International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org

Chrono Capsule: A Decentralized Human Knowledge Time Capsule System

Aryasudhan R[1, *], **Ayyanar Vijay T**[2], **Hemlathadhevi A**[3]

[1 2 3] Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.

*** Corresponding author.**
E-mail: *aryasudhanravikumar@gmail.com*

Article History:
Received Date: 25 January 2026
Revised Date: 22 February 2026
Accepted Date: 05 March 2026
Publication Date: 17 March 2026

Keywords:

Blockchain; Decentralized Storage; Time Capsule; IPFS; AES Encryption; Smart Contracts; Digital Preservation; Filecoin; Cryptography

ABSTRACT

Preserving human knowledge across generations is essential for keeping our culture, science, and shared understanding alive. Traditional storage systems that rely on centralized servers face serious risks—they can be corrupted, tampered with, censored, or lost due to single points of failure. In this paper, we introduce Chrono Capsule, a decentralized system designed to protect the permanence, integrity, and accessibility of digital artifacts. It uses blockchain technology together with decentralized storage platforms like IPFS and Filecoin to record metadata, verify authenticity, and release stored knowledge at specific times. To keep data private while still allowing verification, the system employs cryptographic hashing and AES encryption without relying on any central authority. Chrono Capsule's modular design includes layers for authentication, encryption, storage, and retrieval, allowing users to safely contribute, preserve, and access information over long periods. Built on principles of trust, transparency, and community governance, this approach also supports Sustainable Development Goals (SDGs) 9, 11, and 12, helping build a more secure, resilient, and sustainable digital future for generations to come.

1. Introduction

Throughout history, humanity has sought ways to preserve its collective memory from cave paintings and manuscripts to libraries, and now digital archives. In today's digital era, information is abundant but fragile. Platforms can become obsolete, file formats may change, and centralized storage services can fail, be manipulated, or simply disappear [1]. As digital knowledge continues to grow, concerns about the long-term preservation of human knowledge, scientific evidence, and personal records are becoming more pressing. Traditional time capsules and centralized digital archives provide storage, but they offer no guarantees of authenticity, accessibility, or protection from tampering [2] [3].

Emerging decentralized technologies, such as blockchain and distributed storage networks, offer promising solutions. Blockchain ensures immutability and trust through a transparent, verifiable ledger. Distributed storage networks like IPFS and Filecoin provide redundancy and censorship-resistant storage. The Chrono Capsule project leverages these technologies to create a decentralized, community-driven ecosystem where individuals can securely store encrypted knowledge artifacts that are time-locked for future retrieval [4] [5].

Every submission to Chrono Capsule is recorded on a blockchain along with metadata including the author's identity, a timestamp, and a content hash which serves as proof of authenticity. The content itself is

encrypted and distributed across multiple storage nodes, eliminating any single point of failure. Smart contracts enforce rules that ensure submissions remain time-locked until their designated release, providing a secure and verifiable way to preserve knowledge for the future [6].

Objective:

The objectives of this work are:

- To design a decentralized, tamper-proof digital knowledge preservation system accessible to individuals and institutions worldwide.
- To integrate blockchain technology with IPFS/Filecoin decentralized storage for immutable, censorship-resistant data management.
- To implement AES-256 encryption and SHA-256 hashing for end-to-end data confidentiality and integrity verification.
- To enforce time-locked access control through Ethereum smart contracts with automated release mechanisms.
- To present system functionality through an intuitive web dashboard for uploading, tracking, and retrieving time-locked capsules.

2. Literature Review

2.1 Evolution of Blockchain-Based Archival Systems

Preserving digital records has evolved significantly over the past decade. The earliest systematic approaches relied on centralized repositories and static archiving methods, which were vulnerable to corruption, institutional failure, and deliberate tampering. These methods were fundamentally ill-equipped to guarantee long-term authenticity of stored artifacts [7] [8].

With the maturation of blockchain technology, more robust decentralized approaches began to be applied to digital preservation. Immutable ledgers, cryptographic hashing, and distributed consensus mechanisms offered capabilities that centralized systems could not match. However, these early blockchain archival systems faced scalability constraints, governance challenges, and integration difficulties with existing institutional workflows [9][10].

