ISSN: 0067-2904

Secure Electronic Healthcare Records (SEHR) Using Blockchain Model based on Elliptic Curve Cryptography (ECC)

Ali H. Kashmar*[1], Ammar I. Shihab[2]
[1]*Department of Cyber Security, College of Science, Al-Farabi University, Baghdad, Iraq.*
[2]*Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.*

Received: 30/ 10/2024 Accepted: 5/ 10 /2025 Published: 30/ 8/2026

Abstract

The paper presents the implementation of a system that could block-chain electronic health records (EHR), meeting both the demands of users needing rapid access to information and the privacy-security requirements. This work describes a novel blockchain-based EHR management framework to achieve effective immediate user access with strict requirements of privacy and security, making it different from the current digital EHR management practice. The methodology comprises a design framework of five entities, which includes the Registration Centre (RC), Patients, Doctors, Mobile Device, and Blockchain (BC). Integration of Elliptic Curve Cryptography (ECC) is for data security enhancement, whereas public-based smart contracts facilitate both fast and secure transactions. Management of access control is handled via MetaMask wallets, permitting authorized users like doctors and patients to create, retrieve, and update records while safeguarding their privacy. The system was subjected to evaluation via simulation of transaction workflows and security assessments to show the improvements in data access times and increased trust and robust privacy safeguards. All the results indicate that the proposed framework can effectively ensure, to some extent, interoperability, auditability, and security, thereby making it an efficient solution for health data management.

Keywords: Blockchain, Elliptic Curve Cryptography (ECC), Electronic Healthcare Records (EHR).

السجلات الصحية الإلكترونية الآمنة (SEHR) باستخدام نموذج Blockchain القائم على تشفير المنحنى الإهليلجي (ECC)

علي حبيب كشمر[1], عمار ابراهيم شهاب[2]
[1] قسم علوم الامن السيبراني , كلية العلوم , جامعة الفارابي, بغداد, العراق.
[2] قسم علوم الحاسوب, كلية العلوم, جامعة بغداد, بغداد, العراق.

الخلاصة

سلسلة الكتل نظام قائم على تقنية البلوك تشين لإدارة السجلات الصحية الإلكترونية. في موازنة توقعات المستهلكين بالوصول الفوري إلى بياناتهم الصحية مع الخصوصية والأمان. نقدم إطار عمل يتكون من خمس كيانات – مركز التسجيل (RC)، المرضى، الأطباء، الأجهزة المحمولة، والبلوك تشين (BC) ونؤكد على أنه

*Email: ali.habib@alfarabiuc.edu.iq

Open Access Articles. This articles is licensed under

يجب استخدام تشفير المنحنى الإهليلجي (ECC) لتعزيز الأمان. يستخدم نظامنا سلسلة كتل عامة لتخزين بيانات المرضى بأمان وتشغيل الأشياء باستخدام العقود الذكية للمعاملات السريعة والآمنة. يتيح الوصول من خلال محافظ MetaMask داخل المنصة للمستخدمين المصرح لهم الوصول إليها وإدارتها لاستردادها، على سبيل المثال الطبيب والمريض للحفاظ على الخصوصية أثناء الاحتفاظ بالسجلات الطبية. يسعى مقترح البحث إلى تعزيز الكفاءة والثقة والأمان في تخزين البيانات الصحية ونقلها بالإضافة الى التوازن والتدقيق على الرغم من أن الحل المقترح يعمل للجميع، إلا أنه لا يزال يحقق نفس قابلية التدقيق للتشغيل البيني.

1. Introduction

In a nutshell, blockchain is a distributed database or series of transactions and events in digital form that one can access and view by all participants, transparently sharing info from a technical perspective: [1] Blockchain again is generally seen as a shared database wherein all participants have access to the record of transactions and digital events carried on and shared between themselves. Its add-on value is in increasing all transparency and efficiency by removing intermediaries such as banks and government authorities from the process of transaction. One of these important benefits is that they will then be able to process transactions as peer-to-peer rather than going through intermediaries such as banks and governments, which enhances better transparency and efficiency [2]. However, MOBI is exploring the potential use of blockchain in the mobility domain, including aspects like secure vehicle identities, smart contracts, data monetization, and better coordination of autonomous vehicles. The vision of empowering the mobility ecosystem through enhanced security, efficiency, and sustainability using an army of leading automotive manufacturers and other diverse collaborators to promote innovation in the industry [3]. The three essential components of Blockchain technology are illustrated in Figure 1. Data Storage: A block contains information about transactions, such as the sender and recipient addresses and amounts. Unique Hash: Each block has a distinct hash correlated with data contained within it-that hash changes if any data has been modified. Chained Blocks: Each block includes the hash value of its predecessor block, forming a secure chain that requires all subsequent blocks to update in case any block is altered, which guarantees integrity and security within the system [4].

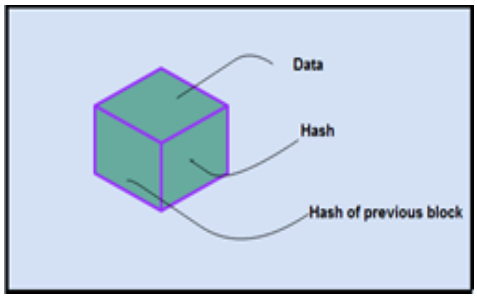


Figure 1: The three essential components of blockchain technology [4].

