FileTrust: A Secure Decentralized System for Privacy-Preserving Data Storage and Sharing

Mohammadali Farahpoor
Electra SI SL.

July 6, 2025

Version 1.2

Abstract

FileTrust is a decentralized, secure file-sharing solution that leverages blockchain, InterPlanetary File System(IPFS), and advanced cryptographic technologies to provide privacy-focused, distributed data storage. Unlike traditional cloud storage, FileTrust ensures that no single entity has complete control over your data. Instead, files are split into encrypted pieces and distributed across multiple nodes, making unauthorized access nearly impossible. This approach enhances privacy, data availability, and system resilience. FileTrust tokens incentivize participants, such as Vault Keepers and Guardians, to provide storage and maintain security. This white paper explores the motivations, architecture, technologies, use cases, and economic model of FileTrust, showcasing how it differentiates itself from existing file-sharing and storage solutions by offering a more secure, private, and user-centric platform. By eliminating the risks associated with centralized control, FileTrust aims to establish a trustless, transparent, and highly secure environment for managing sensitive data, catering to industries like IoT, healthcare, and legal services.

Contents

List of Figures

1 Introduction

The FileTrust project is designed to solve the critical challenges in secure file storage and sharing. With a focus on privacy, resilience, and decentralization, FileTrust leverages advanced technologies, including blockchain, IPFS[1], and cryptographic methods, to create a unique and user-centric platform. This section introduces the project's overall aim, motivations, and the need for a system like FileTrust to address gaps in traditional and existing decentralized storage solutions.

1.1 Overview

The increasing reliance on digital storage has brought about significant concerns over data privacy, security, and control. Centralized cloud storage services dominate the market, but they come with risks related to data breaches, single points of failure, and unauthorized access. The need for a decentralized, privacy-focused storage system has never been greater. FileTrust aims to fill this gap by utilizing blockchain for transparency and security, IPFS for distributed data storage, and advanced cryptographic methods to ensure data privacy. By incentivizing participation through FileTrust tokens, FileTrust creates an ecosystem where all stakeholders benefit.

1.2 Motivation

Traditional cloud storage services, such as those offered by major tech companies, face numerous challenges, including vulnerability to cyberattacks, data privacy concerns, and centralized control over user data. Users often have no choice but to trust these entities with their sensitive information, which can lead to misuse, breaches, or service disruptions. The motivation behind FileTrust is to address these shortcomings by providing a decentralized solution that empowers users with full control over their data. The system ensures that no central authority can access or manipulate the data without user consent, thereby mitigating risks related to data privacy and availability.

The growing popularity of blockchain technology and decentralized storage protocols, such as Filecoin[12] and BitTorrent[9], demonstrates a clear demand for more secure and transparent alternatives. However, these existing solutions often lack the robust privacy features and user-centric control that FileTrust aims to provide. By integrating multi-party computation (MPC), verifiable secret sharing (VSS), and proxy re-encryption (PRE), FileTrust introduces an extra layer of security that ensures user data remains private and protected, even when stored across multiple nodes.

1.3 General Aim

The primary aim of FileTrust is to create a decentralized, secure, and transparent file-sharing and storage platform that prioritizes user privacy and control.

The system is designed to cater to individuals and organizations that require secure management of sensitive data, such as healthcare providers, legal professionals, and IoT networks. FileTrust employs cutting-edge technologies to split files into encrypted pieces, distribute them across a global network of nodes (Vault Keepers), and manage access using cryptographic keys and smart contracts.

In addition to providing a secure file storage solution, FileTrust also aims to incentivize participation through the use of FileTrust tokens. These tokens reward Vault Keepers and Guardians for their contributions to the network, ensuring that the system remains robust and resilient. By combining advanced security protocols, decentralization, and an innovative incentive model, FileTrust seeks to redefine the way sensitive data is stored and shared in the digital age.

1.4 Objectives

To achieve its vision, FileTrust has established several key objectives:

- Develop a decentralized platform that eliminates the need for centralized control over user data.
- Ensure data privacy through the use of cryptographic methods, such as MPC, VSS, and PRE.
- Create a resilient storage system that distributes file pieces across multiple nodes, reducing the risk of data loss or breaches.
- Incentivize participants through FileTrust tokens to maintain the network's security and reliability.
- Provide a user-friendly interface that allows individuals and organizations to easily store, share, and manage their data securely.

2 Background

Before discussing state-of-the-art literature, it is important to outline the fundamental cryptographic techniques and technologies pertinent to our proposal. These include Secret Sharing (SS), VSS, Information Dispersal Algorithm (IDA), MPC, PRE, and the IPFS. Each technique is crucial for ensuring security, privacy, and data management efficiency within IoT environments.

2.1 Secret Sharing (SS)

Secret sharing involves distributing a secret or key across multiple participants, where each receives a unique share. The secret can be reconstructed if a predefined number of shares are combined; however, individual shares alone provide no information. Typically, there is a dealer and n players involved. The dealer distributes shares such that any group of m or more participants can collectively

reconstruct the secret, while groups with fewer than m participants cannot. This approach, known as an (m, n)-threshold scheme, was developed by Blakley and Shamir [15, 2].

- **Secret Sharing (SS):** A secret S is divided into n shares, allowing any t out of n shares to reconstruct S. A polynomial $f(x)$ of degree $t-1$ is used where:
$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$$
with $a_0 = S$.
- **Share Generation:** Each participant P_i receives a share $s_i = f(x_i)$, where x_i is a unique non-zero value assigned to each participant.

Secret sharing techniques ensure high levels of integrity, reliability, and confidentiality, which is essential for securing sensitive information like encryption keys and financial data. Shamir's Secret Sharing (SSS) offers this confidentiality and reliability but relies heavily on the trustworthiness of the dealer. For cases where dealer trust is uncertain, VSS is introduced as a solution.

2.2 Verifiable Secret Sharing (VSS)

A verifiable secret sharing scheme enables participants to verify the integrity of their shares. VSS ensures that if a dealer is malicious, there is still a well-defined secret that participants can reconstruct. Introduced by Chor et al. [4], a common VSS protocol by Feldman is based on Shamir's scheme and includes homomorphic encryption [8]. VSS incorporates mechanisms allowing participants to validate share consistency, thus enhancing security.

- **Public Commitment:** The dealer publicly commits to the polynomial coefficients by publishing commitments $C_0, C_1, \ldots, C_{t-1}$, where $C_j = g^{a_j}$ for a generator g in a cyclic group.
- **Share Verification:** Each participant verifies their share s_i by checking:
$$g^{s_i} \stackrel{?}{=} \prod_{j=0}^{t-1} C_j^{x_i^j}$$
- **Reconstruction:** Any t participants can reconstruct S with Lagrange interpolation:
$$S = f(0) = \sum_{i \in I} s_i \lambda_i$$
where I is a subset of t participants, and λ_i are Lagrange coefficients.

FVSS proves efficient for environments requiring transaction efficiency and is particularly useful in distributed systems [6, 5]. One limitation is the high storage and computational cost for distributing secrets in large systems [3].

