



Enhanced Blockchain-Based Privacy Preservation of Healthcare Insurance Data Using Optimized Elliptic Curve Cryptography

Mahammad Sathik Karimuddin¹, Dr. K. V. Ramana², K Ravi Kiran³

¹ PG Scholar, Department of Computer Science and Engineering, UCEK, JNTUK, Kakinada, Andhra Pradesh, India.

² Senior Professor, Department of Computer Science and Engineering, UCEK, JNTUK, Kakinada, Andhra Pradesh, India.

³ Assistant Professor, Department of Computer Science and Engineering, UCEK, JNTUK, Kakinada, Andhra Pradesh, India.

How to cite this paper:

Mahammad Sathik Karimuddin¹, Dr. K. V. Ramana², K Ravi Kiran³, "Enhanced Blockchain-Based Privacy Preservation of Healthcare Insurance Data Using Optimized Elliptic Curve Cryptography", IJIRE-V7I4-193-205.



Copyright © 2026
by author(s) and
Fifth Dimension
Research

Publication. This work is licensed under the
Creative Commons Attribution International
License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>

Abstract: The growing use of digital healthcare services has greatly improved access to medical information. However, it has also created serious challenges in protecting the privacy and integrity of healthcare insurance data. Traditional centralized storage systems can be vulnerable to unauthorized access, data tampering, and security breaches. This limits their ability to manage sensitive patient records. This paper introduces a blockchain-based framework designed to protect the privacy of healthcare insurance data using optimized Elliptic Curve Cryptography (ECC). In this framework, healthcare records are encrypted with an optimized ECC algorithm before being stored on a blockchain. This ensures secure and tamper-proof data management. A role-based access mechanism allows secure interactions between healthcare providers and insurance organizations while blocking unauthorized access to private information. The framework also includes secure cryptographic key management and verifies blockchain integrity to protect encrypted records during the data-sharing process. Experimental results show that this approach offers secure storage, controlled access, and efficient cryptographic performance while maintaining the confidentiality and integrity of healthcare insurance data. The framework provides a practical and scalable solution for sharing healthcare information securely in today's insurance environments.

Key Words: Blockchain, Healthcare Insurance, Privacy Preservation, Elliptic Curve Cryptography, Secure Data Sharing, Role-Based Access Control.

I. INTRODUCTION

The rapid growth of digital healthcare technologies has changed how medical information is created, kept, and shared among healthcare providers, insurance organizations, and patients. Electronic health records and healthcare insurance systems have made medical services more efficient by allowing quicker access to patient information and streamlining insurance claim processing [10], [11]. However, the rising amount of digital healthcare data has led to serious concerns about privacy, confidentiality, and data integrity. Healthcare records hold very sensitive personal and medical information, which makes them appealing targets for unauthorized access, data manipulation, and cyberattacks. Protecting this information has become a basic need for modern healthcare systems.

Most healthcare information systems depend on centralized data management models where a single organization or service provider handles patient records [12]. While centralized systems make administration easier, they also bring several security issues. These include single points of failure, limited transparency, and a higher risk of insider threats and external attacks [8]. Unauthorized changes or exposure of healthcare insurance records can result in financial losses, identity theft, insurance fraud, and breaches of patient privacy. As a result, there is a growing need for secure data management methods that maintain confidentiality while ensuring data integrity and controlled access. Blockchain technology has emerged as a promising solution for addressing many of these challenges [10], [15]. By maintaining a distributed and tamper-resistant ledger, blockchain improves data integrity and provides transparent verification of stored records [11]. However, blockchain alone cannot guarantee the confidentiality of sensitive healthcare information. Data stored on the blockchain may still need strong cryptographic protection before storage [2]. Therefore, combining blockchain with effective encryption techniques is essential for achieving privacy preservation in healthcare applications.

Elliptic Curve Cryptography (ECC) has gained significant attention due to its ability to provide strong security with relatively smaller key sizes compared to traditional public-key cryptographic algorithms [1], [3]. The reduced computational burden of ECC makes it particularly suitable for healthcare environments, where secure data sharing must happen without introducing long processing delays. Nevertheless, secure encryption alone is not enough unless it is backed by effective key

management and controlled access mechanisms that ensure only authorized users can retrieve protected healthcare information [9]. Motivated by these challenges, this paper presents a better blockchain-based system for protecting privacy in healthcare insurance data using optimized Elliptic Curve Cryptography. In the proposed system, healthcare records are encrypted before they are stored as blockchain blocks. This prevents unauthorized access to sensitive information. A role-based access method is used to ensure secure interactions between healthcare providers and insurance companies. Additionally, secure management of cryptographic keys protects encrypted data during sharing. The framework also includes a verification process to detect unauthorized changes and maintain the authenticity of stored records. Together, these components create a safe environment for sharing healthcare insurance data while keeping confidentiality, integrity, and controlled access.

II. PROPOSED METHODOLOGY

2.1 System Architecture

The proposed healthcare insurance data management system aims to provide secure storage, controlled access, and protection of sensitive healthcare records. It uses blockchain technology and Optimized Elliptic Curve Cryptography (ECC). The overall setup includes a React.js-based frontend, a Flask-based backend, an ECC encryption module, blockchain storage, and a real-time monitoring dashboard. The system has a modular design where each part performs a specific function, ensuring secure management of healthcare data. Authorized users first access the system through a secure login interface. The Role-Based Access Control (RBAC) mechanism categorizes users into two groups: Doctors and Insurance Providers [9], [12].