Cryptography is the study of secure communication methods that safeguard against threats. It utilizes encryption algorithms to convert data into a secure format, which can be decrypted using a secret key. There are two primary encryption types: symmetric encryption, which involves a single key for both encryption and decryption, necessitating secure sharing of this key between the sender and recipient. A blockchain framework combined with elliptic curve cryptography (ECC) to improve security and privacy in medical data transmission was introduced. It leverages blockchain's decentralization for data integrity and transparency, while ECC enables efficient encryption and decryption, addressing scalability and performance issues and facilitating secure sharing among healthcare stakeholders. [5]. Elliptic Curve ECC — Elliptic Curve Cryptography — is an asymmetric encryption method based on the math of two elliptic curves defined over finite fields. It provides public-key encryption, similarly to RSA,

but reaches an equivalent. This results in security with far smaller key sizes since it allows elliptic curve math (key sizes as small as 256 bits vs. the 2048-bit for a large prime) [6]. Elliptic ECC (Curve Cryptography) makes the public key cryptosystem more secure with hybrid (encrypt and decrypt) and not only us. Digital signatures, though the security relies on challenging mathematics, specifically the factorization between points of the elliptic curve, where Y is equal to a scalar multiplication of point G (Y =kG) [7]. The components of Elliptic Curve Cryptography (ECC) consist of ECC keys and a generator point. ECC keys, which include: private key, a quick and simple random integer generated within a specific range, valid as any integer within the field. Public key: grated from points on the elliptic curve represented by integer coordinates (x, y), which can be compressed into a single coordinate plus an additional bit, resulting in a 256-bit compressed public key [8]. Generator point: in ECC, the generator point G is a predefined point on elliptic curves over finite fields. It could generate any point in its subgroup by multiplying G by an integer ranging from 0 to r, with r being the order of the subgroup. While there are various generator points in elliptic curve subgroups, one chosen for its calculative convenience is designated as "G." Elliptic Curve Cryptography (ECC) sets out to offer many tools, much like key exchange using ECDH, message authentication and integrity using ECDSA, and well publicized secure encryption using ECIES, together with elliptic curve operations over finite fields. Besides these, ECC comes with the less-traditional support for special hash functions and identity-based encryption, reducing key management burdens by deriving public keys from user identities. This work intends to secure and optimize Electronic Healthcare Records (EHR) under a blockchain model using the ECC algorithm, a paradigm that therefore engenders superior blockchain security and reliability.

Traditional Electronic Health Record (EHR) storing systems are often based on centralized architecture that provides great challenges in single points of failure, privacy threats, and data breach opportunities. Data stored within such centralized repositories becomes at nasty risk and may invite attempts of a cyberattack that ultimately turns costly and harmfully into breaches that seek to violate patient confidentiality, if not trust. Toward finding a remedy, elliptic curve cryptography (ECC) seems more appropriate than RSA when it comes to securing healthcare data since they provide nearly the same level of security, the advantage of small key sizes with improvement in efficiency, computational overhead, and a better environment for resource constraints. Even though extensive research is being conducted on solutions in cryptography, a significant gap exists in the literature on implementing ECC in healthcare data management, specifically for balancing high security with robust system performance while respecting patient privacy. This paper intends to fill this gap by giving an extensive analysis of ECC's application to the healthcare area and providing optimized frameworks for the resolution of existing security vulnerabilities within the conventional EHR storage systems. The EHR system's principal challenges involve a lack of accessibility and a security trade-off, centralized data vulnerabilities, little interoperability, and significantly diminished patient control over data and poor security. The blockchain framework aims to alleviate these challenges by ensuring timely access to EHRs in a secure and private manner, thus addressing limitations in existing methodologies.

The layout of the paper is as follows: Section 2 will present some related work, Section 3 will explain the methods and methodology that cover the details of the proposed algorithm, while Section 4 will be a discussion of the experimental outcomes and security analysis of the proposed algorithm. Sections 5 and 6 will deal with the conclusions and future works of the paper, respectively.

2. Related Work

Overall, these algorithms, which are discussed in Section 1, improve upon conventional cryptography techniques [10] in terms of security and efficiency. Bitcoin employs elliptic curve cryptography (ECC) for the purpose of securing its transactions and creating new units. ECC produces public and private key pairs for ownership verification and transfer validation, hence improving the security and efficiency of this method. Furthermore, this cryptographic technique makes Bitcoin transactions trustworthy and fraud-proof [11].

Elliptic Curve Cryptography (ECC) has several advantages including quick key generation using random integers, smaller key sizes that are easier to manage but offer equivalent security, faster transaction speeds due to a two-stage signature process, less power consumption in limited-resource devices, and above all, a single small key provides considerable security: a 256-bit ECC key provides the same protection as a 3072-bit RSA key [12]. A private decentralized e-healthcare system was presented that combines Hyperledger Fabric with IPFS for securely storing and accessing electronic health records (EHRs). Hyperledger Fabric ensures authorized access to sensitive data, while IPFS provides decentralized storage for large medical images and files that cannot be stored directly on the blockchain [13]. ECC key basics comprise of two elements, the ECC key and the generator point. ECC keys, which are: private key: A random integer, generally, a small and simple generated number between a specific range, valid for that info character transactions [14]. Two EEG data compression methods were studied and introduced. The first uses wavelet transform and adaptive encoding, while the second employs delta modulation and Huffman encoding. Both effectively reduce file size while preserving data integrity, with the second method being more efficient for large datasets. [15]. The Ethereum blockchain operates with the cryptocurrency ETH, enabling a decentralized community for executing and verifying clever contracts. This capability allows builders to create ETH-based totally tokens that aid decentralized programs (dapps) and help transactions without a government. Smart contracts, which are automatic structures runningunder particular conditions, offer logical operations for the blockchain and outline peer-to-peer agreement phrases [16]. Smart contracts on the Ethereum blockchain are used to facilitate the transfer of facts, permitting scientific professionals to get access to patient statistics at the same time as proscribing access for some healthcare employees [17]. The healthcare enterprise encounters difficulties in dealing with huge volumes of records from hospitals. Smart contracts can protect all sensitive issues from unauthorized access and minimize the possibility of hackers breaking into this information. The technology of blockchain provides various solutions to several issues in the health sector, such as decentralization, security, privacy, and robustness, considering cryptography. Digital healthcare services address the storage of records, but challenges such as patient data disappearing or being leaked or records being accessed without authorization continue to plague and hinder innovation in the sector. Therefore, the focus of the proposed system is basically to ensure a secure provision of information within the healthcare community on a well-devised blockchain.