2.3 Information Dispersal Algorithm (IDA)

Rabin's IDA divides data across multiple locations, ensuring redundancy while protecting against unauthorized access. IDA divides a file into n pieces, allowing reconstruction from specific subsets of pieces.

- **Message Splitting:** A message M of size m is split into n pieces, each m/k in size, with k being the threshold.
- **Encoding:** M is represented by a matrix $\mathbf{M}$ and encoded using a random $n \times k$ encoding matrix $\mathbf{E}$.
- **Piece Generation:** Each i-th piece P_i is generated as:

$$P_i = \mathbf{E}_i \mathbf{M}$$

- **Reconstruction:** To reconstruct M, any k pieces are used with a $k \times k$ submatrix of $\mathbf{E}$.

IDA offers low storage and computational costs, making it suitable for large files [14, 7].

2.4 Multi-Party Computation (MPC)

MPC enables multiple parties to jointly compute a function without revealing their private inputs. In our project, FVSS is used within the MPC framework to generate the secret key for encrypting CID.

- **Secure Multiplication:** A secure multiplication protocol computes $x \times y$ without revealing the values.

$$[z] = [x] \times [y]$$

- **Secret Sharing of Inputs:** Each party secret shares their input using FVSS.

$$x = x_1 + x_2 + \cdots + x_n$$

FVSS facilitates efficient non-interactive verification, crucial for blockchain-based applications [13, 10].

2.5 Proxy Re-Encryption (PRE)

Proxy Re-Encryption (PRE) allows a proxy to convert ciphertexts encrypted under one key to another without revealing the plaintext. This feature is essential for secure data sharing in decentralized environments.

- **Re-Encryption Key Generation:** A re-encryption key $rk_{k1 \to k2}$ allows transformation between keys k_1 and k_2.

$$rk_{k1 \to k2} = f(k_1, k_2)$$

- **Ciphertext Transformation:** The proxy transforms C_{k1} into C_{k2} for the recipient.

This project employs Green and Ateniese's Identity-Based PRE scheme for efficient key management and secure data sharing [11].

2.6 InterPlanetary File System (IPFS)

IPFS is a peer-to-peer file system that connects devices using a unique identifier called CID for each data piece.

- **Content Identification (CID):** Each file in IPFS is hashed to create a unique CID.
$$\text{CID} = H(\text{file content})$$
- **File Chunking:** Files are divided into 256 KB chunks.
- **File Distribution and Retrieval:** Each chunk is stored in the network and retrieved using the CID.

3 FileTrust Overview

The FileTrust platform is a comprehensive solution for secure and decentralized file storage and sharing, leveraging blockchain, IPFS, and advanced cryptographic techniques to ensure data privacy, integrity, and availability. This section provides an in-depth overview of the FileTrust system, including its architecture, components, and the key features that distinguish it from other solutions.

3.1 System Architecture

The architecture of FileTrust is designed to ensure scalability, privacy, and resilience. At its core, the system relies on a combination of blockchain technology, IPFS, and cryptographic protocols. The blockchain is used to store metadata, encryption keys, and access control policies, while IPFS is responsible for storing the actual data fragments. The architecture also includes nodes that perform various functions, including Vault Keepers, Guardians, and Users.

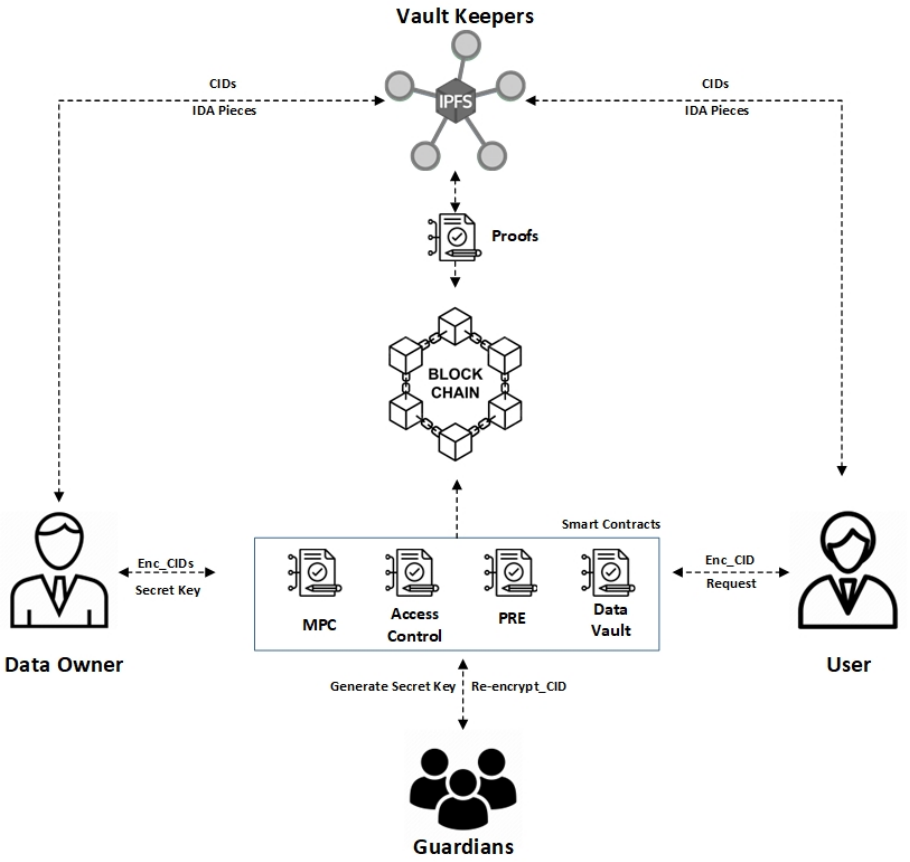

Figure 1: FileTrust work flow

3.2 Key Components

The main components of the FileTrust platform include:

- **Vault Keepers**: These nodes are responsible for storing the encrypted fragments of data. Vault Keepers receive incentives in the form of FileTrust tokens for providing storage and ensuring data availability.
- **Guardians**: Guardians are responsible for managing cryptographic operations, such as MPC and VSS, which ensure that data remains secure without exposing any private keys or sensitive information.
- **Users**: Users interact with the FileTrust platform to store, retrieve, and share data. They have full control over their data and can grant or revoke access permissions as needed.

- **FileTrust Tokens**: FileTrust tokens are used to incentivize participation in the network. Both Vault Keepers and Guardians receive tokens for their contributions, which helps maintain the security and reliability of the platform.

3.3 Data Flow and Encryption

FileTrust uses a combination of cryptographic techniques to ensure data privacy and security. When a user uploads a file, it is first fragmented using the IDA. Each fragment is then encrypted, and the encrypted fragments are distributed across multiple Vault Keepers. The metadata, including encryption keys and access control policies, is stored on the blockchain, ensuring that no single entity has complete control over the data.

Access to the stored data is managed using PRE. When a user requests access to a file, the Guardians perform PRE to re-encrypt the encryption keys for the requesting user without revealing the original keys. This ensures that data sharing remains secure and that access control is managed efficiently.