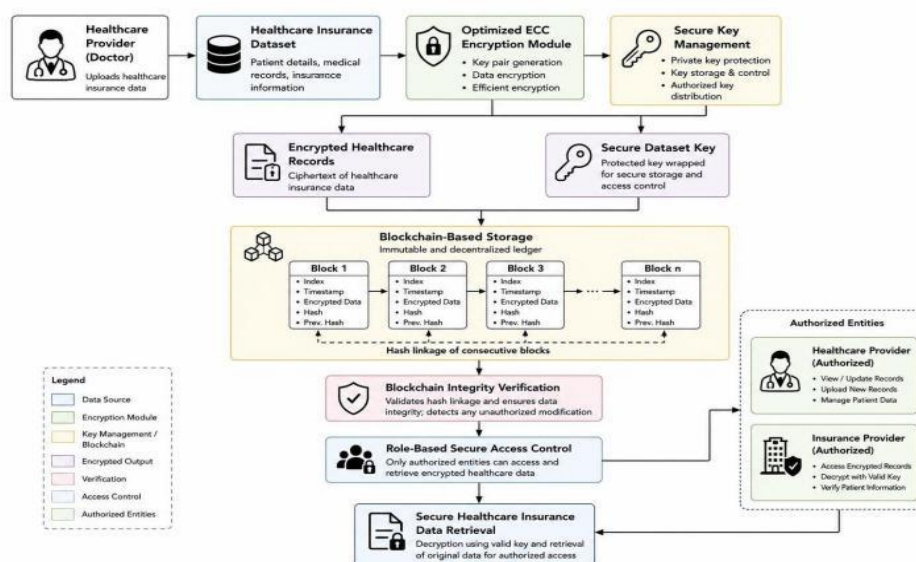


Fig. 1. System architecture of the proposed enhanced blockchain-based privacy preservation framework for healthcare insurance data using optimized Elliptic Curve Cryptography.

Doctors are in charge of uploading healthcare datasets to the system. Insurance Providers can only securely access healthcare records after verifying their private keys. This separation of roles reduces unauthorized access and improves overall system security. When a doctor uploads a healthcare dataset in CSV format, the Flask backend checks the file before processing it. The healthcare records are then encrypted using the Optimized ECC module to keep the information confidential. After encryption, the encrypted records are organized into blockchain blocks. Each block includes the encrypted healthcare data, timestamp, previous block hash, and current block hash, generated using the SHA-256 hashing algorithm. This blockchain setup ensures that any unauthorized changes can be easily detected [10], [13]. To improve transparency, the proposed system includes a Blockchain Explorer module that allows authorized users to view blockchain information such as block index, timestamp, current hash, and previous hash. Instead of revealing confidential healthcare details, only blockchain metadata is shown. This approach maintains privacy while allowing for integrity checks [10], [11]. The system also has a secure method for retrieving datasets. Before downloading healthcare records, the Insurance Provider must enter a valid private key. The backend checks the entered private key against the securely stored key information. The requested healthcare dataset becomes available for download only after this verification, which prevents unauthorized access to data [13], [14].

To ensure blockchain integrity continually, an automated tampering detection module monitors the blockchain regularly. It recalculates and validates cryptographic hash values. If it detects any unauthorized changes within the blockchain, the dashboard updates the blockchain status from "Secure" to "Tampered" and increases the tampering event counter. This monitoring helps identify security violations quickly and builds trust in the healthcare data management process [8], [14].

The proposed architecture combines secure authentication, optimized ECC encryption, blockchain-based immutable

storage, SHA-256 hashing, role-based access control, blockchain visualization, private key verification, and automated tampering detection into a single framework for managing healthcare insurance. This complete system improves confidentiality, integrity, transparency, and controlled access to sensitive healthcare insurance records.

2.2 Optimized Elliptic Curve Cryptography (ECC)

Healthcare insurance records contain very sensitive patient information that needs strong cryptographic protection before being stored on the blockchain. In the proposed system, Optimized Elliptic Curve Cryptography (ECC) is used to encrypt healthcare records before storing them on the blockchain. The implementation uses the BrainpoolP256r1 elliptic curve from the TinyEC cryptographic library. ECC offers the same level of security as traditional public-key cryptography but with much smaller key sizes. This leads to lower computational overhead, reduced memory usage, and faster execution time [1], [3]. The elliptic curve is defined over a finite field F_p and is represented by

$$y^2 = x^3 + ax + b \pmod{p} \quad (1)$$

where a and b are the curve parameters, and p is a large prime number defining the finite field.

To ensure a valid elliptic curve, the following condition must be satisfied:

$$4a^3 + 27b^2 \neq 0 \pmod{p} \quad (2)$$

Initially, a private key is randomly generated as

$$d \in [1, n - 1] \quad (3)$$

where n denotes the order of the elliptic curve.

The corresponding public key is generated using scalar multiplication:

$$Q = dG \quad (4)$$

where d represents the private key, G denotes the generator point, and Q represents the public key.

During encryption, a random ephemeral key k is generated. The shared secret is computed as

$$S = kQ \quad (5)$$

where Q is the receiver's public key.

The first ciphertext component is generated as

$$C_1 = kG \quad (6)$$

The proposed implementation encrypts every character of the healthcare record using the x -coordinate of the shared secret.

The encryption process is expressed as

$$C_i = M_i \oplus S_x \quad (7)$$

where M_i is the ASCII value of the i -th character, S_x is the x -coordinate of the shared secret, and $\oplus$ denotes the XOR operation.

The encrypted healthcare record is therefore represented as

$$C = (C_1, C_i) \quad (8)$$

During dataset retrieval, the receiver regenerates the shared secret using the private key

$$S = dC_1 \quad (9)$$

The original healthcare record is recovered using

$$M_i = C_i \oplus S_x \quad (10)$$

where S_x represents the x -coordinate of the regenerated shared secret.

Unlike the base framework, which combines Improved ECC with deep learning methods for key tuning and optimization algorithms [1], the proposed implementation uses a lightweight Optimized ECC mechanism. The optimization mainly focuses on efficient key generation, faster encryption and decryption times, and easier implementation while maintaining strong cryptographic security. We also measure the time needed for key generation, encryption, and decryption to assess the computational efficiency of the proposed algorithm.

After encryption, the healthcare records are securely stored as blockchain blocks linked with SHA-256 hash values. Combining Optimized ECC with blockchain technology provides confidentiality, integrity, efficient cryptographic processing, and secure management of healthcare insurance data [2], [6].

Algorithm 1: Optimized ECC Encryption

Input: Healthcare Dataset (CSV)

Output: Encrypted Blockchain Records

Step 1: Read the uploaded healthcare dataset.

Step 2: Generate ECC private key (d).

Step 3: Compute public key $Q = dG$.

Step 4: Generate a random ephemeral key (k).

Step 5: Compute shared secret $S = kQ$.

Step 6: Encrypt each healthcare record using XOR with the x -coordinate of the shared secret.