3. Method and Methodology

This segment discusses the proposed frameworks, including the software program platform, in connection with its advantages. Major additives of these frameworks are Ethereum and Interplanetary File System (IPFS), both of which are instrumental in their operation. In this project, IPFS serves as the decentralized storage for enormous healthcare data files. Instead of storing heavy medical records on the blockchain directly, which would be inefficient and costly, the system stores encrypted health data on the IPFS, providing secure, distributed, and tamper-proof storage. Whereas the blockchain, through smart contracts, will keep only the cryptographic hashes of these IPFS-stored files, establishing a verifiable link between data and its reference. This arrangement achieves efficient access and retrieval of huge medical records

while guaranteeing data integrity and privacy. Therefore, overall, IPFS increases the system scalability, reduces the costs of storage on the blockchain, and ensures a resilient and decentralized infrastructure for healthcare data management.

3.1 Healthcare Scenario

The proposed healthcare device entails patients and doctors and employs blockchain to enhance the reliability and compatibility of healthcare structures. It emphasizes the significance of cryptographic security. Both sufferers and doctors can get admission to scientific statistics, which include records, insurance, lab results, and prescriptions, the usage of a secure portal with a key, with the patient's consent. The machine also introduces a cryptographic method for the stable updating and enhancement of patient scientific information inside the blockchain. Figure 2 presents an in depth evaluation of the entities inside the proposed device, outlining their workflows and related activities

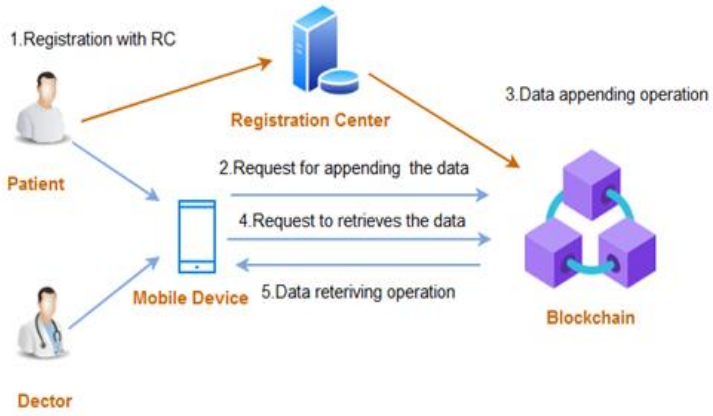


Figure 2: illustrates the system model for the proposed design.

The proposed medical system comprises five key entities: the Registration Centre (RC), the Patient, the Doctor, the Mobile Device, and the Blockchain (BC). Each entity is essential to the overall functioning of the healthcare system, as outlined in the description. The proposed healthcare system enables patients to access information about hospitals and doctors through apps. It requires doctors to be knowledgeable about patients' medical histories and proficient in mobile technology. Patients should designate a trusted individual to handle their reports in emergencies. Patients should designate a trusted individual, such as a family member or caregiver, to handle their reports and healthcare information in emergencies. This ensures that critical health data can be accessed and managed promptly when the patient is unable to do so themselves.

Additionally, both patients and doctors must be aware of security measures, such as password protection and biometric authentication, when using the app.

3.2 The ways in which certain entities function

1. To access the healthcare app, individuals must register at the Registration Center by submitting biometric signatures and IDs. This process is only needed for new apps. Upon completion, patients receive access keys without engaging with the Blockchain. After downloading the app, they can use it by following the instructions provided.
2. The software for the patient's mobile device requires biometric authentication for installation. After installation, the app confirms the keys for both the patient and the doctor. The registration process also requires the patient's identity and a pair of secret keys (private and public) to be completed.

3. The patient must authorize the doctor to access their information, which allows both the patient's and doctor's public keys to be added to the Blockchain. During the registration process, patients may choose to disclose their doctor's identity and details about others, such as friends or relatives. Consequently, a public key and the patient's information will be shared with the doctors, and all patient data is securely recorded on the blockchain.
4. Patients can see a doctor for routine or initial visits. With the patient's consent and biometric data, records can be updated. Both the patient and doctor must authenticate at the same time, using the Schnorr Scheme for verification. If their public keys match, the process continues; otherwise, the request is discarded.
5. Blockchain provides patient data to doctors after a valid request, allowing doctors to update the information. Patients can view these updates to their medical report, but are not permitted to add or edit any details themselves.
6. The page discusses the sharing and retrieval of patient information within a blockchain system. Patients can review their medical details, while doctors have the ability to add and access data. Additionally, patients can present their reports at pharmacies for medication, and lab technicians can access the information for conducting recommended tests. Figure 2.1 visually represents the interaction between different entities and the blockchain framework.