3.4 Incentive Mechanism

The incentive mechanism in FileTrust is designed to encourage active participation in the network. Vault Keepers are rewarded with FileTrust tokens for storing data and ensuring its availability, while Guardians are incentivized for performing cryptographic operations that maintain data privacy and security. This dual-incentive model ensures that all aspects of the platform are adequately maintained, leading to a robust and resilient network.

3.4.1 Proof of Storage (for Vault Keepers)

Use a **Proof of Storage** model where Vault Keepers prove they hold data fragments without revealing the data. This can be achieved using *homomorphic hashing* or *Merkle trees*.

- **Merkle Tree-Based Proof**:
  - Vault Keepers generate a Merkle Tree root M_{root} for each stored file.
  - To prove storage, they periodically reveal a random leaf hash and provide a path in the Merkle tree to verify the root.
  - **Verification**:
    - The requester generates a hash function $H(\text{data\_chunk}) = \text{hash\_leaf}$.
    - Given the path from hash_leaf to M_{root}, the validator checks that the hash matches M_{root}.
  - **Mathematical Incentive**: Vault Keepers passing p consecutive proof challenges receive a reward R_k, adjusted by reputation, $R_k = B_k \cdot (1 + w \cdot \text{Rep}_k)$, where B_k is the base reward, w is the reputation weight, and Rep_k is the reputation score.

3.4.2 Proof of Integrity (for Peer-to-Peer Verification)

The **Proof of Integrity** model ensures data authenticity during peer-to-peer transfers, preventing nodes from injecting malicious code into file pieces. This approach verifies each piece in real-time and connects integrity checks to node reputation.

- **Cryptographic Signature Verification**:
  - Each data piece D_i is signed by the data owner, producing a cryptographic signature σ_i.
  - Upon receiving a piece D_i during download, the requester verifies σ_i using the data owner's public key:

    $$\text{Verify}(\sigma_i, D_i, \text{PK}_{\text{DO}})$$

    where PK_{DO} is the public key of the data owner. The verification ensures D_i is untampered by confirming it matches the original signature.

- **Real-Time Hash Comparison**:
  - Nodes store and provide each data piece D_i with its hash $h_i = H(D_i)$, generated with a secure hash function H.
  - During download, the requester computes a new hash $h_i' = H(D_i)$ for the received piece and checks it against h_i:

    $$h_i' \stackrel{?}{=} h_i$$

    If h_i' does not match h_i, the piece is flagged as compromised, and the node's reputation score is adjusted.

- **Reputation and Penalty Adjustment**:
  - For each compromised piece, the node's reputation score Rep_x is penalized. Define δ as the penalty factor for failed integrity checks:

    $$\text{Rep}_x' = \text{Rep}_x - \delta$$

    where δ depends on the severity and frequency of integrity failures. Repeated integrity breaches lower Rep_x, reducing the node's likelihood of future rewards.
  - Nodes with consistently high integrity scores gain a reputation boost, calculated as:

    $$\text{Rep}_x' = \text{Rep}_x + \gamma \cdot \text{success}$$

    where γ is the positive weight for each successful integrity verification.

- **Incentive Structure for Verified Integrity**:

- Nodes that consistently provide verified data pieces receive an integrity reward R_i, scaled by reputation and the number of integrity checks passed:

$$R_i = B_i \cdot (1 + \beta \cdot \text{Rep}_x)$$

  where B_i is the base reward and β is the integrity incentive factor.
- Nodes failing integrity checks face potential slashing of rewards or stake, proportional to failed attempts.

3.4.3 Proof of Computation (for Guardians)

For Guardians responsible for cryptographic operations (MPC, VSS, PRE), a **Proof of Computation** can be used to validate their work, ensuring integrity without revealing sensitive data.

- **MPC Verification**:
  - Guardians jointly generate Secret_Key through secure multiparty computation.
  - Each Guardian G_i submits a partial computation proof for verification using homomorphic encryption or zero-knowledge proofs (ZKP) to confirm their input without revealing it.
  - **Zero-Knowledge Proof of Correctness**:
    - Guardians prove they followed the correct cryptographic process by providing ZKPs of the form $\text{ZK}(x) : \text{PK}\{x : \text{MPC}(x) = y\}$, where x is a private input and y the result.
    - The reward for each Guardian G_i is calculated as $R_g = B_g \cdot (1 + v \cdot \text{Rep}_g)$, where B_g is the base reward, v the reputation weight, and Rep_g the Guardian's reputation score.

3.4.4 Proof of Speed (for Vault Keepers and Guardians)

The **Proof of Speed** model assesses the efficiency of data transfer by validating upload and download speeds, ensuring that nodes can retrieve and serve file fragments within a performance threshold.

- **Upload Speed Verification**:
  - Nodes periodically upload sample data chunks of predetermined size D_{chunk} to the system.
  - The time taken to upload D_{chunk} is measured, and the upload speed U_{speed} is calculated as:

$$U_{\text{speed}} = \frac{D_{\text{chunk}}}{T_{\text{upload}}}$$

- If $U_{\text{speed}} \geq U_{\min}$, where $U_{\min}$ is the minimum acceptable speed, the node passes the upload proof.

- **Download Speed Verification**:
  - Nodes respond to download requests for file fragments, measuring the time T_{download} taken to deliver the requested chunk D_{chunk}.
  - The download speed D_{speed} is calculated as:

$$D_{\text{speed}} = \frac{D_{\text{chunk}}}{T_{\text{download}}}$$

  - Nodes passing a threshold $D_{\text{speed}} \geq D_{\min}$ qualify for performance rewards.

- **Incentive Structure**:
  - Nodes achieving consistent speed benchmarks receive rewards scaled by speed performance and reputation: $R_s = B_s \cdot (1 + \gamma \cdot \text{Rep}_x)$, where B_s is the base reward, and γ is a speed incentive weight.

3.4.5 Reputation Scoring and Update Mechanism

Each node's **Reputation Score** Rep_x (for Vault Keepers and Guardians) can dynamically reflect their performance. This score could follow a **weighted moving average** model to ensure stability and responsiveness to recent performance.

- **Reputation Calculation**:
  - Suppose $\text{Rep}_x^{(t)}$ is the reputation of node x at time t, updated as:

$$\text{Rep}_x^{(t+1)} = \alpha \cdot Q_x + (1 - \alpha) \cdot \text{Rep}_x^{(t)}$$

    where:
    - Q_x is the quality score of the node in the latest proof.
    - α is a weighting factor (e.g., 0.7) balancing recent performance and historical score.
  - **Quality Score** Q_x Components:
    - Availability A_x: proportion of time the node was online.
    - Reliability R_x: success rate in passing storage or computation proofs.
    - Latency L_x: average response time, normalized.

    Then, Q_x could be defined as:

$$Q_x = \lambda_1 \cdot A_x + \lambda_2 \cdot R_x + \lambda_3 \cdot \frac{1}{L_x}$$

    where λ_i are weights reflecting the importance of each factor.

