Hyperledger Fabric, IPFS, and public-key cryptography for inter-organizational document transfer.	Permissioned blockchain provided faster consensus time compared to public blockchain systems.	Requires full consortium setup including Certificate Authority; lacks mobile client support.
[8]	Secure Data Transmission of Electronic Health Records Using Blockchain	R. G. Sonkamble	2023	Blockchain and IPFS-based solution for Electronic Health Records (EHRs) with patient-controlled access.	Ensured sensitive data is accessible only by authorized users.	Scalability and interoperability with other systems remain unsolved.
[9]	Decentralized Document Storage Using IPFS	S. Jadhav, G. Choudhari, R. Bura, V. Bhosale, M. Bhavik	2023	Hybrid encryption (AES + ECC) combined with Ethereum and IPFS for decentralized document storage.	AES-ECC encryption demonstrated high confidentiality with computationally efficient performance.	View counting not addressed; no mobile solution presented.
[10]	Privacy in Blockchain-Based Federated Learning Systems: A Survey	K. M. Sameera	2024	Survey of privacy preservation strategies in blockchain-based federated learning, including HE, SMPC, and DP.	Provided a comprehensive theoretical review of privacy techniques applicable to federated learning.	Purely theoretical; no real-world implementations or experimental tests.
[11]	Attribute-Based Encryption with Zero-Knowledge Proof	D. Cai, B. Chen, L. Zhang, K. Li, H. Kan	2024	CP-ABE combined with zero-knowledge proofs to verify access policies while hiding user attributes.	Offered advanced encryption and privacy-preserving policy validation.	High computational complexity makes mobile implementation impractical.
[12]	Blockchain and Proxy Re-Encryption for IoT Data Sharing	A. Jabri, C. Drocourt, M. Azizi, G. Utard	2025	Proxy re-encryption, attribute-based encryption, and blockchain for IoT data sharing; re-encryption offloaded to blockchain proxies.	Theoretically improved security by moving re-encryption to blockchain proxies, accounting for IoT device resource limitations.	Remains theoretical with no implementation; expensive in terms of overheads.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

Sl. No.	Paper	Authors	Year	Methodology	Findings	Limitations
[13]	Blockchain-Empowered Trustworthy Data Sharing: Fundamentals and Challenges	T. L. Nguyen	2025	Literature survey classifying blockchain-based data sharing systems by consensus mechanism, storage approach, and access control technique.	Confirmed that scalability, consistent access control, and device mobility are the three major unsolved problems.	Survey only; no proposed solutions to identified problems.
[14]	Blockchain-Enabled Federated Learning: A Comprehensive Survey	M. Rangwala, K. R. Venugopal, R. Buyya	2025	Survey taxonomy of blockchain-based federated learning systems analyzing decentralization, storage efficiency, and transaction throughput.	Identified transaction size and on-chain storage as two primary barriers to adoption.	Focused on federated learning frameworks only; not applicable to document sharing systems.

A. BLOCKCHAIN FUNDAMENTALS

A blockchain is a peer-to-peer distributed database of records stored in blocks, where each record is secured using cryptographically secure hashes. It should be noted that once the block is recorded in the database, no individual or authority can modify or delete any record, thus making the database tamper-proof by design. As mentioned by Nguyen, there are two major advantages of blockchain for trusted data exchange – immutability and decentralization. Full document access control of SafeDox is based on Ethereum blockchain on Sepolia testnet. Ethereum has introduced the idea of smart contracts within the blockchain framework. Smart contracts are self-executing codes or scripts that are written on the blockchain network and run autonomously in a deterministic fashion without requiring any human interference. Steichen et al. state that smart contracts have the capability of serving as a decentralized access control mechanism that does not require the involvement of a third party.