Step 7: Generate ciphertext (C_1, C_i).

Step 8: Store encrypted records in the blockchain.

Step 9: Generate SHA-256 hash for each blockchain block.

Step 10: Return encrypted blockchain records.

2.3 Blockchain-Based Secure Storage

Blockchain technology offers a decentralized and tamper-resistant way to securely store healthcare insurance records [10], [15]. In this system, blockchain is used to maintain the integrity, authenticity, and traceability of encrypted healthcare data. After encrypting the healthcare dataset with the Optimized Elliptic Curve Cryptography (ECC) module, the encrypted records are arranged into blockchain blocks. Each block is linked to the previous one using the SHA-256 hashing algorithm. This setup ensures that any unauthorized changes can be detected right away.

Each blockchain block contains these components: Block Index, Timestamp, Encrypted Healthcare Data, Previous Block Hash, and Current Block Hash.

The structure of a blockchain block can be represented as

$$\text{Block}_i = (\text{Index}, \text{Timestamp}, \text{Data}, \text{PreviousHash}, \text{Hash}) \quad (11)$$

where Index represents the unique block number. Timestamp stores the time when the block was created. Data contains the encrypted healthcare records. PreviousHash stores the hash value of the previous block. Hash represents the cryptographic hash of the current block.

The hash value of every blockchain block is computed using the SHA-256 cryptographic hashing algorithm as

$$\text{Hash}_i = \text{SHA256}(\text{Block}_i) \quad (12)$$

The relationship between consecutive blockchain blocks is expressed as

$$\text{PreviousHash}_i = \text{Hash}_{i-1} \quad (13)$$

where the current block stores the hash value of its previous block. This cryptographic link ensures blockchain immutability because any change in a previous block alters its hash value, thereby invalidating all subsequent blockchain blocks. During the upload process, each encrypted healthcare record is turned into a blockchain block. The first block, called the Genesis Block, starts with a predefined previous hash value. Every following block stores the hash of its immediate predecessor, thus keeping the integrity of the blockchain.

The proposed implementation uses the SHA-256 hashing algorithm to create unique hash values for each blockchain block. Since SHA-256 produces a fixed-length 256-bit digest, even a small change in the stored healthcare data results in a completely different hash value. Therefore, any unauthorized changes to blockchain records are detected automatically by comparing the stored hash values with newly calculated hash values during blockchain verification. The proposed blockchain implementation also has an automatic integrity monitoring mechanism. When blockchain verification occurs, the system recalculates the hash value of each block and compares it with the previously stored hash value. If any mismatch is found, the blockchain status is instantly updated from Secure to Tampered, and the tampering event counter shown on the dashboard increases. This mechanism allows for real-time detection of harmful modifications and improves the trustworthiness of healthcare insurance data [8], [11].

By combining blockchain technology with Optimized ECC encryption, the proposed framework ensures secure storage, unchangeable record management, cryptographic integrity, and clear verification of healthcare insurance information.

Algorithm 2: Blockchain Storage

Input: Encrypted Healthcare Records

Output: Immutable Blockchain

Step 1: Receive encrypted healthcare records.

Step 2: Create the Genesis Block.

Step 3: Store encrypted healthcare data.

Step 4: Generate SHA-256 hash.

Step 5: Store previous block hash.

Step 6: Append block to blockchain.

Step 7: Repeat for all healthcare records.

Step 8: Save blockchain.

Step 9: Verify blockchain integrity periodically.

Step 10: Report tampering if hash mismatch occurs.

2.4 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is used in the proposed healthcare insurance management system to limit unauthorized access to sensitive healthcare records [9], [12]. Instead of giving all users the same privileges, the framework assigns permissions based on specific user roles. This approach improves confidentiality, accountability, and system security by ensuring each user can only perform allowed actions. The system defines two main user roles: Doctor and Insurance Provider. First, every user must log in with a valid User ID and password. Once authenticated, the system determines the user role and grants permissions accordingly. The Doctor can upload healthcare datasets, view blockchain information, and download healthcare records after verifying their private key. The Insurance Provider can only download healthcare records after successfully logging in and verifying their private key. Insurance Providers cannot upload healthcare datasets or access blockchain information. The authentication process can be represented as

$$\text{Auth}(\text{User}) = 1, \text{ if User ID and Password are valid; } 0, \text{ otherwise.} \quad (14)$$

where $\text{Auth}(\text{User})=1$ indicates successful authentication and $\text{Auth}(\text{User})=0$ indicates authentication failure. After authentication, access permission is determined according to the assigned role.

$$\text{Access} = f(\text{Role}, \text{Permission}) \quad (15)$$

where Role represents the authenticated user role and Permission denotes the allowed system operations. The access control policy implemented in the proposed system is summarized in Table 1.

Module	Doctor	Insurance Provider
Login	✓	✓
Dashboard	✓	✓
Upload Dataset	✓	✗
Download Dataset	✓	✓
Blockchain Explorer	✓	✗
Private Key Verification	✓	✓
Logout	✓	✓

Table 1. Role-Based Access Control Matrix

By combining Role-Based Access Control with Optimized ECC encryption and blockchain technology, the proposed framework improves confidentiality, stops unauthorized actions, and ensures secure access management for healthcare insurance records.

2.5 Dataset Upload Module

The Dataset Upload Module enables authorized doctors to securely upload healthcare insurance datasets to the proposed system. Access to this module is restricted through the Role-Based Access Control (RBAC) mechanism, ensuring that only authenticated doctors can upload healthcare records. The uploaded dataset is provided in CSV format and is validated by the Flask backend before processing. After successful validation, the healthcare records are encrypted using the Optimized ECC module and stored as blockchain blocks linked through SHA-256 hash values, thereby ensuring secure and tamper-resistant storage.

The upload process can be represented as

$$D_{en}^c = \text{ECC}(D) \quad (16)$$

where D represents the original healthcare dataset and D_{en}^c represents the encrypted healthcare dataset. Following encryption, the blockchain block is generated as

$$\text{Block}_i = (\text{Index}, \text{Timestamp}, D_{en}^c, \text{PreviousHash}, \text{Hash}) \quad (17)$$

After successful blockchain insertion, the healthcare dashboard automatically updates to show the total number of healthcare records, blockchain blocks, blockchain status, and detected tampering events. This allows authorized users to monitor the current state of the blockchain in real time. By combining dataset validation, optimized ECC encryption, blockchain storage, and dashboard synchronization into a single workflow, the proposed Dataset Upload Module offers secure, reliable, and efficient storage of healthcare insurance records while maintaining data confidentiality and integrity.