3.4.6 Challenge-Response Audits for Vault Keepers

In a **challenge-response** protocol, Vault Keepers periodically respond to random file fragment challenges.

- **Mathematical Formulation**:
  - Let $P(x_i, f_t)$ represent the probability of Vault Keeper x_i responding correctly to a challenge f_t at time t.
  - Define $P(x_i)$ as the aggregate response rate over T trials:

$$P(x_i) = \frac{1}{T} \sum_{t=1}^{T} P(x_i, f_t)$$

  - If $P(x_i) > P_{\min}$, where $P_{\min}$ is the threshold probability, the Vault Keeper receives a reward. Failing the threshold leads to a penalty reducing Rep_x.

3.4.7 Token Incentive Model with Staking and Slashing

To enforce commitment, nodes can **stake** tokens, and their rewards or penalties can depend on the staked amount.

- **Staking Function**:
  - For node x with a reputation Rep_x and stake S_x, the reward R_x is adjusted by:

$$R_x = (1 + k \cdot \frac{S_x}{S_{\max}}) \cdot B_x$$

    where:
    - k is a coefficient scaling the stake impact.
    - $S_{\max}$ is the maximum possible stake, normalizing the reward.

- **Slashing**:
  - If x fails audits or reputation falls below a threshold, a portion δ of S_x can be slashed:

$$S'_x = S_x - \delta \cdot S_x$$

  - This slashing provides a penalty and reinforces the reliability of Vault Keepers and Guardians.

3.5 Security and Privacy Features

FileTrust incorporates several advanced security and privacy features to protect user data:

- **Multi-Party Computation (MPC)**: Ensures that sensitive operations, such as encryption key generation, are carried out securely without exposing any individual input.
- **Verifiable Secret Sharing (VSS)**: Distributes secrets among multiple nodes, ensuring that no single node can reconstruct the secret without reaching the threshold.
- **Proxy Re-Encryption (PRE)**: Allows for secure sharing of data without re-encrypting the entire dataset, providing efficient and dynamic access control.

4 Technology Stack

The FileTrust platform leverages several key technologies to achieve its objectives of secure, decentralized file sharing and storage. This section provides an overview of the primary technologies and frameworks used in the system.

4.1 Blockchain Technology

FileTrust utilizes blockchain technology to store metadata, manage access control policies, and ensure transparency in data transactions. The blockchain provides an immutable ledger that records all activities, enhancing the security and auditability of the system. Smart contracts are employed to automate processes such as access permission management, data sharing, and incentive distribution.

4.2 InterPlanetary File System (IPFS)

The IPFS is used for decentralized file storage. Files are fragmented using the IDA and stored across multiple IPFS nodes (Vault Keepers). IPFS ensures high availability and efficient retrieval of file fragments, while also reducing the reliance on centralized storage providers. The distributed nature of IPFS enhances data redundancy and resilience.

4.3 Cryptographic Techniques

To ensure data privacy and security, FileTrust incorporates several advanced cryptographic methods:

- **Information Dispersal Algorithm (IDA)**: Files are split into multiple fragments before being stored in IPFS. IDA ensures that even if some

nodes become unavailable, the original file can still be reconstructed from the available fragments.

- **Multi-Party Computation (MPC)**: MPC is used to generate and manage encryption keys securely without revealing them to any individual node. This enhances privacy in key management processes.
- **Verifiable Secret Sharing (VSS)**: VSS ensures that secrets, such as encryption keys, are shared among multiple nodes in a secure manner. No single node can reconstruct the secret without the collaboration of other nodes.
- **Proxy Re-Encryption (PRE)**: PRE allows for secure data sharing by enabling encrypted data to be re-encrypted without revealing the original content. This facilitates dynamic access control and efficient data sharing.

4.4 Smart Contracts

Smart contracts are used to automate various operations within the FileTrust platform. These include access management, FileTrust token distribution, and the consensus processes required for adding or revoking members from different groups (e.g., Vault Keepers and Guardians). Smart contracts provide transparency, immutability, and efficiency, ensuring that all processes are executed in a trustless environment.

4.5 FileTrust Tokens

FileTrust tokens are used to incentivize participation in the FileTrust network. Both Vault Keepers and Guardians receive tokens for their contributions, which helps maintain the network's security and reliability. The token-based incentive mechanism encourages active participation, thereby enhancing data availability and the robustness of the platform.

4.6 Ethereum Blockchain

The Ethereum blockchain is utilized for storing metadata, smart contracts, and token transactions. By leveraging Ethereum's established infrastructure, FileTrust benefits from a secure, well-supported blockchain network. The use of Ethereum also facilitates the integration of FileTrust tokens and smart contracts into the platform, providing a seamless experience for participants.

5 System Architecture

The architecture of the FileTrust platform is designed to provide scalability, security, privacy, and resilience in data storage and sharing. The system consists of multiple components that work together to ensure that data is securely managed and available at all times. This section provides an overview of the

architectural components and how they interact to achieve the system's objectives.

5.1 Overview of Components

The FileTrust architecture comprises several key components, each of which plays a specific role in the secure storage and sharing of data. The main components include:

- **Data Owner (DO)**: The entity that owns the data and decides how it should be shared and stored. The DO has full control over the data and can grant or revoke access permissions.
- **Vault Keepers**: These nodes are responsible for storing encrypted data fragments. They participate in the decentralized storage network and are incentivized using FileTrust tokens to maintain data availability and security.
- **Guardians**: Guardians perform cryptographic operations, such as MPC and VSS, to ensure that data encryption and decryption processes are secure. They also manage PRE operations to facilitate secure data sharing.
- **Ethereum Blockchain**: The blockchain stores metadata, access control policies, smart contracts, and token transactions. It acts as an immutable ledger that provides transparency and security to the platform.
- **IPFS Nodes**: These nodes store the fragmented data pieces generated by the IDA. The distributed nature of IPFS ensures high availability and resilience, allowing the data to be reconstructed even if some nodes are unavailable.
- **Users**: Users interact with the FileTrust platform to request access to data. Depending on their permissions, they can decrypt and reconstruct data shared by the Data Owner.

5.2 Data Flow

The data flow in the FileTrust system can be summarized in the following steps:

1. **Data Fragmentation and Encryption**: When a Data Owner uploads a file, it is first fragmented using the IDA. Each fragment is then encrypted to ensure confidentiality before being distributed to multiple Vault Keepers.
2. **Storage in IPFS**: The encrypted fragments are stored across different IPFS nodes (Vault Keepers), ensuring data redundancy and resilience. The corresponding CIDs for each fragment are stored in the blockchain.

3. **Key Generation and Sharing**: Guardians perform MPC and VSS to generate and manage the encryption keys required for data access. These keys are securely shared among Guardians, ensuring that no single entity has complete control over them.

4. **Access Control and Sharing**: When a user requests access to a file, the Data Owner can grant access by using PRE. The Guardians perform PRE to re-encrypt the encryption keys for the requesting user, allowing them to decrypt the data fragments without revealing the original keys.

