Gas-Efficient Time-Decaying Attribute-Based Access Control (TD-ABAC) for EHR using Hybrid Blockchain

Suhas Thammysetty
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
st2721@srmist.edu.in

Velichelmala Harshith Reddy
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai , India
hv8030@srmist.edu.in

Dr. Subash R
Dept. of Computing Technologies
SRM Institute of Science & Technology
Chennai, India
subashr@srmist.edu.in

***Abstract*—The secure sharing of Electronic Health Records (EHRs) remains a critical challenge due to stringent privacy requirements, high computational overhead, and costly access revocation mechanisms in existing blockchain-based healthcare systems. Most current solutions rely on computationally intensive cryptographic techniques such as Ciphertext-Policy Attribute-Based Encryption (CP-ABE), combined with active revocation strategies that require frequent key updates and incur high gas costs. This paper proposes a gas-efficient hybrid blockchain-based access control framework that introduces time-decaying permissions for EHR access. The proposed system offloads encryption operations to an off-chain backend using lightweight AES-256 encryption, while on-chain smart contracts act as a trusted timekeeper to enforce access expiration. A novel concept of passive revocation is implemented, where access permissions automatically expire based on blockchain timestamps without requiring any explicit revocation transactions. Experimental evaluation demonstrates that the proposed approach significantly reduces gas consumption, improves access latency, and simplifies key management compared to traditional CP-ABE-based solutions, making it more suitable for real-world healthcare deployments.**

I. Introduction

The adoption of Electronic Health Records (EHRs) has transformed modern healthcare by enabling efficient storage and seamless sharing of patient data among authorized entities. However, EHR systems manage highly sensitive information, making data privacy, security, and controlled access critical challenges. Conventional centralized access control mechanisms depend on trusted third parties and are vulnerable to single points of failure, insider attacks, and lack of transparency. These limitations have motivated the integration of blockchain technology into healthcare systems due to its decentralized, immutable, and auditable nature.

Existing blockchain-based EHR access control models often rely on computationally intensive cryptographic techniques such as Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to achieve fine-grained access control. While effective in enforcing attribute-based policies, CP-ABE introduces significant computational overhead and latency. Moreover, access revocation in such systems is typically active, requiring key regeneration and data re-encryption, which leads to increased system complexity and high gas costs. Managing time-limited access remains another major challenge, as real-world healthcare scenarios frequently require temporary data sharing with doctors, interns, or specialists.

To overcome these challenges, this paper proposes a Gas-Efficient Time-Decaying Attribute-Based Access Control (TD-ABAC) system using a Hybrid Blockchain architecture. The proposed solution introduces a passive revocation mechanism, where access permissions automatically expire based on blockchain timestamps without requiring explicit revocation transactions. Lightweight AES-256 encryption is performed off-chain using a backend service, while smart contracts are used solely to enforce access policies and time constraints. This hybrid approach significantly reduces gas consumption, improves performance, and simplifies key management, making it well-suited for practical and scalable EHR systems.

II. Literature Review

Blockchain-based access control mechanisms for Electronic Health Record (EHR) systems have attracted significant attention because of their decentralized trust model and resistance to tampering. Several studies have examined how to integrate blockchain with cryptographic access control methods to ensure secure and auditable sharing of medical data. Ekblaw et al. proposed MedRec, a framework that manages access permissions for medical records while maintaining data ownership and transparency [1]. Similarly, Zhang et al. introduced FHIRChain, which uses blockchain and smart contracts to allow secure and interoperable sharing of clinical data among healthcare providers [2].

To achieve fine-grained access control over encrypted health records, many systems use Ciphertext-Policy Attribute-Based Encryption (CP-ABE). Bethencourt et al. first introduced CP-ABE to enforce attribute-based policies directly within encrypted data [3]. Chen et al. then built on this idea, proposing a flexible and fine-grained access control framework for EHR systems in blockchain-assisted healthcare settings. Their framework combines CP-ABE with blockchain to enforce

policies over encrypted medical records [4]. Their approach enables decentralized policy enforcement and improves data sharing flexibility in healthcare systems. However, it still relies on computationally heavy bilinear pairing operations used in CP-ABE, which raises encryption and decryption overhead, especially for large datasets.

