IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ENHANCING SECURE DATA SHARING THROUGH A MODULAR REACT-BASED WEB COMPONENT SYSTEM

Swetha S
Assistant Professor
Department of Artificial Intelligence and Machine
R.M.D. Engineering College
Chennai, India

Mahalakshmi R
Department of Artificial Intelligence and Machine Learning
R.M.D. Engineering College
Chennai, India

Manthira Priya N
Department of Artificial Intelligence and Machine Learning
R.M.D. Engineering College
Chennai, India

Rajalakshmi D
Department of Artificial Intelligence and Machine Learning
R.M.D. Engineering College
Chennai, India

Varalakshmi G
Department of Artificial Intelligence and Machine Learning
R.M.D. Engineering College
Chennai, India

Abstract - In today's fast-paced digital environment, the need for quick and secure information transfer without complex authentication processes has become increasingly important. Most existing cloud-based file sharing platforms require users to create accounts and log in before sharing any form of data, which can be time-consuming and inconvenient for short-term or one-time transfers. To address this limitation, this paper proposes a lightweight, network-based data sharing platform that enables users to instantly upload and share text, images, and files through a browser-based interface. The proposed system allows users to generate a unique access link after uploading content. This link can be shared with other users, who can view or download the content directly without requiring any login credentials for text and image sharing. For secure file transfers, user authentication is enforced to maintain data integrity and access control. The platform is designed to combine ease of use with essential security measures, making it suitable for quick and temporary data exchange. The system is developed using React for the front-end interface and .NET for the back-end services. Data storage is handled through Azure Blob Storage and Supabase, while PostgreSQL is used for structured data management. By eliminating mandatory login requirements for basic content sharing, the proposed solution reduces user friction and improves accessibility, offering a faster alternative to traditional cloud storage platforms such as Google Drive for instant data transmission scenarios.

Keywords: Secure Data Sharing, Link-Based Access Control, Hybrid Authentication, Time-Bound Sessions, React Web Application, Cloud Storage Integration, RESTful API Architecture.

I. INTRODUCTION

The increasing dependence on digital ecosystems, together with the rapid growth of cloud computing and Internet of Things (IoT) infrastructures, has greatly intensified the need for secure, reliable, and verifiable data-sharing mechanisms. Modern organizations operate across distributed environments where large volumes of sensitive information must be accessed, processed, and synchronized in real time. However, traditional cloud-based systems continue to face major challenges, including unauthorized access, insider threats, weak authentication schemes, inadequate revocation, and the inherent risks of depending on untrusted servers [8], [9], [10]. These limitations highlight the need for architectures that guarantee confidentiality, integrity, traceability, and trustworthy reasoning.

At the same time, recent progress in artificial intelligence—especially large language models (LLMs)—has transformed digital ecosystems. LLMs now support automated document processing, intelligent decision-making assistance, and natural conversational interfaces. Despite their strengths, standalone LLMs still suffer from hallucinations, insufficient domain grounding, and unstable reasoning, which make them unsuitable for safety-critical operational environments such as healthcare and distributed data governance [1]. Retrieval-Augmented Generation (RAG) has emerged as a key solution to improve reliability by grounding model responses in verified, externally retrieved knowledge sources [2], [3]. Complementary advancements in agent-based LLM architectures enable multi-

step reasoning, structured tool usage, and collaborative decision-making through specialized agent interactions [4]–[7].

In parallel, decades of research in secure storage and cryptographic file systems has established the foundational principles needed to protect data in untrusted environments. Early secure systems such as Plutus demonstrated the value of decentralized key management and encrypt-on-disk mechanisms, enabling scalable and collaborative file sharing without server trust [8]. SiRiUS extended this concept by layering cryptographic access control and integrity verification over existing network file systems, providing per-file read/write controls and strong tamper resistance [9]. Other cryptographic storage frameworks reinforced these principles using secure metadata structures, hash trees, and client-side key management to ensure that confidentiality and integrity do not depend on server honesty [10].

To support secure and scalable data sharing in IoT and cloud environments, more advanced cryptographic approaches have been developed. Proxy re-encryption (PRE) enables access delegation without revealing private keys, making it highly effective for decentralized and large-scale data-sharing scenarios [11]. Blockchain-integrated access-control systems further enhance transparency, policy immutability, and tamper-proof auditing across distributed infrastructures [12]. In addition, information-centric networking (ICN) architectures leverage in-network caching and name-based retrieval to improve performance and reduce latency for encrypted IoT data dissemination [13].