3.3 System Model

Blockchain technology offers an effective solution in healthcare for patients and medical professionals by utilizing cryptographic methods. It addresses the issues faced in traditional healthcare systems, significantly improving data security and privacy for sensitive medical information [19]. Elliptic Curve Cryptography (ECC) enhances cryptography in the proposed healthcare system by being more efficient and easier to implement than traditional algorithms like RSA. It employs one-way hash functions for encryption and decryption and incorporates methods from ECHLP and ECDHP. The ECC technique is a plane curve defined by an equation of the following form, based on specific conditions for parameters (a) and (b). Where G: generator point (fixed constant, a base point on the EC), k: private key (integer), and P: public key (point)

$$y^2 = x^3 + ax + b \quad \ldots (3.1)$$

It can be shown that a group can be defined based on the set $E(a, b)$ for specific values of a and b
in Equation (3.1), provided the following condition is met:

$$D = 4a^3 + 27b^2 \neq 0 \quad \ldots(3.2)$$

Additionally, using ECC multiplication as: for some scalar (integer) r and a point $P = (x, y)$ that lies on the curve,

$$R = rP \quad \ldots(3.3)$$

Also adding:

$$R1 + R2 \quad \ldots(3.4)$$

The ECDLP and ECDHP act in accordance with the values: q a prime number

$$Q = xR \quad \ldots(3.5)$$

The security of modern ECC depends on the intractability of determining n from $Q = rP$ given known values of Q and P if r is large

$$\text{and} \quad R = xP, Q = yP \quad \ldots(3.6)$$

Encryption is then followed by:

$$C = E_k(M) \quad \ldots (3.7)$$

As well as decryption:

$$M = D_k(C) \quad \ldots(3.8)$$

The proposed approach utilizes cryptographic processes to enhance readability and clarity. It consists of five phases, illustrated in Figure 3.1:

1. **Data Registration Phase**
2. **Data Request Appendix**
3. **Data Append Phase**
4. **Request to Retrieve Data**
5. **Data Retrieve Phase**

Each phase serves a distinct function within the described healthcare scenario.

3.3.1 Phase of Data Registration

The data registration step outlines how patients and doctors can register on the blockchain network using a smartphone app. Both individuals are encouraged to bring their phones to the hospital to streamline the process. Patients initiate the registration, enabling them to share information with their doctor, who is also registered in the network. In order to complete a single registration, both the patient and the physician must utilize a mobile device if possible quickly access to by providing their personal identities with the patient identities id_p , Public key patient (pk_p) , Public key doctor (pk_d), doctor identities (id_d), and id_p attending to the sufferer (patient) identities. In addition to both identities, patients can also add dependable family members or acquaintances. To prevent unneeded alterations, the patients. In this case, each person is attended to separately. The registration stage then sends $(id_p, pk_p pk_d)$ to the BC that the patient's physician both signed, all required information (id_s, pk_d) was previously sent to BC when you registered. The patient's and doctor's signatures are also kept on the Blockchain for RC verification. BC retains copies of the signed documents. If a patient or doctor wants access to the data for additional procedures, they should always utilize public keys.

3.3.2 Data Request Appendix

The data add phase allows a patient to request information from the blockchain (BC). With the patient's permission, the doctor can modify this data by submitting a request to the BC, provided both parties have the necessary medical devices for the health app. The doctor signs a registration form using a public key, which BC then verifies from both the patient and the doctor before moving on to the next process.

First, check the signature's legitimacy with T shows the timestamp as it is right now. The app on the doctor's phone establishes:

$$r = H(sk_d, T) \quad \ldots(3.9)$$

And determines:

$$R = rP \quad \ldots(3.10)$$

Additionally, the doctor's app on a mobile device provides the key:

$$k = rpk \quad \ldots(3.11)$$

The related ciphertext is as follows:

$$C_t = Ek(m, T) \quad \ldots(3.12)$$

This quality Cl combined with T, R to the patient is communicated. The mobile phone of patient's app can also calculate:

$$k = sk_p \mathrm{R} \quad \ldots(3.13)$$

From the previous stages, decrypt Cl. The doctor can receive the message after it has been decrypted, and the patient's data can then be added to the message. BC generates a signature based on the BC check by computing:

$$C_2 = sk_P H(pk_p | pk_d | C_I | R | T) \oplus kH(id_p | R | C_I \mid T) \quad \ldots(3.14)$$

BC calculates for signature verification:

$$K = kP \quad \ldots(3.15)$$

following receipt of tuple data to the patient from the BC:

$$C_{SR} = (id_p, pk_i, T, R, K, C_l, C_2) \quad \ldots(3.16)$$

When a patient transmits this to a doctor, the doctor continues adding to the patient's data and makes modifications.

The blockchain system checks timestamp values for requests from patients and physicians using their registered public keys and performs several key validations after verification. These include validating signatures, confirming patient authorization for data access, ensuring data integrity, checking the timeliness of requests, verifying the use of authorized medical devices, and ensuring compliance with data protection regulations. These steps are vital for maintaining the integrity, security, and trustworthiness of data exchanges on the blockchain.

$$C_2P = pk_PH\left(pk_p|pk_{\dot{a}}|C_l|R|T\right) + KH\left(id_p|R|C_I \mid T\right) \quad \ldots(3.17)$$

The BC invoice additionally confirms that the data is from the patient's ID id_p and $\mathrm{pk_d}$ The doctor's public key must be available for the further stages using BC. BC will proceed to append data after the verification process has verified the public keys. The CSR is finally stored by BC. BC will provide the information to the doctor only after going through a number of measures to confirm the request made by the appropriate patient or doctor $\mathrm{id_d}$.