5. **Data Retrieval and Reconstruction**: The user retrieves the encrypted data fragments from the IPFS nodes using the CIDs. Once the fragments are retrieved, the user can decrypt and reconstruct the original file using the provided keys.

5.3 Smart Contracts

Smart contracts are an integral part of the FileTrust architecture. They are used to manage access permissions, token distribution, and consensus processes. By automating these operations, smart contracts provide transparency and efficiency while reducing the need for manual intervention. Key functions of smart contracts in FileTrust include:

- **Access Management**: Smart contracts enforce access control policies defined by the Data Owner, ensuring that only authorized users can access specific data.

- **Incentive Distribution**: FileTrust tokens are distributed to Vault Keepers and Guardians based on their contributions to the network. This incentivizes active participation and helps maintain the platform's security and availability.

- **Membership Management**: The consensus process for adding or revoking members, such as Vault Keepers or Guardians, is managed through smart contracts, ensuring that changes are transparent and agreed upon by the network.

5.4 Security and Privacy Considerations

The architecture of FileTrust is designed with a strong focus on security and privacy. Several mechanisms are in place to protect data throughout its lifecycle:

- **Decentralized Storage**: By storing data fragments across multiple IPFS nodes, FileTrust eliminates the risks associated with a single point of failure. Even if some nodes are compromised or become unavailable, the data can still be reconstructed.

- **Encryption and Key Management**: All data fragments are encrypted before storage, and encryption keys are managed using MPC and VSS to prevent unauthorized access.
- **Proxy Re-Encryption (PRE)**: PRE allows data to be shared securely without revealing the original encryption keys. This ensures that only authorized users can access the data, even when it is being shared.
- **Blockchain Transparency**: The use of the Ethereum blockchain provides an immutable ledger that records all activities, including access requests, permissions, and token transactions. This ensures accountability and auditability, enhancing the overall security of the platform.

5.5 Scalability and Resilience

The FileTrust architecture is designed to be scalable and resilient, capable of handling large volumes of data and a growing number of participants. Key features that contribute to scalability and resilience include:

- **InterPlanetary File System (IPFS)**: The use of IPFS allows FileTrust to scale by adding more nodes to the network, thereby increasing storage capacity and data availability.
- **Dynamic Node Participation**: Vault Keepers and Guardians can join or leave the network dynamically, and the system can adapt to these changes without affecting data availability or security.
- **Token-Based Incentives**: The incentive mechanism ensures that participants are motivated to contribute resources to the network, helping maintain a high level of service as the platform grows.

6 File Storage and Retrieval Process

The FileTrust platform provides a secure and efficient mechanism for storing and retrieving files. This section outlines the step-by-step workflow for file dispersal, storage, retrieval, and encryption.

6.1 Step-by-Step Workflow

The process of storing and retrieving files in the FileTrust system can be divided into several key steps:

6.1.1 Step 1: File Dispersal and Encryption

When a DO uploads a file, it is first split into multiple fragments using the IDA. Each fragment is then encrypted individually to ensure confidentiality. This means that even if an unauthorized entity gains access to a fragment, it cannot be reconstructed or understood without the encryption key.

6.1.2 Step 2: Storage in IPFS

The encrypted fragments are stored across multiple IPFS nodes, which are managed by Vault Keepers. Each fragment is assigned a unique CID, which serves as a reference to locate the fragment within the distributed storage network. The CIDs are stored on the Ethereum blockchain to ensure transparency and immutability.

6.1.3 Step 3: Key Generation and Management

Guardians generate and manage the encryption keys required to access the encrypted file fragments. Using MPC and VSS, Guardians ensure that no single entity has complete control over the keys, providing a layer of security that prevents unauthorized access to the entire key.

6.1.4 Step 4: Requesting Access

When a user requests access to a file, the Data Owner grants the necessary permissions by leveraging PRE. Guardians perform the PRE operation to re-encrypt the encryption keys for the specific requesting user. This ensures that only the authorized user can decrypt the file fragments without revealing the original keys.

6.1.5 Step 5: Retrieval from IPFS

The user retrieves the encrypted fragments from the IPFS nodes using the CIDs stored on the blockchain. Since the data is distributed across multiple nodes, the decentralized approach ensures high availability and resilience, even if some nodes are temporarily unavailable.

6.1.6 Step 6: Decryption and File Reconstruction

Once the user has retrieved all the encrypted fragments, they use the re-encrypted keys provided by the Guardians to decrypt the fragments. After decryption, the original file is reconstructed using the IDA. The successful retrieval and reconstruction of the file confirm that the system maintains data integrity and privacy.

6.2 Encryption and Security

- **Information Dispersal Algorithm (IDA)**: IDA is used to split files into fragments, ensuring that the original file cannot be reconstructed without a sufficient number of fragments. This adds a layer of redundancy and security.
- **Encryption of Fragments**: Each fragment is encrypted before being stored in IPFS. This ensures that even if an individual fragment is ac-

cessed by an unauthorized entity, it cannot be understood without the appropriate encryption key.

- **Proxy Re-Encryption (PRE)**: PRE allows encrypted data to be re-encrypted for a different user without exposing the original plaintext or encryption keys. This is crucial for maintaining data privacy while allowing authorized users to access data.
- **Key Management with MPC and VSS**: The use of MPC and VSS ensures that no single entity has complete control over the encryption keys. This enhances the security of the key management process and prevents unauthorized access.

6.3 Advantages Storage and Retrieval Process

- **Decentralization**: By using IPFS, the FileTrust platform avoids the risks associated with centralized storage solutions, such as a single point of failure or potential data breaches.
- **Enhanced Security**: The combination of IDA, encryption, and advanced cryptographic techniques ensures that data remains secure throughout the storage and retrieval process.
- **High Availability**: The distributed nature of IPFS ensures that data is always available, even if some nodes are temporarily offline.
- **User Control**: Data Owners maintain full control over who can access their data, and PRE ensures that data sharing is both secure and efficient.

7 Incentive Mechanism

The FileTrust platform uses a token-based incentive mechanism to encourage active participation by Vault Keepers and Guardians, ensuring the network's security, reliability, and scalability. This section describes the role of FileTrust tokens, the reward structure, and the smart contract logic that governs the incentives.

7.1 FileTrust Tokens

FileTrust tokens are the primary currency used to incentivize participants in the FileTrust network. The tokens are used for various purposes, such as rewarding Vault Keepers for providing storage and Guardians for managing cryptographic operations. FileTrust tokens serve as a means to:

- Reward Vault Keepers for storing data fragments and ensuring data availability.

- Reward Guardians for performing cryptographic tasks, such as MPC, VSS, and PRE.
- Allow users to pay for storage, data retrieval, and data sharing services on the FileTrust platform.

7.2 Rewards and Participation for Vault Keepers and Guardians

- **Vault Keepers**: Vault Keepers are rewarded with FileTrust tokens based on their contributions to the network, which include storing encrypted file fragments and ensuring data availability. The reward mechanism takes into consideration the amount of storage provided, the duration of storage, and the availability of data.
- **Guardians**: Guardians are rewarded for performing cryptographic operations that are crucial to maintaining the security of the network. These operations include generating encryption keys through MPC, managing VSS, and executing PRE for data sharing. Rewards are distributed based on the number of cryptographic tasks performed and the level of participation in securing the platform.