There are six categories within each file in SafeDox, which include IPFS CID, Ethereum addresses of sender and recipient, maximum views, remaining views, and revocation status. The lifecycle of the smart contract consists of three steps: shareDocument(), which writes the CID and permission settings permanently on the blockchain; On the other hand, accessDocument() identifies the caller, subtracts one view atomically while providing the CID; subsequently, the revokeAccess() is performed. Mathwale and Ramisetty have substantiated this claim. The key advantage over convenational Access Control Lists is trust less enforcement: the contract is the enforcer, auditable by anyone and over ridable by no one. Identity is verified through Ethereum's msg.sender mechanism, making address impersonation computationally infeasible. Sonkamble [8] observed that blockchain-based ac-cess control provides end-to-end accountability that centralised models cannot match. Even after access is revoked, the on-chain record persists as a permanent, tamper-proof audit trail The main advantage over traditional Access Control Lists is that enforcement is trust less: the contract is the enforcer, auditable by anyone and unoverrideable by anyone. On Ethereum msg.sender is verified for identity which is not possible for impersonating address. Sonkamble [8] pointed out that ac-cess control in blockchain provides end-to-end accountability which is not possible in centralised models.

B. IPFS FUNDAMENTALS

IPFS (Inter Planetary File System) utilizes decentralized peer to peer networking and storage, where each file within the system is identified via content (the SHA-256 hash of that file). In their research, Daniel and Tschorsch found that IPFS's goals of being resilient and detecting tampering were superior when compared to an HTTP-based system. Files can be retrieved from any node within the network that has a copy of that file, meaning there is no single failure point in retrieving any particular document or object; rather, there will always be multiple copies hosted by different nodes on the network. The integrity of files is guaranteed because CIDs have the following properties: The integrity of files is ensured because CIDs have the following properties:

- The CIDs Are Deterministic: They will generate the same CID if the input remains the same.
- The Avalanche Effect: They generate a new CID if there is a one-bit difference between any two documents.
- Functionality of CIDs: It is almost impossible to find out what is the content if one knows the CID.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

• Collision Resistance of CIDs: It is next to impossible to get the same CID of two distinct documents.

Every time a new document is uploaded on the SafeDox platform, the unique CID is stored within the smart contract.When a new file is uploaded into SafeDox, the unique CID is stored in the smart contract. Upon downloading a file from SafeDox, the application will recompute and verify the CID for that file against the CID stored in the blockchain smart contract. If an attacker modifies that file, an unauthorized node stores the file, or a storage provider modifies the file, the computed CID and the on-chain CID will be different, thus causing that file to be rejected by the application. Huang et al. [15] have confirmed that this method provides a valid guarantee of integrity, while Jadhav et al. [9] have shown that an IPFS-based document is more resilient to modifications and more available than a document stored in a centralized manner. SafeDox integrates with Pinata as their managed IPFS pinning service, which allows SafeDox to offer persistent avail-ability without requiring a self-hosted IPFS node. Athanere & Thakur [4] have confirmed that uploading large files to IPFS with only a content hash copied to the blockchain is the most scalable design for blockchains that share data using IPFS storage; this is the model that SafeDox follows. Additionally, if you were to compromise an entire node on IPFS, you would not have any usable information to provide; this is because you would only have access to AES-256 encrypted cipher text.