The patient and the doctor each utilize their publicly available key, send the BC a different request in order to retrieve data order to confirm that they both shared the timestamp $\left(id_p, id_d, T_p\right)$, to verify that the timestamp is utilized by both parties at the same time T_ρ , and to make sure both parties are reliable, the patient must authorize prior to signing up for the system, and the BC was asked to speak with a doctor.

When a physician or patient requests information from the blockchain (BC), the process consists of two steps. First, the patient submits an authorization request that includes data retrieval and a timestamp, which is sent to the BC for verification. Second, the BC verifies the signatures of both the patient and physician, and upon successful verification, retrieves the patient's record for processing, with the data format specified in the output.

$$C_{SR'} = \left(pk_p, T, R, C_1, C_2\right) \quad \ldots(3.18)$$

Considering that the doctor is removing the saved tuple, and told since the doctor is aware of it, id_p is converted to pk_p and pk_d. BC will confirm the accuracy of C_1 the doctor can make the following calculations with the available data:

$$k = H(sk_i, T)p \quad \ldots(3.19)$$

p as well as values for decryption C_l.

4. Result and Analysis

This section assesses the performance of the proposed framework to identify its effectiveness and reduce the risks linked to the advanced technology, which is understood by only a select few.

4.1 Experimental Setup

It has been put to the test the suggested framework's functionality using the following configurations in our experiments:

- Intel(R) Core (TM) i7-11800H, 11th generation, 2.30 GHz
- Additionally, Windows 64-bit OS (version 11) comes with 16.00 GB of RAM.

In order to create our suggested framework, we used:

- Front End: ReactJS, HTML, CSS, JS)
- Back End: NodeJS, Ganache (Blockchain), Truffle Framework, Metamask Browser extension (Connect Browser to Blockchain), Some NodeJS Libraries
- IDE: VSCode

4.2 Performance Evaluation Data Collection

This section describes the data types utilized for assessing the proposed framework's performance and details the metrics used for interpreting the performance review results.

1) Transaction Data

The performance of the suggested framework is assessed using the transaction data and related details below.

- Transaction Deployment Time (tx1)

Its definition is "the moment a transaction is deployed." This moment of deployment in Ethereum refers to the time that the transaction is used to deploy a smart contract.

- Time Spent Completing Transaction (tx2)

It's described as the moment that the Ethereum blockchain, in this instance, completes and confirms the transaction.

2) Valuation Metrics

The suggested framework's throughput, latency, and execution time are among the metrics used for evaluation. The following is a brief explanation of these:

- Execution Time is the duration measured in seconds that elapses between a confirmation and execution of transactions on the blockchain. In mathematics, Min (tx1) - max (tx2) is meant to measure duration.
- Throughput is used to describe the volume of data that can be moved in a specific amount of time from one place to another.
- The latency is the time that passes while a part of the system waits for another system component to act in response to an event. That could be known as the time variation in the transaction's deployment and completion durations.

4.3 Results and Discussions

1) Performance Assessment

It used Apache Version 2.00 and Apache JMeter version 5.1.1 to evaluate performance. to comprehend the performance of our suggested framework in an actual instance of multiple users using the framework for various purposes. A tool for desktop performance testing called Apache JMeter is applied to application examination and evaluation [20].

Average Time of Execution

When the quantity of transactions rises, so does the execution time. These transactions are conducted for the various objectives specified by the smart contract, the algorithm of which is outlined in the preceding chapter. The time it takes to perform the Patient Records Addition, Assign Roles, as well as View Patient Records features on a single user system is 18.29, 1 minute, 48, and 50 seconds, in that order. Once 100 users are utilizing the system at once, this time will increase.

- Throughput

The different functions that are part of the proposed framework's smart contract are explained in Algorithm 1.

Algorithm (1) Pseudo-code Implementation for Smart Contract of Patient Records

Pseudo-code Implementation for Assign Roles function:

```
function for account in
// Define a mapping structure for roles
mapping<string, list<string>> rolesMapping;
// Function to define roles
function defineRoles(newRole, newAccount)
  // Check if the role already exists
  if newRole not in rolesMapping
    // Initialize empty list for new role
```

```
    rolesMapping[newRole] = new list<string>();
  // Check if the account is already assigned to this role
  if newAccount not in rolesMapping[newRole]
    // Add new account to the role's list
    rolesMapping[newRole].add(newAccount);
    print("Account '" + newAccount + "' assigned to role '" + newRole + "'.");
  else
    print("Account '" + newAccount + "' is already assigned to role '" + newRole + "'.");
end function
```

Pseudo-code Implementation for Add Data function:

```
// Assuming patientRecords is a data structure to hold patient records
patientRecords = {}
// Function to add a patient record
function addPatientRecord(msgSender, patientId, patientData):
    // Check if the sender is a doctor
    if msgSender is in rolesMapping["doctor"] then
        // Initialize patient's record if it doesn't exist
        if patientId not in patientRecords then
            patientRecords[patientId] = {} // Create a new record for the patient
        // Add the provided data to the patient's record
        patientRecords[patientId].update(patientData) // Update the record with new data
        return "Patient record updated successfully."
    else
        // Abort session if the sender is not a doctor
        return "Session aborted: Access denied."
    end if
end function
```