The incentive mechanism is designed to ensure that all participants are motivated to contribute to the network's success, which in turn enhances data security, privacy, and availability.

7.3 Smart Contract Logic for Incentives

Smart contracts play a crucial role in automating the distribution of incentives within the FileTrust platform. These contracts are deployed on the Ethereum blockchain and are responsible for ensuring transparency, security, and efficiency in the reward distribution process. Key aspects of the smart contract logic include:

- **Automatic Reward Distribution**: Smart contracts automatically distribute FileTrust tokens to Vault Keepers and Guardians based on their contributions to the network. The reward calculation is done in a transparent manner, with all transactions recorded on the blockchain.
- **Penalty Mechanism**: To ensure reliability, a penalty mechanism is in place for participants who fail to meet their obligations. For instance, Vault Keepers who do not maintain data availability or Guardians who fail to perform cryptographic tasks may face penalties, which are also enforced through smart contracts.
- **Consensus-Based Membership**: The addition or removal of Vault Keepers and Guardians is managed through a consensus process governed by smart contracts. This ensures that only trustworthy participants are allowed to join the network, maintaining the integrity of the platform.

The use of smart contracts ensures that all incentives are distributed fairly and transparently, with no need for intermediaries. This trustless mechanism aligns the interests of all participants and contributes to the overall security and resilience of the FileTrust platform.

8 Security Considerations

The FileTrust platform is built with a strong emphasis on security to ensure data privacy, resilience, and protection against unauthorized access. This section discusses the key security considerations that underpin the design of the FileTrust system.

8.1 Data Privacy

Data privacy is one of the core principles of the FileTrust platform. To achieve robust data privacy, the system employs several advanced cryptographic techniques:

- **Encryption of Data Fragments**: All file fragments are encrypted before being stored in IPFS. This ensures that even if a malicious actor gains access to individual fragments, they cannot reconstruct or understand the data without the encryption keys.
- **Proxy Re-Encryption (PRE)**: PRE allows encrypted data to be securely shared with authorized users without revealing the original encryption keys. This ensures that data privacy is maintained even during the sharing process, allowing only the intended recipient to decrypt the data.
- **Access Control via Smart Contracts**: Access control policies are enforced through smart contracts deployed on the blockchain. These contracts ensure that only authorized users can access specific data fragments, adding an additional layer of privacy.

8.2 Resilience and Redundancy

The decentralized architecture of FileTrust ensures that the platform is resilient and capable of handling node failures without compromising data availability:

- **Information Dispersal Algorithm (IDA)**: The use of IDA allows files to be split into multiple fragments, which are then distributed across different IPFS nodes. Even if some nodes become unavailable, the original file can still be reconstructed from the available fragments. This redundancy ensures that data is not lost due to node failures or other disruptions.
- **Distributed Storage in IPFS**: By storing data across multiple IPFS nodes, FileTrust eliminates the risks associated with a single point of failure. The distributed nature of IPFS ensures high data availability and resilience, even in the face of network disruptions or hardware failures.

- **Dynamic Node Participation**: Vault Keepers and Guardians can join or leave the network dynamically, and the system adapts to these changes without affecting data availability or security. This flexibility contributes to the overall resilience of the platform.

8.3 Multi-Party Computation for Secret_Key

The use of MPC for managing the Secret_Key is a critical security feature of the FileTrust platform. MPC ensures that no single entity has complete control over the encryption key, enhancing the security of the key management process:

- **Collaborative Key Generation**: Guardians collaboratively generate the Secret_Key using MPC, ensuring that the key is never fully revealed to any single participant. This approach mitigates the risk of key compromise and ensures that the encryption process is secure.
- **Verifiable Secret Sharing (VSS)**: VSS is used in conjunction with MPC to distribute shares of the Secret_Key among multiple Guardians. A threshold number of shares is required to reconstruct the key, which means that no individual Guardian can misuse the key without the collaboration of others. This adds an extra layer of security to the key management process.
- **Secure Key Usage**: The Secret_Key is used to encrypt the IDs of file fragments before storing them on the blockchain. This ensures that even if the blockchain is publicly accessible, the actual locations of the data fragments remain protected and confidential.

The combination of MPC and VSS ensures that the Secret_Key remains secure throughout its lifecycle, preventing unauthorized access and ensuring that the encryption and decryption processes are performed in a secure and privacy-preserving manner.

9 Economic Model

The FileTrust platform utilizes a comprehensive tokenomics model to incentivize participants and ensure the economic sustainability of the network. This section describes the tokenomics of FileTrust tokens, their distribution, and their utility within the platform.

9.1 Tokenomics

FileTrust tokens are the native cryptocurrency used within the FileTrust ecosystem. They serve as a medium of exchange and a means to incentivize participants, including Vault Keepers, Guardians, and other stakeholders. The key aspects of FileTrust tokenomics include:

- **Supply**: The total supply of FileTrust tokens is fixed to ensure scarcity and create long-term value. A predefined number of tokens are minted during the initial token generation event (TGE).
- **Utility**: FileTrust tokens are used for various purposes, including paying for data storage, rewarding participants, and staking to gain voting rights within the platform's governance.
- **Burn Mechanism**: A portion of the transaction fees collected from storage and retrieval activities is burned, reducing the overall token supply over time and creating deflationary pressure to increase the value of the tokens.

9.2 Token Distribution and Utility

The distribution of FileTrust tokens is designed to ensure that all stakeholders in the FileTrust ecosystem are incentivized fairly. The key components of the token distribution are as follows:

9.2.1 Initial Distribution

- **Guardians, Vault Keepers & Ecosystem (66%)**: The majority of tokens are allocated to incentivize network participants such as Vault Keepers and Guardians who provide decentralized storage, perform secure cryptographic operations, and contribute to the ecosystem's growth and resilience.
- **Development Contributors (15%)**: Tokens reserved for developers actively building and maintaining the FileTrust platform, rewarding long-term commitment and innovation.
- **Public Sale (ICO/IDO) (5%)**: Allocated for public investors during initial coin offerings to fund the early stages of the project and promote wider adoption.
- **Airdrop & Bounty Campaign (5%)**: Distributed to community members and early supporters through airdrops, marketing bounties, and other incentivized participation methods.
- **ElectraSI Foundation (5%)**: Reserved for the ElectraSI Foundation to support long-term research, development, and ecosystem growth initiatives.
- **Future Airdrops (2%)**: Set aside for future community growth and promotional campaigns to boost adoption.
- **Advisors (2%)**: Allocated to strategic advisors contributing to the project's direction, governance, and network growth.