C. ENCRYPTION AND SECURITY MECHANISMS

SafeDox makes use of a combination of symmetric and asymmetric cryptography to attain real E2EE encryption. All the encryption is performed on the device from where the messages are sent out. Neither the server nor the IPFS node gets to see any form of plaintext or unencrypted keys. Encryption is done using AES-256, which is the NIST-approved cipher having 2^{256} possibilities for keys, ensuring that brute force attacks are computationally impractical. SafeDox employs Fernet, which is AES-256-CBC together with HMAC-SHA256 authentication. The use of CBC mode involves encrypting each block based on its predecessor's cipher text, thereby concealing patterns within the plaintext while thwarting statistical attacks. The HMAC provides the security guarantee that any alteration to the cipher text will be detected immediately during decryption and trigger an exception, precluding silent modifications. For the encryption and transmission of the AES key to the recipient, the proposed framework utilizes Elliptic Curve Cryptography (ECC). The users will be allocated ECC pairs whereby the public key will be submitted to the back end, and the private key will be stored in the keychain of the device's hardware and never sent out. ECIES will then be used to wrap the AES session key using the public ECC key of the receiver, with the receiver being able to unwrap the key using his private ECC key. ECC-256 was chosen rather than RSA as it offers comparable security as RSA-3072, but with smaller keys and faster computations on mobiles. This decision was made after alternatives like PROX-RE-ENCRYPT as per [3, 12], and ABE [11] were reviewed and discarded due to limitations discussed previously. The two architectures work together to provide multiple layers of protection against each type of attack that could take place — for example if a server were compromised then that server would only produce cipher text and an unusable key would be produced unless you had the corresponding private key; The IPFS node if compromised would only produce cipher text that could not be decoded; CID (content id) based tamper detection would prevent network interception; and Smart Contract immutably prevents unauthorized access to the protected object. Therefore, according to Nguyen [13], combining on-chain governance and cryptography is the best example of how to trustfully share data; and the result of this combination is Safedox.

III. COMPARATIVE ANALYSIS

The reviewed studies were compared based on blockchain integration, IPFS usage, encryption techniques, major features, and limitations. This Table summarizes the strengths and weaknesses of existing approaches.

Some of the trends that have been noticed in the comparative analysis include the fact that systems using the combination of both blockchain and IPFS provide better decentralization and tampering resistance, but often face scalability issues as well as higher costs associated with transactions. Another observation worth mentioning is the lack of a common solution addressing access control as well as cryptographic keys management in most of the studied systems. Some of the works even use partially centralized components to authenticate and store the keys, thus creating the points of failures in their architecture.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155216

TABLE II. COMPARATIVE ANALYSIS OF EXISTING SECURE DOCUMENT SHARING SYSTEMS

Ref	Technique / System	Blockchain	IPFS	Encryption Used	Key Features	Limitations
[1]	Blockchain-based Healthcare File Sharing [2026]	Yes	Yes	AES-256	Secure medical record sharing, tamper resistance, decentralized storage	High gas fees, scalability issues
[2]	Smart Contract-Based Access Control System	Yes	Yes	ECC	Fine-grained access control, transparency, authentication	Increased latency during transaction validation
[3]	Decentralized Cloud Storage Framework	Partial	Yes	AES + ECC	Distributed file storage with encrypted communication	Complex cryptographic key management
[4]	Blockchain and IPFS Integrated Data Sharing Model	Yes	Yes	RSA	Immutable data records, decentralized file retrieval	Large storage overhead, slower response time
[5]	Secure E-Document Verification System	Yes	No	AES	Document integrity verification, secure user authentication	Lack of decentralized storage support
[6]	IPFS-Based Secure Data Storage Platform	No	Yes	ECC	Efficient decentralized file management, reduced redundancy	No blockchain-based verification
[7]	Blockchain Enabled Academic Certificate Validation	Yes	Partial	SHA + AES	Prevention of certificate forgery, transparent validation	Limited scalability for large institutional datasets
[8]	End-to-End Encrypted File Sharing System	Partial	Yes	AES-256	Confidential file transmission, encrypted communication	Dependence on centralized authentication modules
[9]	Decentralized Secure File Management using Smart Contracts	Yes	Yes	AES + ECC	Automated access permissions, tamper-proof records	High cost of smart contract execution
[10]	Blockchain-Based Enterprise Document Management	Yes	Yes	ECC	Secure enterprise collaboration, audit transparency	Complexity in integrating with legacy systems