Pseudo-code Implementation for view Patient Record function:

```
// Assuming patientRecords is a data structure to hold patient records
patientRecords = {}
// Function to view patient record
function viewPatientRecord(msgSender, patientId):
    // Check if the sender is a doctor or the patient themselves
    if msgSender is in rolesMapping["doctor"] or msgSender is patientId then
        // Check if the patient identifier exists
        if patientId in patientRecords then
            // Retrieve data from the specified patient ID
            patientRecord = patientRecords[patientId]
            return patientRecord // Return the patient record to the account that requested the
operation
        else
            // Patient ID does not exist
            return "Error: Patient ID not found."
        end if
    else
        // Abort session if the sender is neither a doctor nor the patient
        return "Session aborted: Access denied."
    end if
end function
```

Pseudo-code Implementation for Update Data function:

```
// Assuming patientRecords is a data structure to hold patient records
```

```
patientRecords = {}
// Function to update patient record
function updatePatientRecord(msgSender, patientId, patientName, updatedData):
    // Check if the sender is a doctor
    if msgSender is in rolesMapping["doctor"] then
        // Check if the patient ID and name match
        if patientId in patientRecords and patientRecords[patientId].name == patientName
then
            // Update data for the specified patient record
            patientRecords[patientId].update(updatedData) // Assuming updatedData contains
key-value pairs for updates
            return "Success: Patient record updated."
        else
            // Return failure if ID or name does not match
            return "Fail: Patient ID and name do not match."
        end if
    else
        // Abort session if the sender is not a doctor
        return "Session aborted: Access denied."
    end if
end function
```

Through the use of JMeter, we were able to simulate a range of between 100 and 500 users utilizing the system and carrying out its different functions over a period of 10 to 35. JMeter's rate of flow is expressed in KB/sec, Data/time, units. We assessed the system's performance and the number of users was simulated as stated above while carrying out the experiments. The suggested framework is used to run these simulations, and throughput is examined at the conclusion. Figure 3 below illustrates the proposed framework's throughput. During the experiment, the system's throughput was discovered to significantly increase as the quantity of users and requests grew, in a linear manner. The efficiency of the linear increase in throughput illustrates the proposed framework.

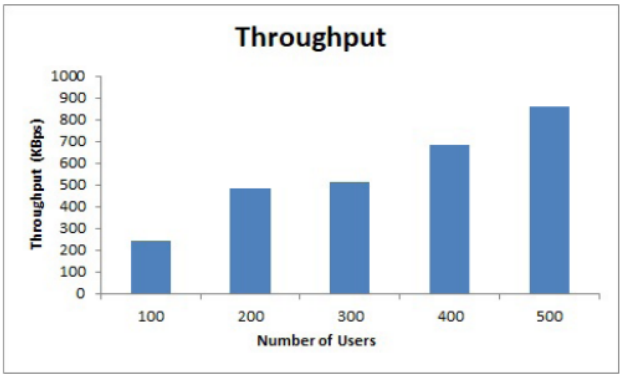


Figure 3: The rate of the suggested architecture.

Latency Average

According to the previous definition, delay is the interval of time between a request sent by one system component and the generation of a response by any additional system element. Latency is the distinction between these two actions. Here, we have used JMeter to assess the typical delay of the suggested structure. It used JMeter to simulate the number of users while assessing the proposed framework's latency. Milliseconds are used in JMeter to measure latency. Figure

4. of the accompanying graph provides a summary of average system latency and the system's throughput suggested framework. In this trial, the maximum latency recorded was 14 ms.

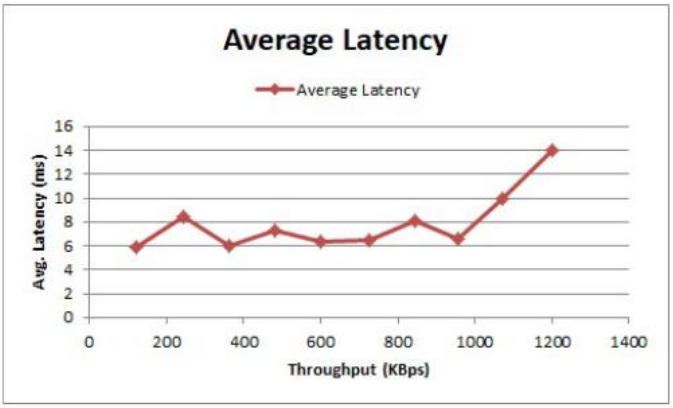


Figure 4: The rate of the suggested architecture**.**

The effectiveness of the proposed structure was assessed through an analysis of transaction volume and cost, along with an examination of the exchange payload to establish transaction size.

2) Performance Evaluation (Transaction)

In an Ethereum transaction, the data payload field is present. That transaction contains a payload of data intended to trigger smart contract capabilities. This payload of data has bytes attached to it and is serialized in hexadecimal format. Here, we'll talk about two operations using Algorithm 1 to understand the payload of data that's being generated along with the transactions. The transaction's optional data payload field is only used in situations where contract functions are accessed in some way. It consists of two key sections:

- Selector for Function
- Function-Based Debates

The function selector, which determines which smart contract function is being called, utilizes the first four bytes of the Keccak-256 hash. Different static and dynamic element types are included in the function arguments, and their encoding in the payload follows distinct rules. To grasp how the data payload is created, let's now examine the output of algorithm 1's defined roles function. First, we would divide the arguments and the function selector. In this case, the function signature is the function selector, actually:

```
DefineRoles (string, address)
```

The Keccak-256 hash for the aforementioned function is the following:

```
0x6c0abd24edce8ce20a2dfb1cd2026179214
468cde47681e871b6e14bf9d39efd
```