2.6 Blockchain Explorer

The Blockchain Explorer module offers a clear interface for viewing blockchain information without revealing sensitive healthcare records. It allows authorized users to look at blockchain metadata, which helps verify data integrity and the structure of the blockchain. The module shows important details for each blockchain block, such as the Block Index, Timestamp, Current Hash, and Previous Hash. Because it only displays blockchain metadata, the confidentiality of encrypted healthcare records remains intact while users can still confirm the integrity and continuity of the blockchain. The link between consecutive blockchain blocks is illustrated as

$$\text{PreviousHash}_i = \text{Hash}_{i-1} \quad (18)$$

where Hash_{i-1} denotes the hash value of the previous blockchain block. This cryptographic linkage ensures that every block is connected to its predecessor, making unauthorized modifications immediately detectable.

Fig. 2 illustrates the Blockchain Explorer interface, which displays the blockchain metadata including block index, timestamp, current hash, and previous hash for blockchain verification. The Blockchain Explorer fetches blockchain information from the Flask backend and shows the blockchain contents through the React-based user interface. Whenever a new healthcare dataset is uploaded, the blockchain updates automatically, making the latest information available for visualization. The Blockchain Explorer improves transparency, helps with blockchain verification, and allows users to check the integrity of healthcare records without risking their confidentiality [11], [13].

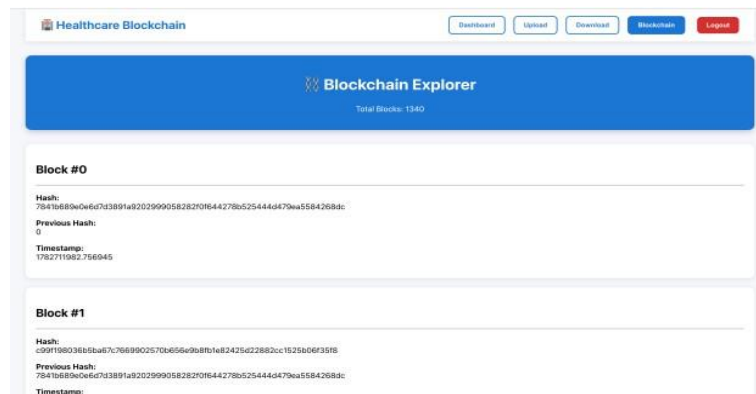


Fig. 2. Blockchain Explorer interface displaying block metadata for integrity verification.

2.7 Private Key Verification

The proposed healthcare insurance management system includes a key verification mechanism to ensure that only authorized insurance providers can access encrypted healthcare records. Before downloading the stored dataset, users must provide the generated dataset key. The Flask backend pulls the key identifier from the submitted key file and compares it to the corresponding identifier stored during dataset upload. Users can only retrieve the dataset if the verification succeeds, which prevents unauthorized access to encrypted healthcare records.

The verification process can be expressed as

$$V(K_i, d) = 1, \text{ if } K_i, d = K_{\text{stored}}; 0, \text{ otherwise} \quad (19)$$

where K_i, d represents the private key entered by the user, K_{stored} denotes the securely stored private key, $V(K_i, d)=1$ indicates successful verification, and $V(K_i, d)=0$ indicates verification failure.

Only after successful verification is the encrypted healthcare dataset decrypted with the Optimized ECC decryption module and made available for download. If the verification fails, the system rejects the request and shows an authentication failure message. This verification mechanism stops unauthorized users from accessing protected healthcare records while ensuring secure data retrieval. By combining key verification with Role-Based Access Control (RBAC), blockchain-based secure storage, and Optimized ECC encryption, the proposed framework adds another layer of security for managing healthcare insurance data [9], [14].

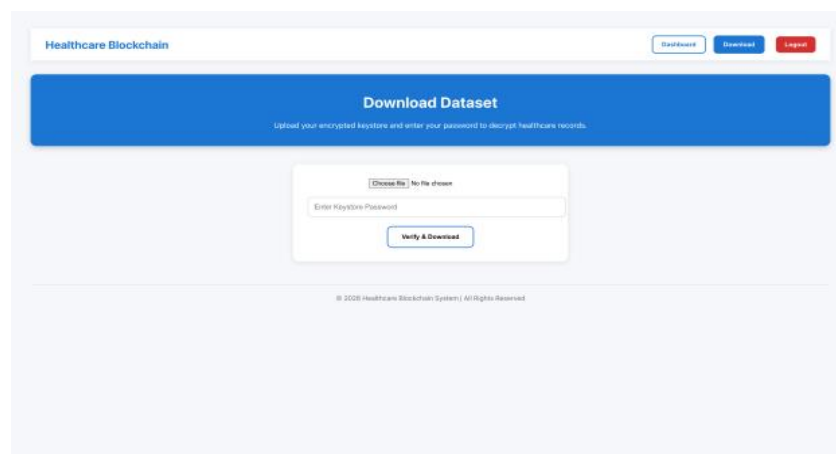


Fig. 3. Key verification interface for secure healthcare dataset retrieval.

2.8 Real-Time Tampering Detection

The proposed healthcare insurance management system includes a Real-Time Tampering Detection mechanism to monitor the integrity of blockchain data continuously. The system checks each blockchain block regularly by recalculating its hash value and comparing it with the hash stored earlier. This process allows for the quick detection of unauthorized changes and guarantees the integrity of healthcare records stored on the blockchain.

The blockchain verification process is represented as

$$V_i = 1, \text{ if } Hash_i = SHA256(Block_i); 0, \text{ otherwise} \quad (20)$$

where $Hash_i$ represents the stored hash value of the i -th blockchain block, $SHA256(Block_i)$ shows the newly computed hash value. $V_i=1$ means that block integrity is preserved, while $V_i=0$ indicates that the blockchain block has been tampered with.