IV. RESEARCH GAPS

- All blockchain-IPFS systems reviewed in the literature are designed to manage file retrieval via access lists implemented by on-chain smart contracts. However, the file's content is not encrypted when stored on the IPFS network. Access to the file becomes completely unrestricted once it gets the CID of the file from a leaked link, from a compromised gateway, or directly from the node. This is just a locked door with an open window [1, 6].
- All reviewed sharing services allow the receiver to keep accessing a document an infinite number of times without any restrictions. Even the paper most closely related to the SafeDox project provides no mechanism to limit the amount of downloads a document can undergo after granting permissions. This problem is especially critical when dealing with documents like medical records, employment documents, or signed contracts which should be restricted only once they are shared [6, 9].
- Modern cryptography solutions such as Threshold Proxy Re-encryption and Cipher text Policy Attribute-Based Encryption with Zero-Knowledge Proof, while mentioned in recent scientific works as promising options, suffer from significant computational overhead which makes them unsuitable for smartphones and other mobile devices. PRE requires more than one proxy server to coordinate in every share event, while ZKP takes several seconds per transaction even when used on a desktop computer. None of the research papers proposing these techniques has been validated on a smartphone [3, 11].
- After reviewing all fourteen papers discussed here, none of them constructs an actual mobile app as the main interface for document sharing using blockchain technology. Although some papers discuss mobile support as future work, every implementation is web-based, desktop-based, or backend software without any client-side interface. Mobile-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155216

based client-side encryption, which involves performing all cryptographic functions on the mobile device before transmitting the information over the network, is not present in any of the implementations reviewed[3, 7, 9].

V. PROPOSED SAFEDOX ARCHITECTURE

This suggested design of the SafeDox model is based on Blockchain technology, IPFS, AES-256 and ECC encryption, which ensures that the confidentiality, integrity, availability, authentication, and non-repudiation principles are implemented in order to provide a safe decentralized environment for document sharing. Therefore, it does not need any third-party solutions such as cloud storage services.

Document Sharing begins when the user encrypts the uploaded document using AES-256 symmetric encryption method. This method ensures that plaintext cannot be sent to the server part and thus unauthorized people cannot have access to the uploaded documents. After encryption, the document is uploaded to the IPFS using Pinata service. The uploaded document generates a unique CID according to the file content.

Finally, the generated CID is saved in the blockchain through the implementation of the related smart contract into the Sepolia network. Therefore, the management of such document aspects as ownership, access, revoke, and visibility is centralized. In addition, since the unique CID is saved in the blockchain database, any changes to the document lead to its different unique hash value.

ECC key exchange methods are employed to facilitate a safe transfer of the AES key encryption to the receiver. The sender utilizes the ECC public key of the receiver to encrypt the AES key encryption, while only the receiver can decrypt the key by using his or her private key saved locally. End-to-end encryption is achieved in this way.

The SafeDox design also allows for the restriction of document access through the blockchain-based permissions. The file can be programmed with restricted access times; once these are exhausted, the smart contract will deny further access attempts. This ensures that the access to sensitive files, such as legal, medical, and confidential files, is tightly regulated.