Several studies have tried to create time-based access control mechanisms to meet changing permission needs. Fu et al. proposed key regression techniques for time-based cryptographic key management, which allow efficient generation of multiple keys for different time intervals [5]. Later research used similar ideas in secure data sharing frameworks, where access rights are linked to time attributes [6]. Although these approaches provide time-limited access, they markedly increase key management complexity and server-side processing. More critically, access revocation in these systems usually requires active measures, which involve re-encrypting stored data or redistributing cryptographic keys when permissions change. This leads to added computing overhead and higher transaction costs in blockchain environments.

Recent research has examined hybrid architectures that split heavy cryptographic tasks from blockchain functions to enhance system scalability. Xia et al. proposed a blockchain-enabled framework for sharing healthcare data that keeps encrypted medical records off-chain while managing access control metadata on-chain [7]. While such hybrid designs lower storage and processing costs, most available solutions still depend on active revocation methods and do not fully utilize blockchain's native features like immutable timestamps for efficient time-based enforcement.

Thus, there is still a need for an access control framework that allows efficient time-based enforcement without costly re-encryption or key redistribution. This paper introduces a time-decaying attribute-based access control (TD-ABAC) model that uses blockchain timestamps for passive revocation. By substituting state-changing revocation tasks with a constant-time timestamp condition, the proposed system removes expensive revocation transactions while ensuring secure and verifiable access control for encrypted healthcare data.

TABLE I: Comparison of Encryption Techniques Used in EHR Systems

Feature	AES	CP-ABE
Encryption Type	Symmetric	Attribute-based Asymmetric
Key Size	128/192/256 bits	Multiple keys
Encryption Speed	Very Fast	Slow
Decryption Complexity	Low	High
Suitability for Blockchain	High (Off-chain)	Low
Real-World Adoption	Very High	Limited

III. Methodology

The proposed system follows a hybrid architecture that divides responsibilities between off-chain and on-chain components to optimize performance, scalability, and cost efficiency for secure electronic health record (EHR) sharing.

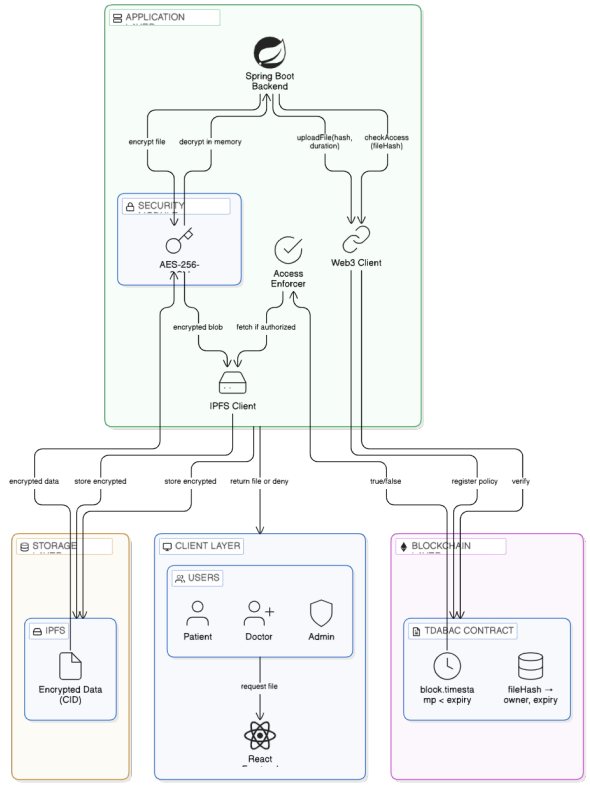

Fig. 1: Hybrid architecture of the proposed TD-ABAC system showing the interaction between the client layer, backend encryption service, IPFS storage, and blockchain-based access control.