Whenever a hash mismatch is found, the blockchain status on the healthcare dashboard automatically updates from “Secure” to “Tampered.” At the same time, the tampering event counter increases, allowing authorized users to track security

violations in real time. This automated monitoring system lets users quickly identify harmful changes without needing manual checks. The real-time monitoring process happens through regular communication between the React frontend and the Flask backend. The dashboard retrieves the latest blockchain status consistently, ensuring that any unauthorized changes are immediately shown in the user interface. This feature greatly improves the reliability and trust of the proposed healthcare insurance management system [8], [15]. By combining automated blockchain verification with real-time dashboard updates, the proposed framework offers continuous integrity monitoring, quick tampering detection, and better security for healthcare insurance records. Fig. 4 shows the healthcare dashboard, which monitors blockchain integrity by displaying the total number of blocks, total healthcare records, blockchain status, and tampering events in real time.



Fig. 4. Healthcare dashboard displaying blockchain status, total records, and tampering events in real time.

III. IMPLEMENTATION

3.1 Healthcare Dataset

To implement and evaluate the proposed healthcare insurance privacy preservation framework, we used a publicly available healthcare insurance dataset. This dataset contains 1,338 records with seven attributes: age, sex, body mass index (BMI), number of children, smoking status, region, and medical insurance charges. These attributes offer demographic and insurance-related information commonly used in healthcare analytics and provide a solid foundation for testing secure data management methods.

Before encryption, we examined the dataset to confirm that the records were suitable for processing within the proposed framework. Each healthcare record was organized into a structured format and inputted into the optimized Elliptic Curve Cryptography (ECC) module. After encryption, we stored the protected records in the blockchain, ensuring their integrity and confidentiality during both storage and retrieval. We used the selected dataset to validate the framework's functionality, including secure data encryption, blockchain-based storage, role-based access control, and blockchain integrity verification. While the experimental evaluation relied on this dataset, the framework is designed to support other healthcare insurance datasets with a similar structure after appropriate data preparation.

age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.47061
32	male	28.88	0	no	northwest	3866.8552
31	female	25.74	0	no	southeast	3756.6216
46	female	33.44	1	no	southeast	8240.5896
37	female	27.74	3	no	northwest	7281.5056
37	male	29.63	2	no	northeast	6406.4107
60	female	25.84	0	no	northwest	28923.13692
25	male	26.22	0	no	northeast	2721.3208
62	female	26.29	0	yes	southeast	27808.7251
23	male	34.4	0	no	southwest	1826.843
56	female	39.82	0	no	southeast	11090.7178
27	male	42.13	0	yes	southeast	39611.7577
19	male	24.6	1	no	southwest	1837.237
52	female	30.78	1	no	northeast	10797.3362
23	male	23.845	0	no	northeast	2395.17155
56	male	40.3	0	no	southwest	10602.385

Fig. 5. Representative sample of the healthcare insurance dataset used for implementation.

3.2 Optimized ECC Implementation

The optimized Elliptic Curve Cryptography (ECC) module was set up to secure healthcare insurance records before they are stored in the blockchain. During the data upload, each healthcare record gets processed individually to create its encrypted version. The encryption module generates a cryptographic key, encrypts the healthcare record, and provides the encrypted output needed for blockchain storage. The generated cryptographic information is stored securely to support authorized data retrieval. By encrypting the records before inserting them into the blockchain, this method ensures that only protected healthcare information is saved. This process reduces the risk of unauthorized disclosure during storage and data sharing [1], [3].

Algorithm 3: Optimized ECC Encryption Process

Input: Healthcare Dataset D

Output: Encrypted Healthcare Records (EHR)

1. Read the uploaded healthcare dataset D.
2. Load the BrainpoolP256r1 elliptic curve parameters.
3. Generate the ECC private key $d \in [1, n - 1]$ (Eq. 3)
4. Compute the corresponding public key $Q = dG$ (Eq. 4)
5. For each healthcare record $R_i \in D$ do
6. Generate a random ephemeral key k .
7. Compute the shared secret $S = kQ$ (Eq. 5)
8. Generate the first ciphertext component $C_1 = kG$ (Eq. 6)
9. For every character M_i in R_i do
10. Encrypt the character using $C_i = M_i \oplus S_x$ (Eq. 7)
11. End For
12. Construct the encrypted healthcare record $EHR = (C_1, C_2, C_3, \dots, C_n)$ (Eq. 8)
13. Forward EHR to the blockchain module.
14. End For
15. Return all encrypted healthcare records.

3.3 Blockchain Implementation

The blockchain module offers secure storage for the encrypted healthcare insurance records. After encryption, each record is contained within a blockchain block that holds the encrypted data and its cryptographic metadata. Every new block links to the previous one through its hash value before being added to the blockchain. This system keeps the chronological order of records and ensures data integrity. The completed blockchain is available for secure retrieval and integrity checks during authorized access [10], [12].

Algorithm 4: Blockchain Block Creation and Secure Storage

Input: Encrypted Healthcare Records (EHR)

Output: Immutable Blockchain

1. Initialize the blockchain.
2. If the blockchain is empty then
3. Create the Genesis Block.
4. End If
5. For each encrypted healthcare record EHR do
6. Construct a blockchain block $Block = \{Index, Timestamp, Data, PreviousHash, Hash\}$ (Eq. 11)
7. Compute the current block hash using SHA-256 $Hash = SHA-256(Block\ Contents)$ (Eq. 12)
8. Store the hash of the previous block $PreviousHash_i = Hash_{i-1}$ (Eq. 13)
9. Link the current block with its predecessor.
10. Append the validated block to the blockchain.
11. End For
12. Store the updated blockchain.
13. Recalculate blockchain hashes during verification.
14. If any computed hash differs from the stored hash then
15. Update blockchain status as Tampered.
16. Else
17. Maintain blockchain status as Secure.
18. End If
19. Return the immutable blockchain.