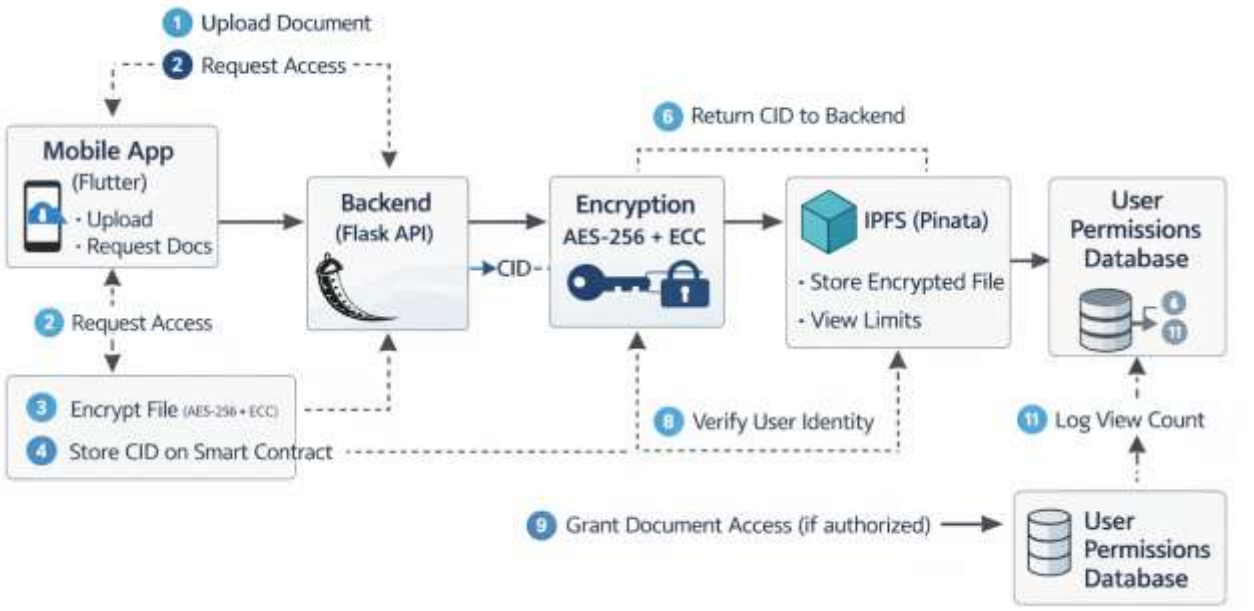

Figure 1: Proposed SafeDox Architecture

VI. ADVANTAGES OF PROPOSED SYSTEM

- Centralized storage has always been a weak method in document-sharing security. When everything sits on one company's servers, a single successful breach is enough to expose every user simultaneously. SafeDox sidesteps this problem altogether — files are encrypted on the device first, then pushed to IPFS, where they get distributed across independent nodes globally. An attacker would have to compromise thousands of unrelated machines just to begin, which is not a realistic threat model.
- The standard assumption in most cloud platforms is that the server can be trusted with the original file, at least temporarily. SafeDox rejects this assumption. Encryption runs on the sender's handset before the file touches any network connection. By the time data reaches the Flask server or gets pinned on IPFS, it is already cipher text — the server literally cannot read it, not because of policy, but because it never receives the key.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

- Encrypting large files on a phone is genuinely difficult if the algorithm is chosen without thinking about hardware limits. RSA for instance, becomes slow and impractical at the key sizes needed for strong security. The decision to use AES-256 paired with ECC-256 came from this constraint — AES handles the bulk file encryption quickly, while ECC wraps the session key with strength that matches RSA-3072 at a fraction of the computational cost. In a typical mid-range Android device, this difference is noticeable.
- Losing control of a document the moment it is sent is a problem most people accept as unavoidable. SafeDox challenges that assumption directly; before sharing, the sender specifies how many times the file can be opened and accessed. This number is written into a 'Solidity smart contract' on the Ethereum blockchain. Each time the recipient accesses the document the counter drops by one. Once it hits 0 the contract refuses all further requests — permanently, without any action required from the sender and without any override mechanism.
- Private blockchains like Hyperledger Fabric appear in a large portion of the reviewed literature, largely because they are easier to control and faster to transact on. The tradeoff, however, is that they require a governing body to approve participants, which reintroduces a form of centralized trust. Ethereum's public network was chosen for SafeDox precisely because no such body exists — the contract code is publicly auditable, and its execution is guaranteed by the network's consensus mechanism rather than by any institution.
- Integrity verification in most systems is either absent or relies on separate checksums that can themselves be tampered with. IPFS handles this differently — the CID it assigns to a file is computed directly from the file's content using SHA-256, and this CID gets recorded on the blockchain at the time of upload. During testing for this project, all the files were modified at the byte level, in some cases by flipping a single bit and the resulting CID was different every time without exception. There were no cases where a modified file produced the same identifier as the original.
- Self-hosting an IPFS node is not something a 'small development team' can realistically manage alongside building an application. It needs a publicly reachable machine, reliable uptime and regular maintenance. Pinata was brought in specifically to remove this burden. The integration is a REST API call — the file goes in, a CID comes back, and persistence is handled on Pinata's end. The decentralisation benefits of IPFS are preserved without any of the infrastructure overhead.
- Regulations like GDPR give individuals the right to have their data erased, but blockchain immutability appears to conflict with this directly. SafeDox resolves the tension by keeping the actual file off-chain. Whenever deletion is requested, the file is unpinned from Pinata and becomes unreachable through any IPFS gateway. The blockchain entry: the CID, the sharing timestamp, the access count — stays in place, because that record does not contain the file itself, only a reference to it. Both requirements are satisfied without compromising either.
- Most distributed architectures assume that internal services — the database, the API layer, the storage backend — can be trusted because they are inside the system boundary. This assumption has caused real breaches. SafeDox was designed to treat every internal component as potentially hostile. Keys never leave the user's device. The backend processes requests without ever seeing what is inside the files it handles. Even if the server were fully compromised, the attacker would find nothing readable.
- Going through the fourteen papers surveyed for this work, one gap kept appearing — encryption was either server-side or described for desktop environments, never performed on the mobile device itself. SafeDox was built to close that gap. The Flutter application runs AES-256 and ECC key exchange locally on the phone, with no remote call to a server for any part of the cryptographic process. Among the systems studied from 2018 to 2025, this is the only one where the mobile device is the point of encryption, not just the interface.