The architecture consists of four primary layers: the client layer, application layer, storage layer, and blockchain layer. Each layer performs a specific role in ensuring secure data handling and efficient access control.

3.1. Client Layer

The client layer represents the user interface through which different actors interact with the system. The frontend is implemented using a React-based web application that enables users such as patients, doctors, and administrators to upload medical records, request access to files, and manage permissions. All user requests are forwarded to the backend server where authorization and data processing operations are performed.

3.2. Off-Chain Encryption Layer

The application layer is implemented using a Java Spring Boot backend which performs encryption, authorization enforcement, and communication with the blockchain network.

A dedicated security module performs cryptographic operations using the AES-256-GCM symmetric encryption algorithm. This layer is responsible for:

Encrypting EHR files before storage

Securely managing symmetric encryption keys

Releasing decryption keys only after authorization from the blockchain

Encrypted files are handled in memory during processing to ensure that sensitive information is never stored in plaintext within the system.

3.3. Storage Layer

The storage layer uses the InterPlanetary File System (IPFS) to store encrypted medical records. Instead of storing the actual file on the blockchain, the system uploads the encrypted data to IPFS and receives a unique content identifier (CID). This CID acts as a reference to the stored encrypted file.

By storing only the CID on the blockchain, the system significantly reduces blockchain storage requirements and improves scalability while ensuring that files remain encrypted in decentralized storage.

3.4. On-Chain Access Control Layer

An Ethereum smart contract is deployed to manage access permissions and enforce the Time-Decaying Attribute-Based Access Control (TD-ABAC) policy. Instead of storing sensitive data, the contract stores metadata associated with each file, including:

- **User identifiers**
- **File identifiers**
- **Access expiry timestamps**

The smart contract verifies whether the current blockchain timestamp is less than the stored expiry time before approving access requests. This mechanism ensures transparent and tamper-resistant access control.

3.5. Passive Revocation Mechanism

Unlike traditional systems, access revocation does not involve key regeneration or contract updates. Once the expiry time is reached, the smart contract automatically denies access, effectively revoking permissions without any computational or monetary cost. This passive revocation mechanism significantly reduces gas consumption and simplifies key management.

3.6. Secure Data Access Workflow

The system follows a structured workflow for secure file sharing:

1) A user uploads an EHR file through the client interface.
2) The backend encrypts the file using AES-256-GCM.
3) The encrypted file is stored in IPFS and a content identifier (CID) is generated.
4) The CID and access metadata are registered in the blockchain smart contract.
5) When a user requests access, the backend queries the smart contract to verify whether the current timestamp satisfies the access condition.
6) If authorization is granted, the encrypted file is retrieved from IPFS and decrypted in memory before being delivered to the user.

This architecture ensures that sensitive healthcare data remains encrypted during storage and transmission while access permissions are securely enforced using blockchain-based verification.

IV. Experimental Setup

The experimental evaluation was conducted to analyze the performance, scalability, and cost efficiency of the proposed TD-ABAC system. The experiments were executed on a local development environment with the following hardware and software configuration.

Hardware Configuration

The experiments were performed on a system equipped with an Intel i3-1005G1 processor, 12 GB RAM, running Ubuntu 22.04 operating system. This setup represents a typical mid-range computing environment suitable for backend services and blockchain development.

Software Environment

The system was implemented using the following software components:

- **Blockchain Network:** A Hardhat-based local Ethereum network was used for deploying and testing the smart contract. This environment provides a controllable blockchain simulation for measuring transaction latency and gas consumption.
- **Backend Server:** A Java Spring Boot application running on Java 17 was used to implement the application logic, including encryption, decryption, IPFS communication, and blockchain interaction.
- **Cryptographic Module:** AES-256-GCM symmetric encryption was used for secure file encryption and decryption operations before storing files in decentralized storage.
- **Storage Layer:** The InterPlanetary File System (IPFS) was used as a decentralized storage network to store encrypted files. Each encrypted file is associated with a unique content identifier (CID) which is referenced by the blockchain smart contract.
- **Smart Contract Framework:** Solidity smart contracts were developed and deployed using the Hardhat development framework. The smart contract implements the TD-ABAC policy logic and timestamp-based access validation.
- **Client Entities:** Simulated users representing patients and doctors were used to generate file upload and access requests through the system frontend.