9.2.2 Utility of FileTrust Tokens

FileTrust tokens play a critical role in the functioning of the FileTrust platform. Their utility includes:

- **Storage and Retrieval Fees**: Users pay FileTrust tokens to store and retrieve data on the platform. The fees collected are distributed among the Vault Keepers and Guardians as rewards for their contributions.
- **Incentives for Participation**: Vault Keepers and Guardians receive FileTrust tokens as incentives for storing data, managing encryption keys, and ensuring data security and availability. The incentives are distributed based on the level of participation and the value of services provided.
- **Staking and Governance**: Token holders can stake their FileTrust tokens to participate in the governance of the platform. Stakers gain voting rights and can propose or vote on changes to the platform, such as updates to the incentive mechanism or modifications to the system architecture.
- **Marketplace Transactions**: FileTrust tokens can be used within the FileTrust marketplace for additional services, such as enhanced data privacy options, increased storage capacity, and access to premium features.

The economic model of FileTrust is designed to create a sustainable and self-reinforcing ecosystem, where participants are rewarded for their contributions, and the value of FileTrust tokens is linked to the growth and success of the platform.

10 Governance Model

The governance model of FileTrust is designed to ensure that decisions are made transparently and collaboratively, involving all key stakeholders. This section discusses the consensus-based membership mechanism, as well as the roles of Guardians and Users in the governance of the platform.

10.1 Consensus-Based Membership

FileTrust utilizes a consensus-based membership process for adding or revoking participants such as Vault Keepers and Guardians. This approach ensures that the network remains secure and that only trustworthy participants are allowed to join. The consensus mechanism includes:

- **Membership Proposal**: Potential members, such as Vault Keepers or Guardians, must submit a proposal to join the network. This proposal may include identity verification and other credentials that demonstrate the applicant's reliability.

- **Voting Process**: Existing members, including Guardians and token holders, can vote on the proposal. A certain threshold of votes is required for the proposal to be approved, ensuring that new members are accepted based on community consensus.
- **Revocation of Membership**: If a member is found to be acting maliciously or failing to meet their obligations, their membership can be revoked through a similar voting process. This ensures that the network remains secure and that all participants are held accountable.

10.2 Role of Guardians and Users in the System

- **Guardians**: Guardians play a crucial role in maintaining the security and integrity of the FileTrust platform. Their responsibilities include managing cryptographic operations, such as MPC, VSS, and PRE. Guardians also participate in the governance process by voting on membership proposals, system upgrades, and changes to the platform's policies. Their role is essential in ensuring that data remains secure and that the platform operates in a decentralized manner.
- **Users**: Users of the FileTrust platform include Data Owners and individuals or organizations that request access to stored data. Users have the ability to participate in the governance process by staking FileTrust tokens, which grants them voting rights. This allows users to have a say in decisions that affect the platform, such as changes to the incentive mechanism or updates to the system architecture. By involving users in the governance process, FileTrust ensures that the platform remains user-centric and responsive to the needs of its community.

The governance model of FileTrust is designed to be transparent, decentralized, and inclusive, allowing all stakeholders to participate in decision-making processes. This collaborative approach ensures that the platform remains secure, resilient, and aligned with the interests of its users and participants.

11 Advantages over Competitors

FileTrust offers several key advantages over existing decentralized storage and file-sharing solutions, such as Filecoin, BitTorrent, and other similar systems. This section provides a detailed comparison of FileTrust's unique features and the benefits it offers over its competitors.

11.1 Existing Solutions

Filecoin and BitTorrent are two prominent decentralized storage solutions that address some of the limitations of centralized storage. Filecoin utilizes a blockchain-based incentive mechanism to encourage users to provide storage space, while

BitTorrent focuses on peer-to-peer file sharing with its BitTorrent Token (BTT) as an incentive.

Filecoin relies on a proof mechanism called Proof-of-Storage, which involves two main components: Proof-of-Replication (PoRep) and Proof-of-Spacetime (PoSt). PoRep ensures that the data has been replicated into a unique physical storage, while PoSt verifies that the storage provider continues to dedicate storage to that data over time. While effective in maintaining data integrity and incentivizing storage, Filecoin's primary focus is on cost-effective storage and availability rather than end-to-end privacy. Users must take additional steps to encrypt their data before storing it on the network, which can be cumbersome and relies on user diligence for maintaining privacy.

BitTorrent, on the other hand, leverages the efficiency of peer-to-peer (P2P) technology to distribute large files across a network of participants. The introduction of BitTorrent Token (BTT) was intended to incentivize users to continue seeding files and contributing bandwidth. However, BitTorrent lacks robust encryption for data privacy, and file availability depends heavily on user participation. This makes BitTorrent more suitable for public or non-sensitive data rather than confidential information. Additionally, there is no mechanism for fine-grained access control, meaning anyone with access to a file can potentially share it without restrictions.

11.2 Addressing the Gaps

FileTrust differentiates itself from existing solutions by introducing advanced privacy-preserving techniques, such as MPC, VSS, and PRE. Unlike Filecoin and BitTorrent, FileTrust ensures that data is encrypted, fragmented, and stored across nodes in such a way that ensures confidentiality, integrity, and privacy throughout the entire process.

One of the main limitations of existing decentralized storage solutions is their reliance on user-managed encryption for privacy. FileTrust addresses this gap by incorporating MPC and VSS, which together ensure that data encryption and decryption processes are carried out securely and collaboratively, without exposing private keys or sensitive information to any individual party. This approach makes data management seamless and highly secure, allowing users to store confidential information without concerns about unauthorized access.

FileTrust also uses PRE as a solution to the challenges related to access control and data sharing. PRE allows data owners to grant and revoke access permissions without re-encrypting the entire dataset, making it more efficient for managing dynamic access control. This feature is particularly important for industries such as healthcare and legal services, where access to sensitive information needs to be controlled in real time while maintaining compliance with strict privacy regulations.

Moreover, FileTrust incentivizes participation through the use of FileTrust tokens, ensuring that both Vault Keepers and Guardians are rewarded for their roles in storing, securing, and managing the data. Unlike Filecoin, where incentives are focused primarily on storage provision, FileTrust rewards nodes not

only for storage but also for their active participation in security protocols, such as MPC and VSS. This multi-faceted incentive mechanism enhances the overall robustness and security of the network.

In contrast to BitTorrent, where data availability is highly dependent on user participation, FileTrust's design ensures redundancy and resilience by leveraging IDA and VSS. This means that even if some nodes become unavailable or fail, the system can still reconstruct the file from the available pieces, providing higher reliability and reducing the risk of data loss. Additionally, the integration of MPC and PRE guarantees that data sharing remains secure, with fine-grained access control that is efficient and scalable for real-world applications.

By combining these advanced cryptographic and distributed storage techniques, FileTrust addresses the inherent gaps in existing decentralized storage systems, providing a secure, privacy-focused, and user-friendly solution for managing sensitive data. The unique combination of incentives, security protocols, and privacy guarantees sets FileTrust apart as a leading solution for decentralized file storage and sharing.