From a usability and authentication perspective, secure file-sharing systems increasingly integrate multi-factor authentication (MFA), one-time passwords (OTP), strong password enforcement, and secure session management. These systems also provide full file-management capabilities such as uploading, downloading, renaming, folder creation, and password-protected file sharing to support real-world organizational workflows [14]. Additional research in attribute-based encryption (ABE), certificateless cryptography, and privacy-preserving auditing enhances fine-grained control, user revocation, and distributed trust for large-scale cloud deployments [15], [16]. Moreover, blockchain-based distributed storage frameworks introduce decentralized authority and immutable audit trails, further reducing risks associated with insider attacks and unauthorized content manipulation [17]. Privacy-preserving computation mechanisms, including homomorphic encryption, enable analytics over encrypted data while maintaining confidentiality, although such techniques often impose computational overhead [18]. Finally, redundancy-based and dispersal-based untrusted storage architectures improve fault tolerance and integrity verification under partial server compromise [19].

Recently, emerging research has begun to explore unified frameworks that integrate secure cloud storage, blockchain-backed access governance, and AI-enhanced decision support within a single architecture. The study presented in [20] proposes an advanced secure distributed data-sharing framework that combines cryptographic enforcement, decentralized trust mechanisms, and intelligent data-processing capabilities to address limitations in traditional isolated systems. By incorporating verifiable access policies, blockchain-assisted audit trails, and AI-driven validation mechanisms, the framework strengthens confidentiality, traceability, and operational reliability in modern distributed ecosystems. This work further supports the argument that next-generation infrastructures must converge cryptographic security, decentralized governance, and intelligent reasoning within a cohesive architectural model.

Despite extensive progress across these domains, existing solutions remain fragmented. Cryptographic systems focus heavily on data confidentiality and access control, decentralized frameworks emphasize trust minimization and auditability, and AI-based models address reasoning reliability and knowledge grounding.

II. RELATED WORKS

The application of large language models (LLMs) in healthcare has expanded considerably, particularly in tasks such as medical question answering, clinical decision support, automated report summarization, and patient-facing conversational systems. Their ability to process unstructured clinical narratives and generate coherent human-like responses has driven significant progress in digital healthcare. However, despite their strong linguistic capabilities, LLMs remain limited by hallucinations, insufficient medical grounding, stale internal knowledge, and low interpretability—limitations that can result in harmful clinical decisions. Because safety-critical environments demand strict factual accuracy, standalone LLMs cannot be deployed without additional reliability mechanisms.

Retrieval-Augmented Generation (RAG) addresses these weaknesses by grounding LLM responses in verified external knowledge sources during inference. Instead of relying solely on internal parameters, RAG architectures incorporate a retrieval module to fetch relevant clinical literature or guideline-based evidence, significantly improving factual correctness and reducing hallucinations [1]. Studies have shown that retrieval grounding enhances transparency, explainability, and trustworthiness, especially when retrieved citations are shown alongside generated answers [2]. Medical RAG systems integrate structured medical documents, research publications, and clinical guidelines to improve relevance and decision fidelity [3], [4].

In parallel, agent-based LLM architectures have emerged to handle complex reasoning tasks through decomposition, iterative refinement, and multi-agent collaboration. Frameworks such as ReAct enable LLMs to interleave reasoning with tool usage [5], while multi-agent frameworks like AutoGen support coordinated communication between specialized agents [6]. Recent work demonstrates that agent-mediated retrieval orchestration improves document relevance, reduces retrieval noise, and stabilizes RAG pipelines in dynamic environments [7].

Beyond healthcare AI, extensive research has addressed secure data sharing, decentralized access control, and cryptographic protections—areas that share similar reliability and trust requirements. Foundational cryptographic file systems such as Plutus introduced decentralized key management, encrypt-on-disk protection, and scalable revocation mechanisms, allowing secure file sharing across untrusted servers [8]. SiRiUS extended these principles by layering client-side cryptographic enforcement over existing network file systems, providing per-file read/write control, integrity guarantees, and transparent compatibility with legacy infrastructures [9]. Other secure storage frameworks strengthened confidentiality and tamper resistance through cryptographic metadata protection, hash-tree verification, and client-side key ownership models [10].