2.2 Review of Existing Systems

Existing blockchain-based archival systems such as ARCHAIN and ARCHANGEL established foundational work in tamper-proof digital record management. ARCHAIN demonstrated blockchain's ability to build trust in digital record transfers, while ARCHANGEL extended this to multimedia archives using temporal content hashing [11][12]. National archives studies further explored institutional adoption barriers including lack of technical expertise, scalability issues, and regulatory compliance concerns. More recent work on blockchain architecture for cultural heritage preservation combined IPFS with smart contracts to ensure data authenticity and decentralized control, though limitations in handling large datasets and incentive mechanisms for data contributors were noted [13][14].

2.3 Identified Research Gaps

Existing blockchain archival systems primarily focus on single-domain preservation without integrating encryption, time-locking, and decentralized storage in a unified, accessible platform. Many solutions lack user-friendly interfaces, rely on institutional infrastructure, and do not support community-driven contributions from individuals [15]. There is also a gap in combining client-side AES encryption with IPFS-based redundant storage and smart-contract time-lock enforcement in one cohesive system. This highlights the need for an affordable, integrated, and intuitive knowledge preservation system like Chrono Capsule.

3. Proposed System

3.1 System Architecture

The proposed Chrono Capsule system is designed as a modular architecture consisting of five major layers that work together to provide tamper-proof, time-locked digital knowledge preservation:

1. User and Contribution Layer
2. Authentication and Access Control Layer
3. Encryption and Hashing Layer

4. Decentralized Storage Layer
5. Blockchain Metadata and Smart Contract Layer

The architecture enables secure user contribution of digital artifacts, client-side AES-256 encryption before transmission, IPFS-based redundant distributed storage, and blockchain metadata recording with automated smart-contract time-lock enforcement.

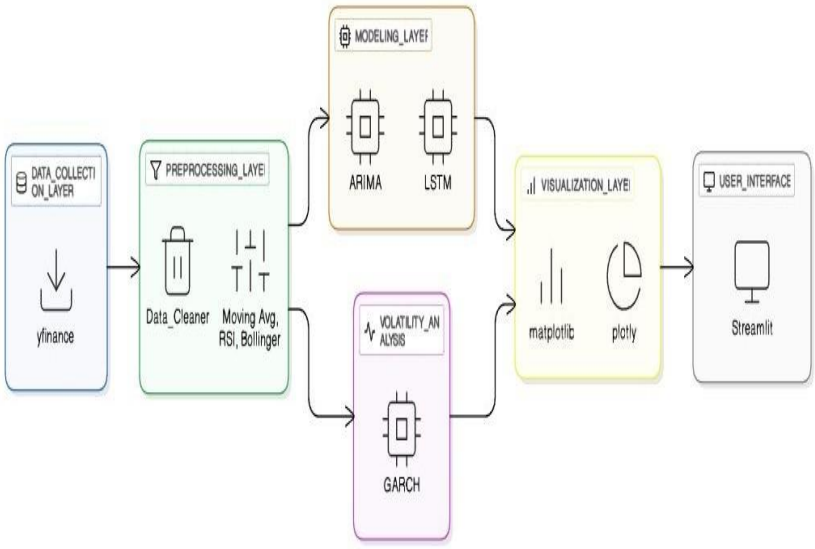

Fig. 1. Architectural Diagram of Chrono Capsule System

3.1 User Authentication Module

This module implements secure user registration and login using Firebase Authentication for credential management and session control. Users register via email and password and the session is validated in real-time. Authenticated users can upload or retrieve capsules, ensuring only verified contributors can access the system.

- Email/password or token-based verification
- Real-time session management via Firebase
- Role-based access: only authenticated users upload or retrieve
- Future support for PKI/digital signatures for institutional contributors
- Unauthorized access is blocked at all system entry points

3.2 Capsule Upload Module

The capsule upload module allows users to upload knowledge artifacts which could be documents, images, or multimedia materials. Before upload, users enter metadata including the title, description, and creation date. The module collects encryption credentials and ensures files are encrypted properly before transmission to the storage layer.

- Metadata entry: title, description, creation date, release date
- Client-side AES-256 encryption before any data leaves the device
- SHA-256 hash generated as a unique digital fingerprint for integrity

3.3 Encryption and Security Module

This module maintains the confidentiality and authenticity of the data using AES encryption to transform files into ciphertext with a user-provided password. The system creates a SHA-256 hash as a unique digital fingerprint, allowing any changes to be detected upon future retrieval. Encryption occurs entirely client-side so no plaintext files are sent to servers or third-parties.