3.4 Doctor Module

The Doctor Module lets authenticated doctors securely upload healthcare insurance datasets via a React.js interface connected to a Flask backend. Access is managed by a Role-Based Access Control (RBAC) system. Once authenticated, the uploaded CSV dataset is validated and sent to the Optimized Elliptic Curve Cryptography (ECC) module, where the healthcare records are encrypted. These encrypted records are stored in the blockchain, with each block containing the encrypted data, timestamp, previous hash, and SHA-256-generated hash. After successful storage, the system automatically creates and downloads the secure dataset key needed for future data retrieval. Finally, the healthcare dashboard updates with blockchain statistics, including the number of records, blockchain blocks, blockchain status, and tampering information. This setup ensures secure healthcare data uploads while maintaining confidentiality, integrity, and blockchain-based storage.

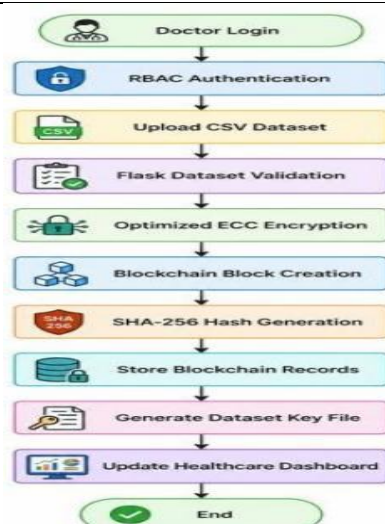


Fig. 6. Implementation workflow of the Doctor Module for secure healthcare dataset upload using Optimized ECC and blockchain storage.

3.5 Insurance Provider Module

The Insurance Provider Module allows authorized insurance organizations to securely retrieve healthcare insurance records from the blockchain. Unlike the Doctor Module, this version only provides read-only access and does not allow uploading datasets or changing the blockchain. Access to the module is limited by authentication and secure key verification. After logging into the system, the insurance provider uploads the secure key file created during the dataset upload process. The Flask backend checks the uploaded key information and validates its authenticity before granting access to the encrypted healthcare records stored in the blockchain. If the verification is successful, the corresponding encrypted records are retrieved from the blockchain and sent to the Optimized ECC decryption module. The decryption module rebuilds the original healthcare dataset using the verified cryptographic information. Once decryption is successful, the recovered dataset is generated in CSV format and made available for secure download through the web interface. If key verification fails, the download request is denied, preventing unauthorized access to confidential healthcare information. This setup combines authentication, secure key verification, blockchain retrieval, Optimized ECC decryption, and controlled dataset download into a single retrieval mechanism. This ensures that healthcare insurance records stay accessible only to authorized insurance providers [9], [14].

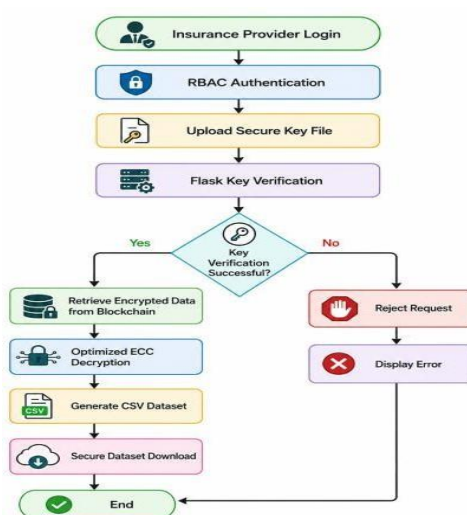


Fig. 7. Implementation workflow of the Insurance Provider Module for secure retrieval and decryption of healthcare records from the blockchain.

3.6 Blockchain Integrity Verification

The Blockchain Integrity Verification module was set up to monitor the integrity of the blockchain after healthcare records are saved. The module regularly retrieves blockchain blocks, recalculates their SHA-256 hash values, and compares them to the stored hash values. It also checks the previous hash connection between consecutive blocks to ensure blockchain consistency. If there is any hash mismatch or invalid block connection, the affected block is marked as tampered. The healthcare dashboard is then updated, changing the blockchain status from Secure to Tampered and increasing the tampering

event counter. If everything matches, the blockchain status stays Secure. The verification process connects with the Flask backend, while the React dashboard regularly retrieves the latest blockchain status through API requests. This setup allows for continuous blockchain monitoring and quick detection of unauthorized changes, improving the reliability and security of the healthcare insurance management system [8], [11].

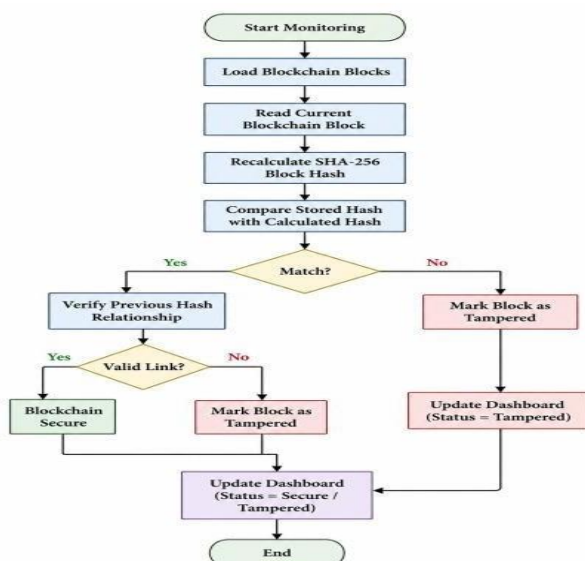


Fig. 8. Implementation workflow of the Blockchain Integrity Verification module for continuous monitoring and tamper detection of blockchain records.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

4.1 Comparative Performance Analysis

The computational performance of the proposed healthcare insurance management system was evaluated by comparing it with the existing SI-COA + M-HAS + Improved ECC framework [1]. The comparison included execution time, key generation time, CPU utilisation and scalability. The experimental results from the implemented system are presented and discussed in the following subsections.

4.1.1 Total Pipeline Execution Time

The total pipeline execution time reflects the overall time needed to complete healthcare data processing. This includes key preparation, encryption, blockchain storage, and related processing operations.