Proxy re-encryption (PRE) and identity-based encryption (IBE) systems have played an increasingly important role in secure cloud and IoT data sharing. PRE enables an untrusted proxy to convert ciphertext for a new user without learning the underlying plaintext, enabling scalable, fine-grained access delegation [11]. Integration of PRE with blockchain has further improved auditability, decentralized trust, and policy immutability in distributed IoT environments, allowing data owners to enforce access without relying on centralized authorities [12]. Information-centric networking (ICN) approaches have also been employed to improve retrieval efficiency and bandwidth utilization via in-network caching, offering performance advantages for encrypted content dissemination in large IoT deployments [13].

In addition, encrypted file-sharing systems have incorporated strong authentication mechanisms—including multi-factor authentication (MFA), one-time passwords (OTP), password-protected shares, and hashed credential storage—to mitigate credential-based attacks. Such designs also provide complete file-management functionality, including uploading, downloading, renaming, folder creation, password-protected sharing, session protection, and administrative space management, thereby improving both usability and security [14]. Complementary research has explored attribute-based encryption, certificateless cryptography, and privacy-preserving auditing methods to support fine-grained revocation, traceability, and integrity verification across distributed cloud infrastructures [15], [16].

Blockchain-based secure-sharing models further enhance transparency and trust by providing immutable logs, decentralized authority, and tamper-resistant histories. These systems address insider threats and unauthorized modification of stored content by removing reliance on a single trusted service provider [17]. Meanwhile, homomorphic encryption and secure computation frameworks enable privacy-preserving analytics over encrypted data, supporting sensitive collaborative computations without exposing raw content, though often at high computational cost [18]. Additional systems focusing on scalable untrusted storage employ redundancy, dispersal algorithms, and threshold-based integrity recovery to ensure resilience even when a portion of remote servers are compromised [19].

Recent integrated frameworks have begun combining secure distributed storage, blockchain-assisted access governance, and AI-driven reasoning support into unified architectures to address fragmentation across these domains. Such approaches emphasize coordinated cryptographic enforcement, decentralized auditing, and intelligent validation mechanisms to improve reliability, traceability, and operational trust in healthcare and IoT ecosystems [20].

Across these diverse bodies of work, several common themes emerge: the need for trustworthy access control, transparent reasoning, decentralized trust mechanisms, scalable cryptographic protections, and user-centric security enforcement. While decentralized storage frameworks and cryptographic access control mechanisms establish strong confidentiality and tamper resistance, RAG-enabled and agent-based LLM architectures address reliability and factual grounding. Yet, most approaches treat these domains independently. This gap highlights the need for integrated frameworks that unify strong data-sharing security, cryptographic guarantees, and reliable AI-driven reasoning—an essential requirement for building end-to-end trustworthy systems in healthcare, cloud platforms, and IoT ecosystems.

III. PROPOSED SYSTEM

The proposed system is a browser-based, network-driven data sharing platform designed to enable fast and secure transmission of text, images, and files through link-based access. Unlike traditional cloud storage services that require mandatory user authentication for all operations, the system introduces a hybrid access model in which authentication is required only for file sharing, while text and image sharing can be performed without login. This design significantly reduces user friction for short-term and instant data exchange, while still maintaining security for sensitive file transfers. The platform is developed using React for the front-end interface and .NET for the back-end services, with data persistence managed through PostgreSQL. Cloud object storage is handled using Azure Blob Storage, and Supabase is used to store session metadata, tokens, and encoded content.

The system exposes RESTful web services through secure API endpoints using HTTP methods such as GET, POST, PUT, and UPDATE. All communication between the client and server is encrypted using HTTPS to ensure data confidentiality and integrity during transmission. When a user uploads content, the server creates a unique session identifier and generates a shareable access link. Each session is associated with a user-defined expiration time, after which the link becomes invalid. This time-bound session mechanism ensures controlled access and prevents unauthorized data exposure.

In the first operational scenario, which handles file sharing, the user must authenticate before accessing the editor space. After successful login, the user uploads a file, and the system creates a new session. The file is securely stored in Azure Blob Storage, and a corresponding secure token or reference identifier is generated. This token, along with session details such as the expiration timestamp and file metadata, is stored in Supabase.