11.3 Comparison with Filecoin

Filecoin is a decentralized storage network that incentivizes users to provide storage space using a blockchain-based proof mechanism. While Filecoin offers a cost-effective storage solution, it has some limitations compared to FileTrust:

- **Privacy and Security**: Filecoin relies on users to encrypt their data before storing it, which can be cumbersome and relies on user diligence. In contrast, FileTrust uses built-in encryption and advanced cryptographic techniques, such as MPC and VSS, to ensure end-to-end privacy and security without burdening users with complex encryption processes.
- **Proxy Re-Encryption (PRE)**: FileTrust uses PRE to facilitate secure data sharing without revealing the original encryption keys. This feature provides dynamic access control, allowing Data Owners to grant and revoke access efficiently. Filecoin lacks a similar mechanism for dynamic, fine-grained access control.
- **User-Centric Governance**: FileTrust's governance model allows users and stakeholders to participate in decision-making processes through staking FileTrust tokens, giving them a say in platform updates and policies. Filecoin, on the other hand, has a more centralized decision-making process driven primarily by the core development team.

11.4 Comparison with BitTorrent

BitTorrent is a popular peer-to-peer (P2P) file-sharing protocol that uses the BitTorrent Token (BTT) as an incentive for users to share files. However, FileTrust offers several advantages over BitTorrent:

- **Data Privacy**: BitTorrent is primarily designed for public file sharing and lacks robust encryption mechanisms to ensure data privacy. FileTrust, on the other hand, is designed for secure, private data storage and sharing, making it suitable for sensitive information that requires strong privacy protections.
- **Resilience and Redundancy**: FileTrust uses the IDA to split files into multiple fragments and store them across different IPFS nodes. This ensures that even if some nodes become unavailable, the data can still be reconstructed. BitTorrent's availability relies heavily on user participation, and if seeders leave, the availability of the file is compromised.
- **Access Control**: BitTorrent lacks any form of access control, meaning that once a file is shared, anyone can access it. FileTrust uses smart contracts and PRE to enforce access control policies, ensuring that only authorized users can access specific data fragments.

11.5 Advantages over Other Systems

In addition to Filecoin and BitTorrent, FileTrust also offers advantages over other decentralized storage and sharing solutions:

- **Integrated Incentive Mechanism**: FileTrust's economic model is designed to incentivize all participants, including Vault Keepers and Guardians, for their contributions. FileTrust tokens are used for storage, retrieval, and staking, creating a self-sustaining ecosystem. Many other systems lack a comprehensive incentive mechanism that rewards both storage providers and security maintainers.
- **MPC for Key Management**: FileTrust uses MPC for generating and managing encryption keys, ensuring that no single participant has full control over the keys. This enhances the security of key management, which is often a weak point in other decentralized systems.
- **Decentralized Governance**: FileTrust's consensus-based membership and governance model ensure that all stakeholders, including users, Vault Keepers, and Guardians, have a say in decision-making processes. This decentralized governance model contrasts with other systems that rely on more centralized control, leading to a more transparent and community-driven platform.
- **Suitability for Sensitive Data**: FileTrust is specifically designed to handle sensitive data, such as healthcare records, legal documents, and confidential enterprise information. Its combination of encryption, PRE, and decentralized storage makes it a suitable solution for industries that require stringent data privacy and security standards.

The unique combination of advanced cryptographic techniques, a robust incentive mechanism, and decentralized governance gives FileTrust a competitive

edge over existing decentralized storage and file-sharing solutions. By addressing the limitations of other systems, FileTrust provides a secure, user-centric, and privacy-focused platform for managing sensitive data.

12 Conclusion

12.1 Vision and Future of FileTrust

FileTrust is on a mission to revolutionize the way data is stored and shared, placing a strong emphasis on privacy, security, and decentralization. By leveraging cutting-edge technologies, such as blockchain, IPFS, and advanced cryptographic techniques, FileTrust aims to provide individuals and organizations with full control over their sensitive data while ensuring resilience and availability. The long-term vision is to establish FileTrust as a leading solution for secure data management, catering to various industries such as healthcare, legal services, and IoT, where data privacy and integrity are of utmost importance.

As FileTrust continues to evolve, the focus will be on expanding the platform's capabilities, enhancing the user experience, and fostering a vibrant community of participants, including Vault Keepers, Guardians, and users. Future plans include expanding the token ecosystem, establishing new partnerships, and continuously improving the platform's security features. FileTrust is committed to building a decentralized, user-centric ecosystem that empowers individuals and businesses to manage their data securely and efficiently.

12.2 Key Takeaways

- **Privacy and Security**: FileTrust provides end-to-end data privacy and security using encryption, MPC, VSS, and PRE.
- **Decentralized Storage**: By utilizing IPFS and the IDA, FileTrust ensures data resilience, redundancy, and availability without relying on centralized entities.
- **Incentive Mechanism**: FileTrust tokens are used to incentivize Vault Keepers and Guardians, ensuring active participation and the continued security of the platform.
- **User-Centric Governance**: The governance model allows stakeholders, including users, Vault Keepers, and Guardians, to participate in decision-making processes through staking and voting, ensuring transparency and community-driven development.
- **Real-World Use Cases**: FileTrust is well-suited for industries that require secure and privacy-focused data management, including healthcare, legal services, IoT, and enterprise file sharing.

FileTrust's unique combination of decentralized storage, advanced cryptographic security, and a robust incentive model positions it as a groundbreaking solution in the field of secure data management. As the platform continues to grow, FileTrust remains committed to its core values of privacy, security, and user empowerment.

13 References

References

[1] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[2] George Robert Blakley. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*, pages 313–313. IEEE Computer Society, 1979.

[3] Anirudh Chandramouli, Ashish Choudhury, and Arpita Patra. A survey on perfectly secure verifiable secret-sharing. *ACM Computing Surveys (CSUR)*, 54(11s):1–36, 2022.

[4] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 383–395. IEEE, 1985.

[5] Sourav Das, Zhuolun Xiang, and Ling Ren. Asynchronous data dissemination and its applications. In *2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2705–2721, 2021.

[6] Sourav Das, Zhuolun Xiang, Alin Tomescu, Alexander Spiegelman, Benny Pinkas, and Ling Ren. Verifiable secret sharing simplified. *Cryptology ePrint Archive*, 2023.

[7] Mohammadali Farahpoor. *Distributed Information Dispersal Approach in Cloud Storage Systems for Enhancing Availability and Reliability.* PhD thesis, Universiti Teknologi Malaysia, 2014.

[8] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 427–438. IEEE, 1987.

[9] BITTORRENT FOUNDATION. Bittorrent (btt) white paper. *White paper*, 2019.

[10] Rosario Gennaro, Michael O Rabin, and Tal Rabin. Simplified vss and fast-track multiparty computations with applications to threshold cryptography. In *Seventeenth annual ACM symposium on Principles of distributed computing*, pages 101–111, 1998.

[11] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007. Proceedings 5*, pages 288–306. Springer, 2007.

[12] Protocol Labs. Filecoin: A decentralized storage network. *White paper*, 2017.

[13] Yehida Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005–1009. IGI global, 2005.

[14] Michael O Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM)*, 36(2):335–348, 1989.

[15] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.