The function selector occupies the first four bytes of the hash, which indicates the function being called from the contract that was generated (0x6c0abd24). Let's now examine the encoding of the function arguments after the function selector has been calculated. In order to do this, we encrypt the first part of each of the two arguments: the dynamic type is string, and the static type is address. While the offset in bytes is used for the dynamic type, and is additionally measured starting with the static type of value encoding is passed directly. It does not count the first four bytes that contain the function signature hash. The dynamic type string "Doctor" would have the following encoding:

```
0x0000000000000000000000000000000000000
000000000000000000000000006
```

Six elements, or bytes, make up the total, and their representation is shown above. The string value "Doctor" is shown next, padded to the right by 32 bytes.

```
0x48976c7c7f2c20667f626c6421000000000
0000000000000000000000000000000
```

This function's encoding, limited to the following, is its dynamic types:

```
0x6c0abd24
  0x0000000000000000000000000000
  0000000000000000000000000000000000000006
  0x48976c7c7f2c20667f626c64210000000
  0000000000000000000000000000000
```

In this function, both the dynamic and static types would each have a 32-byte size. This function has a total byte size of 64 bytes. It is capable of computing the additional algorithm's data payload functions in a similar manner. The amount of data payload for each function is shown in Table 1.

Table 1: Transaction payload data used in the suggested framework.

Data Payload	Constants types	size
Payload of $F_{asssign}$	String, address	64 bytes
Payload of $F_{address}$	String,string,string, String,string, address	192 bytes
Payload of F_{view}	string, unit	64 bytes
Payload of F_{update}	string, unit,string, string, string	160 bytes
Payload of F_{delete}	string, unit	64 bytes

It can compute the various functions' transaction sizes within our suggested structure by utilizing the data payload. For these functions, the transaction sizes are shown in bytes in Table 2. It is important to remember that the size of the transaction determined in this section is specifically computed for various Algorithm 1 functions and is from the perspective of the data payload, whereas the transaction size was determined in a more generalized way and from the perspective of the block size.

Table 2: Size and cost of transactions for the suggested framework.

Data Payload	Constants types	size
Payload of $F_{asssign}$	String, address	64 bytes
Payload of F_{add}	String, string, string,	192 bytes
Payload of F_{view}	String ,string, address	64 bytes
Payload of F_{update}	string, unit string, unit ,string,	160 bytes
Payload of F_{delete}	string, string string, unit	64 bytes

4.4 Proposed Framework Comparison with Related Work

It also discusses a few of the parameters that our framework uses and compares them to related research in this domain. Ensuring that the framework has these parameters included does not come at the expense of the privacy and security of the system. In every one of the parameters discussed below, as shown in Table 3, security and privacy are addressed.

Table 3: The suggested framework is compared to relevant works

Parameters	Existing [21]	Proposed
Blockchain-Based	N	Y
Scalability	N	Y
Access Control	Y	Y
Decentralized	N	Y
Interoperability	N	Y

The proposed framework was evaluated through 50 independent authentication attempts, demonstrating an average smart contract execution time of 2.3 seconds with a standard deviation of 0.15 seconds. The system achieved a success rate of 98% under simulated attack conditions, indicating robust security performance. These metrics underscore the framework's efficiency and reliability in securing health data transmission.

5. Conclusions

The traditional medical record-keeping system is impractical and takes up considerable storage space for unstructured patient test data, making information transfer difficult. More data generation currently takes place in healthcare systems. Therefore, improvement in data management processes must consider security and confidentiality. Amendment of fundamental data management processes might be required to provide solutions to such privacy concerns, wherein possible solutions have been presented by blockchain technology. The proposed system is expected to significantly improve healthcare data management through a decrease in access time, increasing patient satisfaction due to more streamlined and secure processes, and minimizing privacy breach risks. These improvements collectively contribute to more efficient and trustworthy healthcare delivery, ultimately benefiting both providers and patients. The system allows patients to issue and obtain authorizations for specific records easily. It is facilitated via tap, Ethereum, and smart contracts. It includes a wallet that provides secure access to the blockchain and helps store the private key. In addition, it helps reluctant patients, whose personal information is shifting to electronic health records (EHRs), by means of strong cryptographic encryption. Among several ways the program has realized its objectives is secure medical report data exchange using blockchain technology by monitoring the safety and reliability of the processes. Overall, the proposed solution effectively handles accurate medical reporting, data exchange, and security.

6. Future Work

Blockchain technology allows for the creation and deployment of new programs based on a decentralized model, moving away from traditional cloud-based applications. The current smart contract will be expanded to enhance lookup capabilities and provide advanced features for Electronic Health Record (EHR) management systems. Future developments may include a real-time video conferencing feature, especially relevant during the COVID-19 pandemic, as well as the integration of a payment module that enables patients to pay for specialist consultations via credit or debit card within a decentralized architecture. The significance of telemedicine extends well beyond the context of the COVID-19 pandemic, offering enduring

benefits to healthcare delivery in diverse settings. Its advantages include: Enhanced Access to Care, Continuity of Care, Cost Efficiency, Emergency and Disaster Response, Preventive Care, and Wellness and Healthcare System Resilience. Verification could involve using the patient's National ID number. The introduction of Ganache opens the door for experimentation with private blockchains to improve lookup processes and support the additional functionalities necessary for EHR management along with a comparison of current and future methodologies, this comparison will be based on key performance metrics such as accuracy, efficiency, scalability, security, and user satisfaction. This comprehensive assessment will enable us to determine the improvements offered by future methodologies over current practices, guiding effective implementation strategies. Ganache is a local blockchain development environment that simulates a full Ethereum blockchain, enabling rapid and cost-effective testing. Its use in our paper facilitates: Testing Lookup Features and EHR Enhancements. Ganache supports iterative development, debugging, and validation of our blockchain-based health record system's lookup and enhancement features, thereby accelerating development and increasing system reliability.