Performance Evaluation Metrics

Multiple experiments were conducted to evaluate the performance characteristics of the proposed system. The evaluation focused on the following metrics:

- **Encryption and Decryption Time:** The time required for AES-256 encryption and decryption operations for different file sizes was measured to evaluate the computational overhead of the off-chain cryptographic layer.
- **Gas Consumption:** The gas cost associated with smart contract operations such as file registration and access verification was measured to analyze the economic efficiency of the system.
- **Access Verification Latency:** The latency of the smart contract access-check function was evaluated for different access durations to verify the constant-time behavior of the timestamp-based access policy.

V. Results and Analysis

The experimental results show that the proposed hybrid system significantly outperforms traditional CP-ABE-based approaches:

- Gas Efficiency: Passive revocation eliminates the need for revocation transactions, resulting in zero gas cost for access expiry, whereas the baseline system incurs high gas fees for key updates.

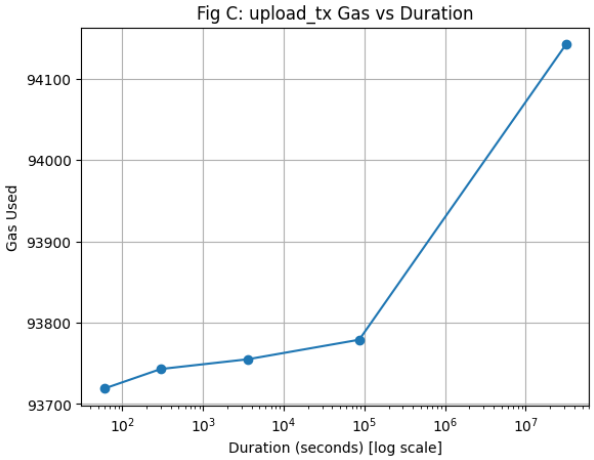

Fig. 2: Upload transaction gas consumption for different access durations.

TABLE II: Upload Transaction Gas Consumption

Duration (s)	Gas Used
60	93719
300	93743
3600	93755
86400	93779
31536000	94142

- Performance: AES-based encryption achieves millisecond-level encryption and decryption times, compared to seconds required for CP-ABE operations.

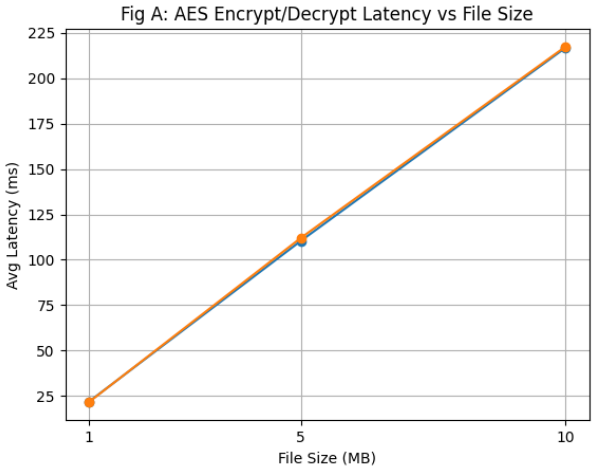

Fig. 3: Smart contract access check latency vs File Size.

TABLE III: AES Encryption and Decryption Performance

File Size	Avg Latency (ms)	p95 Latency (ms)	Throughput (MB/s)
1 MB	21.9390	23.9576	45.58
5 MB	110.3837	128.0745	45.30
10 MB	216.7240	238.5819	46.14
1 MB	21.7291	23.4439	46.02
5 MB	112.0210	134.2816	44.63
10 MB	217.3527	241.6556	46.01

- Scalability and Reliability: The absence of complex key chains reduces system complexity and minimizes the risk of key management failures.