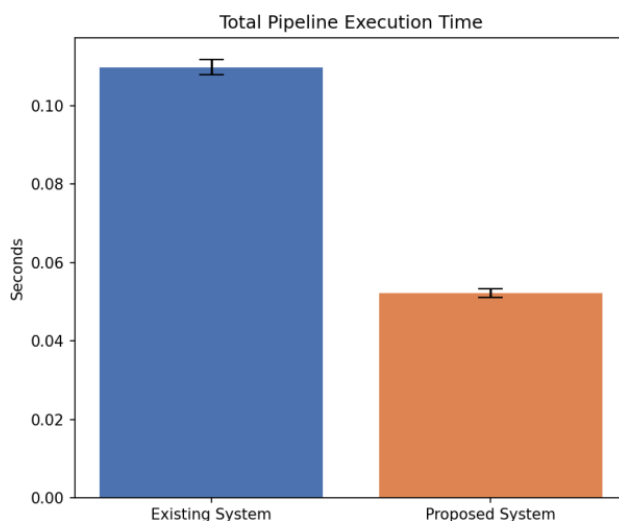


Fig. 9. Comparison of total pipeline execution time between the existing system and the proposed system.

As shown in Fig. 9, the proposed system finishes the healthcare data processing pipeline faster than the existing framework [1]. The current system carries out extra processing through the SI-COA optimization algorithm and the M-HAS deep learning module before encryption, which adds to the computational burden. In contrast, the proposed system directly applies the Optimized ECC module for encryption and blockchain storage, simplifying the processing pipeline. As a result,

the proposed framework achieves better computational efficiency while keeping healthcare data secure.

4.1.2 Key Generation Time

The key generation time indicates how long it takes to produce the cryptographic keys before starting the encryption process. This metric is significant since it influences the overall efficiency of secure healthcare data processing.

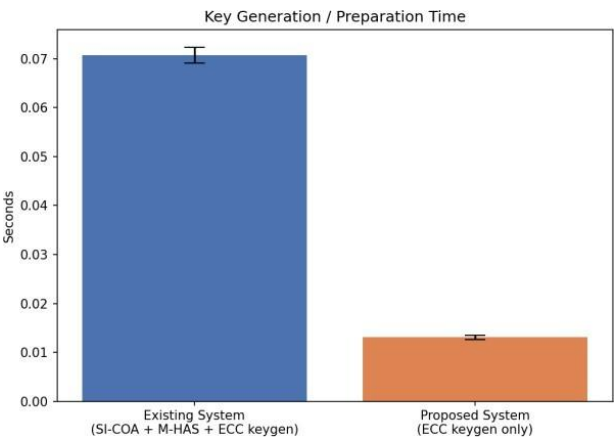


Fig. 10. Comparison of key generation time between the existing system and the proposed system.

As illustrated in Fig. 10, the proposed system takes less time to generate keys than the existing framework [1]. The existing system goes through additional preprocessing with the SI-COA optimization algorithm and the M-HAS module before generating ECC keys, which increases preparation time. In contrast, the proposed system generates the ECC key pair directly without any extra optimization steps, thus cutting down computational overhead. The shorter key generation time leads to faster execution of the entire healthcare data protection workflow while ensuring secure cryptographic processing.

4.1.3 CPU Utilisation

CPU utilization represents the average processor usage during healthcare data processing. This metric examines the computational load that the proposed framework imposes while executing encryption, blockchain storage, and related processing tasks.

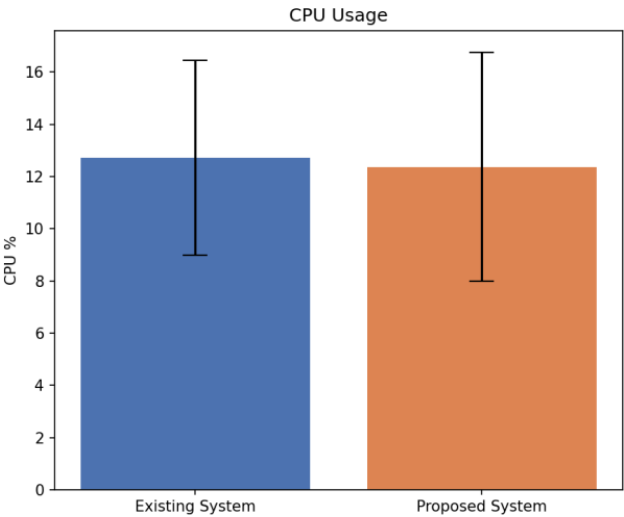


Fig. 11. Comparison of CPU utilization between the existing system and the proposed system.

As shown in Fig. 11, the proposed system exhibits slightly lower CPU utilization than the existing framework [1]. This decrease in processor usage mainly stems from removing the SI-COA optimization algorithm and the M-HAS deep learning module from the processing pipeline. By directly using the Optimized ECC implementation, the proposed framework lowers computational overhead while ensuring secure encryption and blockchain storage. These findings suggest that the proposed system uses processor resources effectively without compromising security or functionality in healthcare data management.

4.1.4 Scalability Analysis

A scalability analysis was carried out to assess how the proposed framework performs as healthcare dataset sizes increase. The total processing time was recorded for various numbers of healthcare records to see how well both the existing

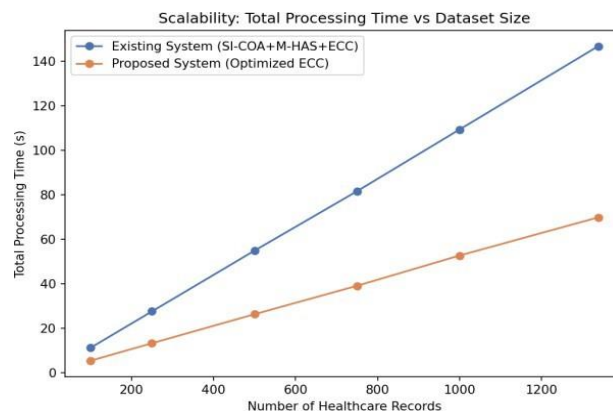


Fig. 14. Comparison of total processing time for different healthcare dataset sizes in the existing and proposed systems.

As shown in Fig. 14, processing time rises with the size of the healthcare dataset for both systems. However, the proposed framework consistently requires less processing time than the existing system across all dataset sizes [1]. The gap in performance becomes more apparent as the number of healthcare records grows, meaning the proposed implementation scales more effectively with larger workloads. The enhanced scalability results from eliminating the computational burden associated with the SI-COA optimization algorithm and the M-HAS deep learning module. By directly using the Optimized ECC for encryption and blockchain storage, the proposed framework maintains efficient processing while lowering overall execution time. These results prove that the proposed system is better equipped for secure healthcare data management involving large-scale datasets.