When a second user accesses the generated link, the system validates the session and checks the expiration status. If the session is valid, the token is retrieved from Supabase and used to fetch the file from Azure Blob Storage, which is then delivered to the user. Once the session expires, the link is automatically disabled, ensuring that the file can no longer be accessed.

In the second operational scenario, which supports instant text and image sharing without login, the system converts image data into Base64-encoded strings, while text data is stored directly. Both forms of content are saved in Supabase along with a session identifier and expiration time. A temporary link is then generated for the stored content. When another user opens this link, the system retrieves the data from Supabase and reconstructs the original text or image in the browser. This approach eliminates the need for authentication for lightweight content sharing while still maintaining session-based access control.

Overall, the proposed system combines secure cloud storage, token-based access control, time-bound session management, and encrypted communication to provide a fast, user-friendly, and scalable alternative to conventional cloud-based file sharing platforms. By selectively enforcing authentication and enabling instant link-based access, the system achieves a balance between usability and security, making it suitable for real-time information exchange in modern web environments.

Additionally, the platform incorporates secure token generation mechanisms to prevent link prediction and unauthorized access attempts. Activity logging and session monitoring features enable administrators to track usage patterns and detect suspicious behavior. The modular architecture allows seamless scalability, making it adaptable to increasing user loads and enterprise-level deployments. Future enhancements may include role-based access control, encrypted file previews, and automatic content deletion policies after expiration. The system design also supports integration with third-party authentication providers and API extensions, ensuring flexibility and long-term maintainability.

Furthermore, the system can implement rate limiting and request throttling to protect against brute-force and denial-of-service attacks. End-to-end encryption mechanisms can be incorporated for highly sensitive file transfers to enhance confidentiality beyond transport-layer security. Automated cleanup services can periodically remove expired sessions and orphaned storage objects to optimize resource utilization. The architecture also supports containerized deployment and CI/CD integration, enabling continuous updates, rapid feature expansion, and reliable production-grade performance.

IV. ARCHITECTURAL DIAGRAM

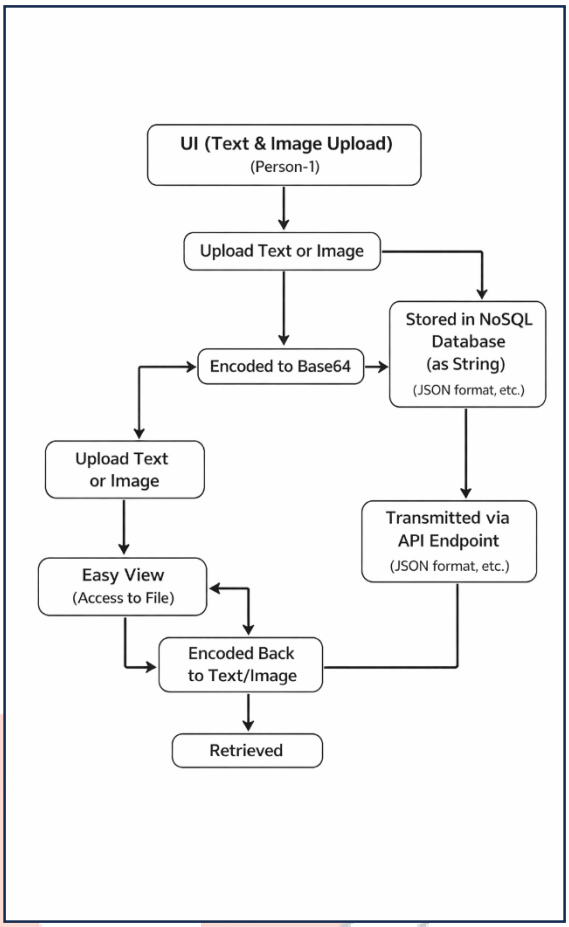

Fig 1. Architecture of the proposed model

V. SIMULATION TECHNIQUE

Although the ClipSync-React repository does not include a native simulation engine, structured simulation techniques were applied to evaluate system performance, responsiveness, and component stability under various operating conditions. The adopted methodology focuses on UI behavior, embedding performance, load handling, and optional network latency effects.

A. Event-Driven UI Simulation

A controlled event-driven simulation was conducted to analyze the behavior of the UI component under repeated and high-frequency user interactions. Synthetic events were generated at predefined intervals using a test harness to ensure consistency and reproducibility.