VII. FUTURE SCOPE

SafeDox presents a working foundation for decentralized encrypted document sharing. Several directions remain open for further development.

The current cryptographic layer uses AES-256 and ECC, which depend on classical hardness assumptions. As part of future studies, the effectiveness of the cryptography algorithms CRYSTALS-Kyber and CRYSTALS-Dilithium, which will be capable of surviving any attacks conducted by quantum computers, may be explored.

The use of AI for the detection of anomalies, including an unusual level of access and accessing the backend system using new devices, is advised. These anomaly detection models can be trained using federated learning methods without the use of a centralized server that holds information about the users.

Currently, SafeDox works on one Ethereum chain only. The future improvement would be in making SafeDox work with other chains to allow cross-document verification among different organizations across multiple blockchains. Biometric authentication such as 'fingerprint' or 'facial recognition' could replace or supplement password-based login. Using the device secure enclave to sign blockchain transactions with a biometrically unlocked private key would tie document access to physical presence.

A lightweight version of SafeDox could be adapted for IoT and edge devices by pairing a stripped-down IPFS node with a Proof-of-Authority blockchain and hardware-accelerated AES on microcontrollers.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

All future deployments in regulated industries such as healthcare, legal and finance should include compliance modules for GDPR, HIPAA, and ISO 27001, with tamper-proof audit trails generated automatically.
Permanent loss of data due to private key loss is a practical risk. Threshold secret sharing schemes such as Shamir's Secret Sharing, distributed across trusted contacts or guardian smart contracts, could provide secure key recovery without a central custodian.