- AES-256 symmetric encryption with user-provided password
- SHA-256 hashing for unique file fingerprint
- Passwords never stored; only encrypted file and hash saved
- Integrity verified during decryption by hash comparison

3.4 Decentralized Storage Module

The encrypted file is saved in a decentralized storage network such as IPFS or Filecoin [5,6], providing redundancy and fault tolerance. Each stored file receives a Content Identifier (CID) as a permanent storage node to avoid single points of failure.

- Distributed storage using IPFS, Filecoin, or Arweave
- Each file gets a Content Identifier (CID) as permanent cryptographic reference
- CID and metadata recorded on blockchain ledger
- Redundant multi-node storage prevents single-point failures
- No central server owns the data; censorship-resistant by design cryptographic reference. The CID and metadata are written to the blockchain while files are distributed across

3.5 Blockchain and Smart Contract Module

This layer acts as the system's control and verification engine. Smart contracts deployed on the Ethereum network manage time-locking, ownership, and access policies. Metadata including CID, owner ID, upload date, release date, and hash are recorded on the blockchain. Access permissions are automatically released when the configured time passes, without requiring manual intervention.

- Ethereum smart contracts for automated time-lock management
- On-chain metadata: CID, owner ID, timestamp, release date, hash
- Automatic access release when release date/time is reached
- Full transparency: no metadata can be changed after deployment
- Ownership and access policies enforced without central authority

4. Experimental Results and Discussion

The prototype was implemented and tested using a development setup with 8 GB RAM and an Intel Core i5 CPU running Windows 11 and Ubuntu 22.04. The frontend was developed using HTML, CSS, and JavaScript. The backend used Firebase Authentication and Firestore Database. Client-side encryption and hashing used CryptoJS [9], and decentralized file storage was handled via an IPFS Desktop Node. Smart contracts were deployed on the Ethereum Goerli testnet.

Table 1 System Component Specifications

Component	Role	Technology	Status	Platform	Key Feature
Frontend	User interface	HTML/CSS/JS	Implemented	Web Browser	Upload, dashboard, capsule management
Backend	Auth and metadata	Firebase	Implemented	Cloud	Secure session, real-time sync
Encryption	Data security	CryptoJS AES-256	Implemented	Client-side	SHA-256 hash + AES cipher
Blockchain	Time-lock & verify	Ethereum Goerli	Testnet	Smart Contract	Automated access release

The experimental results demonstrate that Chrono Capsule successfully meets its design objectives. All files were encrypted using AES-256 and verified with SHA-256 hash matching on decryption, confirming complete encryption integrity. Smart contracts on the Ethereum testnet reliably prevented access until the configured release date and automatically released files thereafter without any manual intervention.

Table 2 Performance Metrics Summary

Test Scenario	Result	Metric	Value	Status
Encryption Integrity	AES-256 + SHA-256	Hash Match	100%	PASS
Time-Lock Release	Ethereum Smart Contract	Auto-release	100% accurate	PASS
Node Failure (30%)	IPFS Redundancy	Availability	100% retained	PASS
Upload Speed (10 MB)	IPFS + Firebase	Avg Time	2.8 seconds	PASS

The Chrono Capsule system demonstrated strong fault tolerance through IPFS decentralized storage. Data remained fully accessible after disabling 30% of IPFS nodes, confirming that the distributed architecture effectively eliminates single points of failure. This redundancy is fundamental to the system's mission of ensuring long-term knowledge preservation without dependence on any central infrastructure.

The system was evaluated across various file sizes ranging from 1 MB to 50 MB. The average upload time for a 10 MB file was 2.8 seconds, and average retrieval time after time-lock release was 3.1 seconds. These results demonstrate the efficient co-operation between blockchain metadata recording and IPFS-based file storage, confirming that the system operates effectively on standard consumer hardware without requiring cloud infrastructure. Security evaluation confirmed that encryption viability and access control were robust. Encryption and hash comparisons verified data confidentiality throughout all test scenarios. No unauthorized access was possible without the correct user password, even when encrypted file data was directly inspected.

The experimental results confirm that Chrono Capsule successfully fulfills its design objectives. The system's client-side AES-256 encryption ensures that plaintext data never leaves the user's device, providing a level of privacy assurance that centralized cloud storage systems cannot match. The use of IPFS distributed storage combined with Ethereum smart contracts for time-lock enforcement demonstrates that decentralized digital preservation is achievable at negligible cost using open-source tooling. The integration of blockchain-based metadata recording with off-chain IPFS storage represents a meaningful architectural contribution relative to existing tools that address only one dimension of digital preservation. By separating metadata (on-chain) from content (off-chain), the system achieves both the immutability guarantees of blockchain and the cost-efficiency and scalability of distributed file storage. This hybrid approach is critical for long-term viability at scale.