4.2 Security Analysis

The proposed healthcare insurance management system includes several security measures to ensure the confidentiality, integrity, and controlled access to healthcare records. First, healthcare datasets are encrypted using the Optimized Elliptic Curve Cryptography (ECC) algorithm before they are stored in the blockchain. This protects sensitive information from unauthorized access [1], [3]. The encrypted records are then organized into blockchain blocks, with each block linked to its predecessor using SHA-256 cryptographic hash values. This linkage helps maintain blockchain immutability and allows for the detection of unauthorized changes [10], [15].

To enhance system security, the Role-Based Access Control (RBAC) mechanism restricts system operations based on user roles [9], [12]. Authorized doctors can upload healthcare datasets, while insurance providers can retrieve healthcare records only after successful authentication and private key verification. Additionally, the Blockchain Integrity Verification module continuously checks blockchain blocks by recalculating and comparing hash values. Any inconsistencies detected update the blockchain status and reveal potential tampering attempts [8], [14]. These security measures together provide a secure and reliable framework for managing healthcare insurance data.

4.3 Discussion

The experimental results show that the proposed healthcare insurance management system improves efficiency and securely manages healthcare data. Compared to the existing SI-COA + M-HAS + Improved ECC framework [1], this new implementation cuts down the total pipeline execution time and significantly reduces key generation time by removing unnecessary optimization steps. The encryption and decryption processes perform at similar levels, and CPU usage stays consistent. This indicates that the proposed framework does not add extra computational load. Additionally, the scalability analysis reveals that the proposed system requires less processing time as the healthcare dataset size grows, proving it can handle large-scale healthcare data management. Along with Optimized ECC encryption, blockchain-based storage, Role-Based Access Control, private key verification, and real-time tampering detection, the proposed framework offers an efficient and secure way to protect healthcare insurance records.

V. CONCLUSION AND FUTURE WORK

This paper presented a blockchain-based privacy preservation framework for secure healthcare insurance data management using Optimized Elliptic Curve Cryptography (ECC). The framework combines Optimized ECC encryption, secure blockchain storage, SHA-256 hashing, Role-Based Access Control, private key verification, and blockchain integrity monitoring [1], [2], [10]. The system uses a React.js frontend and a Flask backend, creating a secure and efficient platform for managing healthcare data. Experimental evaluation showed that the implementation reduces overall pipeline execution time and key generation overhead compared to the existing SI-COA + M-HAS + Improved ECC framework [1]. The system also maintained similar encryption, decryption, and CPU usage while showing better scalability as the healthcare dataset size increased. The security analysis confirmed that using Optimized ECC, blockchain immutability, cryptographic hashing, and real-time tampering detection offers reliable protection for healthcare insurance records [2], [15].

Future work will focus on expanding the framework to handle larger healthcare datasets. It will integrate cloud-based blockchain deployment, smart contracts, decentralized identity management, and new cryptographic techniques to improve scalability, security, and interoperability in real-world healthcare settings [4], [6], [7].

References

- [1]. J. Buvana and R. Gayathri, "Blockchain-driven privacy-preservation of healthcare insurance data using improved ECC based encryption and deep learning based key tuning," *Knowledge-Based Systems*, vol. 327, Art. no. 114078, Oct. 2025. DOI: 10.1016/j.knsys.2025.114078.
- [2]. A. Ali, B. A. S. Al-rimy, F. S. Alsubaei, and A. A. Almazroi, "HealthLock: Blockchain-Based Privacy Preservation Using Homomorphic Encryption in Internet of Things Healthcare Applications," *Sensors*, vol. 23, no. 15, Art. 6762, Jul. 2023.
- [3]. S. Preethi and C. Priyadharsini, "Deep Learning with Blockchain Technology for Secure Data Management in Healthcare Sector Using Hybrid Elliptic Curve- RSA Cryptography," 2022.
- [4]. J. A. Alzubi, O. A. Alzubi, A. Singh, and M. Ramachandran, "Cloud-IIoT-Based Electronic Health Record Privacy-Preserving by CNN and Blockchain-Enabled Federated Learning," 2022.
- [5]. A. Ali, H. Ali, A. Saeed, A. A. Khan, and H. G. Mohamed, "Blockchain-Powered Healthcare Systems: Enhancing Scalability and Security with Hybrid Deep Learning," 2023.
- [6]. K. Mahesh Babu, M. Syed, S. Shaik, and S. Thalari, "Fully Homomorphic Encryption Framework for Privacy Preserving in Healthcare through Decentralized Machine Learning," 2024.
- [7]. E. Afrihyia, E. C. Chianumba, A. Y. Mustapha, and A. Y. Forkuo, "Developing Privacy-Preserving Data Sharing Protocols for Healthcare Systems Using Cryptographic and Blockchain-Based Techniques," 2025.
- [8]. G. Rama Devi, "Enhanced Privacy for Health Records Using Blockchain Technologies," 2024.
- [9]. M. B. Singh and A. Pratap, "BPFISH: Blockchain and Privacy-Preserving Federated Learning Inspired Smart Healthcare," 2022.
- [10]. T. T. Kuo, H. E. Kim, and L. Ohno-Machado, "Blockchain Distributed Ledger Technologies for Biomedical and Health Care Applications," *Journal of the American Medical Informatics Association*, 2022.
- [11]. A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," *Proc. IEEE Open & Big Data Conference*, 2023.
- [12]. Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments," *Information*, vol. 8, no. 2, 2023.
- [13]. A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and Trustable Electronic Medical Records Sharing Using Blockchain," *AMIA Annual Symposium Proceedings*, 2024.
- [14]. Y. Zhang, X. Liu, and H. Chen, "Privacy-Preserving Healthcare Data Sharing Using Blockchain and Attribute-Based Encryption," *IEEE Access*, 2021. Principal results from EUROASPIRE II. *Eur Heart J* 2001,22:554-572.
- [15]. M. J. Mettler, "Blockchain Technology in Healthcare: The Revolution Starts Here," *Proc. IEEE Healthcare Innovations and Point-of-Care Technologies*.