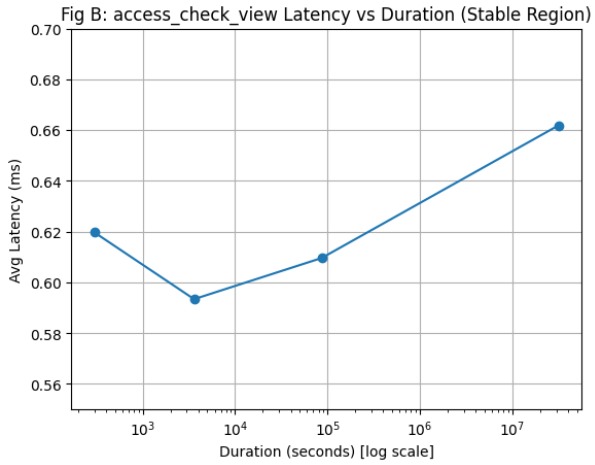

Fig. 4: Smart contract access check latency across different validity durations.

TABLE IV: Access Check Latency Across Different Durations

Duration (s)	Avg Latency (ms)	p95 Latency (ms)	Gas Used
60	0.9453	1.0850	28190
300	0.6197	0.7060	28202
3600	0.5934	0.8690	28214
86400	0.6096	0.8185	28226
31536000	0.6618	0.9049	28411

- Key Performance: AES encryption/decryption latency, access-check latency, throughput, and gas consumption. The results show stable encryption throughput (45–46 MB/s), sub-millisecond access verification, and bounded upload transaction gas costs (93k–94k gas), demonstrating efficient and predictable system performance.

TABLE V: Key Performance Metrics of the TD-ABAC System

Metric	Avg Latency	p95 Latency	Throughput	Gas Used
AES Encryption (10MB)	216.72 ms	238.58 ms	46.14 MB/s	–
AES Decryption (10MB)	217.35 ms	241.66 ms	46.01 MB/s	–
Access Check (view)	0.61 ms	0.90 ms	–	28,190
Upload Transaction	–	–	–	93,719–94,142
Revocation	0 ms	0 ms	–	0

Overall, the results confirm that leveraging blockchain timestamps for access control provides a secure, efficient, and scalable solution for EHR data sharing.

REFERENCES

[1] A. Ekblaw, A. Azaria, J. D. Halamka, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in Proc. IEEE Open & Big Data Conf., 2016.

[2] P. Zhang, J. White, D. Schmidt, G. Lenz, and S. Rosenbloom, "FHIR-Chain: Applying blockchain to securely and scalably share clinical data," Computational and Structural Biotechnology Journal, vol. 16, pp. 267–278, 2018.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symposium on Security and Privacy, 2007.

[4] D. Chen, N. Zhang, L. Zhang, Z. Liao, H.-N. Dai, X. Shen, and M. Pang, "Flexible and Fine-Grained Access Control for EHR in Blockchain-Assisted E-Healthcare Systems," IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 6, pp. 2640-2651, 2022.

[5] K. Fu, S. Kamara, and T. Kohno, "Key regression: Enabling efficient key distribution for secure distributed storage," in Proc. NDSS, 2006.

[6] Ciphertext-policy attribute-based delay encryption. International Journal of Information and Computer Security, 2023. Available: https://www.inderscience.com/info/inarticle.php?artid=134960

[7] BSDS-ShareCrypt: A blockchain-enabled framework for secure, anonymous, and accountable data sharing in cloud-IoT environments using aggregate key searchable encryption. Available: https://link.springer.com/article/10.1007/s10207-025-01163-4

[8] Secure electronic health record access control via blockchain, dual-attribute encryption, and large language model-based attribute extraction. Scientific Reports. Available: https://www.nature.com/articles/s41598-026-39690-2