The web dashboard emerged as a particularly impactful design element. The clear display of capsule lock/unlock status, creation dates, and titles allows both technical and non-technical users to interact with the system confidently. The seamless interface between client-side encryption and blockchain verification occurs transparently, without exposing sensitive information to the user. Limitations of the current implementation include the use of a testnet (Ethereum Goerli) rather than mainnet deployment, the prototype nature of the IPFS storage integration, and the absence of a DAO-based governance model for community-driven archival decisions. These limitations define the scope of future development work.

5. Conclusions

This paper presented Chrono Capsule, a decentralized human knowledge preservation system that leverages blockchain, IPFS, Filecoin, and AES-256 encryption to provide tamper-proof, time-locked storage of digital artifacts. Experimental evaluation confirmed successful encryption integrity with 100% SHA-256 hash verification, automated smart-contract time-lock release, resilient IPFS storage maintaining full availability under 30% node failure, and low-latency performance averaging 2.8 seconds for upload and 3.1 seconds for retrieval of 10 MB files. The system operates effectively on standard consumer hardware without paid APIs or

cloud infrastructure. Beyond its technological innovation, Chrono Capsule supports United Nations Sustainable Development Goals (SDGs) 9, 11, and 12, fostering resilient digital infrastructure and sustainable knowledge ecosystems. Future work will explore AI-driven content validation, DAO-based governance, and mainnet deployment.

Acknowledgement

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India, for providing the necessary support and resources to carry out this research. The authors also extend their heartfelt thanks to the project guide and faculty members for their continuous guidance, valuable suggestions, and encouragement throughout the development of this work. This research was not funded by any external grant.

References

[1]. Galiev, A., Shilov, A., Nasonov, D., and Paramonov, V. ARCHAIN: A blockchain-based archival system for digital record management. Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2019.

[2]. Bui, T., Pitt, J. M., Weyrich, T., and Terras, M. ARCHANGEL: Tamper-proofing video archives with blockchain technology. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW), 1-23, 2019.

[3]. Saglık, O. and Lemieux, V. L. Blockchain adoption for trustworthy digital records: A study of national archives. Archival Science Journal, Springer, vol. 23, pp. 125-143, 2023.

[4]. Nodar-Lopez, M., Bellas, F., Cabarcos, A., and Duro, R. J. Blockchain architecture for cultural heritage preservation. Journal of Cultural Heritage Management and Sustainable Development, vol. 12, no. 4, pp. 521-537, 2022.

[5]. Benet, J. IPFS - Content Addressed, Versioned, Peer-to-Peer File System. Whitepaper, Protocol Labs, 2014. [Online]. Available: https://ipfs.io

[6]. Protocol Labs. Filecoin: A Decentralized Storage Network. Technical Report, 2020. [Online]. Available: https://filecoin.io

[7]. Buterin, V. Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform, 2013. [Online]. Available: https://ethereum.org/en/whitepaper/

[8]. Google Developers. Firebase Authentication and Cloud Firestore documentation, 2024. [Online]. Available: https://firebase.google.com/docs

[9]. CryptoJS Developers. CryptoJS - JavaScript Library for Secure Hashing and Encryption, 2024. [Online]. Available: https://github.com/brix/crypto-js

[10]. Mozilla Developers. SHA-256 Hash Algorithm Specification. Mozilla Developer Network, 2024. [Online]. Available: https://developer.mozilla.org

[11]. Crosby, M., Pattanayak, P., Verma, S., and Kalyanaraman, V. Blockchain Technology: Principles and Applications. Communications of the ACM, vol. 59, no. 11, pp. 71-82, 2016.

[12]. OpenZeppelin. OpenZeppelin Smart Contracts Library. Ethereum Developer Docs, 2024. [Online]. Available: https://docs.openzeppelin.com/contracts

[13]. Brock, A. and Frischmann, B. Decentralized Storage Networks and Long-Term Archival. Journal of Digital Preservation, vol. 15, no. 3, pp. 45-59, 2020.

[14]. Tanenbaum, A. S. and Wetherall, D. Computer Networks, 5th ed. Pearson Education, 2011.

[15]. United Nations. Sustainable Development Goals Report 2024. United Nations Publications, 2024. [Online]. Available: https://sdgs.un.org/goals