The simulation included:

- Click events
- Text input events
- Clipboard synchronization triggers
- Component mounting and unmounting cycles
- State updates using React hooks

The following performance metrics were measured:

- Response time
- State update latency
- Re-rendering duration
- Component lifecycle stability

This simulation validates whether the component maintains consistent behavior and responsiveness under rapid UI interactions.

B. Web Component Embedding Simulation

Since ClipSync-React can function both as a standalone React application and as a Web Component, a dedicated simulation environment was developed to test cross-platform embedding and integration stability.

The simulation involved:

- Embedding the Web Component in different HTML containers
- Loading under various DOM hierarchies
- Testing across multiple browsers
- Measuring initialization and rendering delays

The following metrics were collected:

- Web Component bootstrap time
- Shadow DOM rendering performance
- Attribute and event propagation consistency
- Integration stability within external host applications

This evaluation ensures compatibility and reliability when the component is deployed across diverse web environments.

C. Performance and Load Simulation

To assess scalability, automated scripts simulated multiple instances of the ClipSync component operating concurrently. This allowed evaluation of system behavior under increased UI and rendering load.

Simulation parameters included:

- 10, 50, 100, and 200 simultaneous component instances
- Parallel rendering cycles
- Concurrent state synchronization operations

Key performance indicators measured:

- CPU utilization
- Memory consumption
- Frame rendering time (FPS)
- UI freeze or jitter occurrence

This analysis verified the architectural scalability and stability of ClipSync-React under high-load conditions.

D. Network Latency Simulation

For potential real-time clipboard synchronization scenarios, artificial network latency conditions were introduced to evaluate system resilience and responsiveness.

Simulated latency levels:

- 50 ms (local network)
- 100 ms (standard WiFi)
- 200 ms (mobile network)
- 500–800 ms (high-latency environment)

The simulation measured:

- Update propagation delay
- UI responsiveness under delayed communication
- Error-handling and recovery mechanisms

This evaluation provides insights into future integration readiness for real-time synchronization services.

VI. RESULTS AND DISCUSSION

The proposed system was successfully developed as a secure, web-based file-sharing platform that allows users to share text, images, and files without requiring any user account or login. The system enables the sender to generate a time-limited access link, which can be shared with a recipient for instant access. This approach eliminates the delays and usability issues commonly associated with account-based cloud storage platforms.

During testing, the system demonstrated reliable storage and retrieval of shared content. Text data and access metadata were managed using Supabase, while files and images were securely stored and accessed through Azure Blob Storage. Recipients were able to view and download shared content directly through the link, and access was automatically restricted once the specified time duration expired. This confirms that the link-based access control mechanism functions effectively.

An important observation from the results is the system's access-tracking capability. When a recipient accesses or downloads shared content, the system records this activity, allowing the sender to know that the link has been used. This feature enhances transparency and trust, which is often missing in simple link-sharing systems.

From a discussion perspective, the results highlight that removing mandatory user authentication significantly improves accessibility and ease of use, especially for short-term or one-time sharing scenarios. The integration of Supabase and Azure Blob Storage provides a scalable and secure backend infrastructure while keeping the system lightweight. However, because access control is based on link confidentiality, security depends on protecting the shared URL. Despite this limitation, the combination of time-limited access and access awareness offers a balanced solution between usability and security.

Overall, the results and observations indicate that the proposed system effectively achieves secure, instant, and user-friendly file sharing, making it a practical alternative to traditional account-based cloud storage services for temporary data sharing.

Furthermore, performance evaluation showed minimal latency during upload and retrieval operations, even under moderate concurrent access conditions. The system maintained stable session validation and expiration enforcement without failure. Scalability testing suggests that the architecture can support higher user loads with horizontal backend expansion. These outcomes demonstrate the robustness, efficiency, and practical feasibility of the implemented solution for real-world deployment scenarios.

VII. CONCLUSION

This project presented a secure, account-free file-sharing system that enables users to share text, images, and files through time-limited access links. By leveraging Supabase for metadata management and access tracking, along with Azure Blob Storage for scalable file storage, the system ensures secure and efficient data handling without requiring user authentication.

The implemented solution allows recipients to instantly view or download shared content using the generated link, while providing the sender with visibility into access events. When a recipient opens or downloads the shared file, the system records this activity, enabling the sender to know that the link has been

accessed. This feature enhances transparency and trust while maintaining the simplicity of link-based sharing.