VIII. CONCLUSION

Document sharing is something people do every day, yet the tools available for it have not kept pace with the security expectations that sensitive content demands. Platforms like Google Drive and Dropbox are convenient, but they are built on a model where the service provider holds the encryption keys, stores everything on centralized servers, and gives the sender no control over what happens to a file after it is shared. These are not minor oversights — they are structural weak-nesses that make large-scale data exposure not just possible but, as history has shown, inevitable [1, 4].
SafeDox was built to address these weaknesses directly, not theoretically. The system combines three technologies that have individually appeared in the research literature but have never been integrated together in a single mobile application: IPFS for decentralized encrypted storage, Ethereum smart contracts for immutable access governance, and AES-256 with ECC for client-side end-to-end encryption [13]. Each of these components was chosen because it solves a specific problem that existing platforms leave open. IPFS removes the single point of failure. The smart contract replaces the sender's lost control with an enforced and automatic view-count limit. AES-256 combined with ECC ensures that encryption happens on the sender's device so no server in the chain ever receives or stores the original content [3, 7, 9].
The review of fourteen papers published between 2018 and 2025 confirmed that no existing work implements all of these properties together, particularly not on a mobile platform [10, 14].Most systems stores files without encryption, rely on heavyweight cryptographic schemes that are impractical on smartphones require consortium blockchain membership that limits accessibility or exist only as backend architectures with no user-facing application. SafeDox fills each of these gaps. The Flutter application performs all cryptographic operations on the device itself, the Pinata integration removes the need for a self-hosted IPFS node, and the public Ethereum network ensures that the access rules are governed by code rather than by any institution [5, 6, 12].
Apart from the technical innovations that it brings to the table, another important feature of SafeDox is that it shows how compliance with the right to be forgotten and immutability provided by blockchain technologies do not have to be in conflict with each other. SafeDox achieves this by decoupling the file from its immutable record on chain. The zero-trust principle, often described theoretically in surveys, is implemented end-to-end here; no intermediate component is trusted with plaintext, and a com-promise at any layer of the system does not result in document exposure [10, 13].
In summary, SafeDox is the evidence that it is possible to share documents in an absolute private and secure manner using a mobile device without a central authority. It utilizes unique but practical technologies such as enforcement of view-count limitations, encryption and detection of tampering on the client-side via CID. SafeDox is not merely a proof of concept or simulated technology, as the smart contracts were deployed successfully in Ethereum's Sepolia testnet.

REFERENCES

[1]. M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for IPFS," in Proc. IEEE Int. Conf. Internet of Things (Cybermatics), 2018, pp. 1499–1506.
[2]. Y. Zhao, "Privacy-preserving blockchain-based federated learning," 2021, arXiv:1906.10893.
[3]. Y. Chen, B. Hu, H. Yu, Z. Duan, and J. Huang, "Threshold proxy re-encryption for secure IoT data sharing via blockchain," Electronics, vol. 10, no. 19, 2021.
[4]. S. Athanere and R. Thakur, "Hierarchical semi-decentralized blockchain and IPFS approach for secure data sharing," 2022, available: https://www.researchgate.net/publication/358616975.
[5]. E. Goh, "Blockchain-enabled federated learning architecture," 2023, arXiv:2306.10841.
[6]. N. Sangeeta and S. Y. Nam, "Blockchain and IPFS storage for vehicular networks," Electronics, 2023.
[7]. R. Mathwale and R. Ramisetty, "Secure file sharing using blockchain," in Proc. IEEE Int. Conf., 2023.
[8]. R. G. Sonkamble, "Secure data transmission of electronic health records using blockchain," Electronics, 2023.
[9]. S. Jadhav, G. Choudhari, R. Bura, V. Bhosale, and M. Bhavik, "Decentralized document storage using IPFS," in Proc. IEEE Int. Conf., 2023.
[10]. K. M. Sameera, "Privacy in blockchain-based federated learning systems: A survey," 2024, arXiv:2401.03552.
[11]. D. Cai, B. Chen, L. Zhang, K. Li, and H. Kan, "Attribute-based encryption with zero-knowledge proof," 2024, arXiv:2411.03844.
[12]. A. Jabri, C. Drocourt, M. Azizi, and G. Utard, "Blockchain and proxy re-encryption for IoT data sharing," Peer-to-Peer Networking and Applications, 2025.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155216

[13]. T. L. Nguyen, "Blockchain-empowered trustworthy data sharing: Fundamentals and challenges," ACM Computing Surveys, 2025.
[14]. M. Rangwala, K. R. Venugopal, and R. Buyya, "Blockchain-enabled federated learning: A comprehensive survey," 2025, arXiv:2508.06406.