Expanding the smart contract in the proposed system typically involves adding new functionalities to support evolving healthcare needs, such as integrating additional data access controls, facilitating smart consent management, or supporting interoperability with other health information systems. This can be achieved through modular smart contract design—using upgradeable contracts or proxy patterns—that allow new features to be added or existing ones modified without disrupting the entire system. Scalability concerns include: Transaction Throughput, Storage Limitations, Cost of Operations, Latency and Decentralization vs. Performance.

References

[1] P. Dudczyk, J. K. Patrick, and G. V. Crosby, "Blockchain technology for global supply chain management: A survey of applications, challenges, opportunities & implications," *IEEE Access*, vol. PP, pp. 70065–70088, Jan. 2024, doi: 10.1109/ACCESS.2024.3399759.

[2] H. Taherdoost, "Blockchain innovations, applications, and future prospects," *Electronics*, vol. 13, no. 2, Jan. 2024, doi: 10.3390/electronics13020422.

[3] V. Buterin, J. Illum, M. Nadler, F. Schär, and A. Soleimani, "Blockchain privacy and regulatory compliance: Towards a practical equilibrium," *Blockchain: Research and Applications*, vol. 5, 100176, pp. 1–9, 2024.

[4] M. Nadler and F. Schär, "Tornado cash and blockchain privacy: A primer for economists and policymakers," *Federal Reserve Bank of St. Louis Review*, vol. 105, no. 2, pp. 122–136, 2023, doi: 10.20955/r.105.122-136.

[5] J. B. Awotunde, E. T. Oladipupo, A. E. Adeniyi, A. O. Babatunde, and D. R. Aremu, "An enhanced internet of medical things using blockchain-enabled elliptic curve cryptography for security and privacy," *International Journal of Critical Computer-Based Systems*, vol. 11, no. 4, pp. 280–306, 2025, doi: 10.1504/IJCCBS.2024.146785.

[6] Y. Yan, "The overview of elliptic curve cryptography (ECC)," *Journal of Physics: Conference Series*, vol. 2386, no. 1, 012019, Dec. 2022, doi: 10.1088/1742-6596/2386/1/012019.

[7] D. T. Thanh, N. Q. Toan, N. V. Son, and N. V. Duan, "An algorithm to select a secure twisted elliptic curve in cryptography," *Journal of Science and Technology on Information Security*, vol. 1, no. 15, pp. 17–25, Jun. 2022, doi: 10.54654/isj.v1i15.832.

[8] D. Maimuţ and A. C. Matei, "Speeding-up elliptic curve cryptography algorithms," *Mathematics*, vol. 10, no. 19, 3676, Oct. 2022, doi: 10.3390/math10193676.

[9] M. M. Trung, L. P. Do, D. T. Tuan, N. V. Tanh, and N. Q. Tri, "Design a cryptosystem using elliptic curves cryptography and Vigenère symmetry key," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, pp. 1734–1743, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1734-1743.

[10] C. Easttom, "Elliptic curve cryptography," in *Modern Cryptography*. Cham: Springer, 2022, pp. 253–263, doi: 10.1007/978-3-031-12304-7_11.

[11] X. Chen et al., "Patient-controlled health records on blockchain: A Bitcoin-inspired model," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[12] K. M. Abirami, R. Srikanth, and R. Kavitha, "Analysis of elliptic curve cryptography methods and survey of its applications," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, pp. 430–434, 2023.

[13] H. A. A. Hadi and R. F. Ghani, "The future of e-health: Blockchain solutions with Hyperledger Fabric and IPFS," *Ingénierie des Systèmes d'Information*, vol. 30, no. 1, pp. 243–255, 2025, doi: 10.18280/isi.300121.

[14] A. Sudarsono, M. Yuliana, and H. Darwito, "A secure data sharing using identity-based encryption scheme system," in *Proc. Int. Conf. Science in Information Technology (ICSITech)*, 2018.

[15] H. A. A. Hadi and R. F. Ghani, "Secure cloud-based storage framework for EEG files in blockchain e-health systems using compression, digital signatures, and chaotic encryption," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 4, p. 851, 2025, doi: 10.22266/ijies2025.0531.55.

[16] V. Ramani, T. Kumar, A. Breaken, M. Liyanage, and M. Ylianttila, "Secure and efficient data accessibility in blockchain-based healthcare systems," in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, Abu Dhabi, UAE, 2018.

[17] T. Kumar, V. Ramani, I. Ahmad, A. Breaken, E. Harjula, and M. Ylianttila, "Blockchain utilization in healthcare: Key requirements and challenges," in *Proc. IEEE 20th Int. Conf. E-Health Networking, Applications and Services (Healthcom)*, 2018.

[18] T. Kumar, A. Breaken, V. Ramani, I. Ahmad, E. Harjula, and M. Ylianttila, "SEC-BlockEdge: Security threats in blockchain-edge based industrial IoT networks," in *Proc. 11th Int. Workshop Resilient Networks Design and Modeling (RNDM)*, 2019.

[19] O. Kuznetsov, P. Sernani, L. Romeo, and E. Frontoni, "On the integration of artificial intelligence and blockchain technology: A perspective about security," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2023.3349019.

[20] M. Niranjanamurthy, K. Kumar S, A. Saha, and D. D. Chahar, "Comparative study on performance testing with JMeter," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 70–76, 2016.

[21] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 267–278, Jul. 2018.