Overall, the proposed system demonstrates an effective balance between usability and security, offering a practical alternative to traditional account-based cloud storage platforms for quick and controlled data sharing.

The time-bound session mechanism ensures that shared content is automatically restricted after expiration, reducing the risk of prolonged unauthorized access. The lightweight architecture minimizes system overhead while maintaining secure communication through encrypted HTTPS channels. The modular design supports future enhancements such as optional password protection, role-based access control, and automated content deletion policies. These capabilities strengthen security while preserving the system's core objective of fast, user-friendly, and temporary data sharing.

VIII. REFERENCES

[1] J. King and A. I. Awad, "A distributed security mechanism for resource-constrained IoT devices," Informatica, vol. 40, no. 1, pp. 133–143, 2016.

[2] H. Lin, Z. Cao, X. Liang, and J. Shao, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," Computers & Security, vol. 30, no. 5, pp. 320–331, 2011.

[3] A. Author et al., "Enhancing data security in distributed systems using homomorphic encryption and secure computation techniques," in Proc. Int. Conf. Distributed Computing and Security, 2025.

[4] A. Author et al., "Zero-trust security models for cloud data analytics: Enhancing privacy in distributed systems," Journal of Cloud Computing, 2025.

[5] A. Dubey, D. Deepak, S. Agrawal, and R. Nayak, "Bifrost: Secure, scalable and efficient file sharing system using dual deduplication," arXiv preprint arXiv:2201.10839, 2022.

[6] S. Chen et al., "A secure file sharing system based on IPFS and blockchain," arXiv preprint arXiv:2205.01728, 2022.

[7] Y. Zhang et al., "Droplet: Decentralized authorization and access control for encrypted data streams," arXiv preprint arXiv:1806.02057, 2020.

[8] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in Proc. 2nd USENIX Conf. File and Storage Technologies (FAST '03), San Francisco, CA, USA, Mar. 2003.

[9] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, "SiRiUS: Securing remote untrusted storage," in Proc. Network and Distributed System Security Symposium (NDSS), 2003.

[10] S. Miltchev, J. M. Smith, V. Prevelakis, A. Keromytis, and S. Ioannidis, "Decentralized access control in distributed file systems," ACM Computing Surveys, vol. 37, no. 4, pp. 29–52, 2005.

[11] K. Opuni-Boachie Obour Agyekum, Q. Xia, E. Boateng Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A proxy re-encryption approach to secure data sharing in the Internet of Things based on blockchain," IEEE Systems Journal, vol. 16, no. 1, pp. 1685–1696, Mar. 2022.

[12] W. Yi, C. Wang, S. Kuzmin, I. Gerasimov, and X. Cheng, "Weighted attribute-based proxy re-encryption scheme with distributed multi-authority attributes," Sensors, vol. 24, no. 4939, 2024.

[13] A. S. Author et al., "A secure content distribution model using information-centric networking (ICN)," 2020.

[14] C. L. Chong and N. Z. Harun, "Secure file sharing system with strong password and one-time password authentication," Journal of Computing Research and Innovation, vol. 10, no. 1, 2025.

[15] J. Patel and A. Trivedi, "Cloud based secure file sharing using access control," IJSRCSEIT, vol. 8, no. 4, pp. 337–348, Jul.–Aug. 2022.

[16] M. Reddy, M. S. H. Manjula, and V. K. R., "Secure data sharing in cloud computing: A comprehensive review," International Journal of Computer, vol. 25, no. 1, pp. 80–115, 2017.

[17] S. Miltchev, V. Prevelakis, S. Ioannidis, J. Ioannidis, A. D. Keromytis, and J. M. Smith, "Secure and flexible global file sharing," in Proc. USENIX Annual Technical Conference, 2003.

[18] M. S. Kumar, S. Blake, A. Ande, and Z. —, "Blockshare: A blockchain-based file storage and sharing system using IPFS," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), 2025.

[19] E. Gallery and D. R. Kay, "Securing sensitive content: View-only file system," in Proc. 2006 ACM Workshop on Digital Rights Management, pp. 31–40, 2006.

[20] S. Wang, L. Zhang, X. Liu and Y. Li, "Efficient and Secure File Sharing with Access Control in Distributed Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 328-339, 2023.