RBHAP-HLB framework with high data privacy for secured EHR storage

R. Saranya[1], A. Murugan[2,*]

[1]Research Scholar, Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India

[2]Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India

Emails: sr5287@srmist.edu.in; murugana@srmist.edu.in

Abstract

For data security and integrity, the sharing of Electronic Health Records (EHRs) utilizing blockchain is becoming a vital vision. However, blockchain and storage wielded in prevailing studies arises security and scalability issues. To overcome these issues, this paper proposes a novel Quadratic Interpolation-based Brownian Motion-Double Elliptic Curve Cryptography (QI-BM-DECC)-centric EHR securing in Hyper-Ledger Blockchain (HLB) with Inter-Planetary File System (IPFS). Primarily, the patient and doctor are registered on the hospital website; then, the keys and QR codes are generated for the patient. After that, the patient login with the credential details, QR code, and the purpose of login. The patient did the online consultation booking after successful login; then, the consultation is done grounded on the time scheduled by the doctor. Afterward, the patient securely uploads the EHR on the HLB with IPFS utilizing QI-BM-DECC. Meanwhile, an attribute-centric hashed access policy is created with the selected attributes. After that, utilizing the Mean Public keys- Digital Signature Algorithm (MP-DSA) approach, the hashed access policy is signed. When a doctor request for EHR access, the signature is verified and the access request is sent to the patient. Now, the doctor downloads the EHR from IPFS after being accepted by the patient. The experiential outcomes exhibited the proposed technique's dominance over the other mechanisms.

Keywords: Electronic Health Record (EHR); Hyper-Ledger Blockchain (HLB); Inter-Planetary File System (IPFS); Directed Acyclic Graph (DAG); Hashed access policy

1. Introduction

To generate connected environments and develop integrated patient health records, recent advancements in medical services have created new demand for efficient treatment in healthcare services [1]. The documents are stored as EHRs to satisfy these requirements, thus enabling the care providers to give good care to the patient and make quick decisions [2]. Patients often outsource their PHRs to cloud servers owing to storage requirements. Several cloud-centric healthcare applications were developed [3]. Nevertheless, providing control to the third parties in the cloud servers has various issues, namely less privacy [4]. Hence, since the EHRs contain more private information, cloud storage is not reliable. Authorized individuals should only attain permission for storing and retrieving data; in addition, interactions betwixt a patient and the system require to be secured [5]. To attain these requirements, research was made on IPFS and a typical distributed storage application that employs blockchain as a core infrastructure [6].

Since data security is one of the major concerns, the need for blockchain technology in the healthcare field is mounting day by day [7]. High degrees of security and privacy to healthcare operations are provided potentially by blockchain with its decentralization along with traceability [8]. For securing the EHRs, namely Ethereum, Hyperledger, and Consortium blockchain, various blockchain technologies are utilized. Among these, the Hyper-ledger, which utilizes consensus algorithms for securing EHRs, is more reliable.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Hyperledger, which provides the services like confirmation of the transaction, authentication, along with access protection, is a blockchain technology. In addition, participants could track the transactions' visibility, and do not require mining [9]. However, EHR's huge volume is not appropriate to be deposited in HLB; thus, for medical data storage, the IPFS is introduced; also, for attaining decentralized data storage, the data in the HLB is stored in the hash address returned by IPFS [10]. In the prevailing works, numerous transactions were generated cantered on the hashed access policies for securely accessing the EHR. Nevertheless, such techniques are limited to privacy. Thus, to overcome these issues, this work proposes a novel QI-BM-DECC algorithm with HLB technology-centric secure EHR sharing. In the proposed system, a novel attribute-centric hashed access policy is presented and only after patient's grants permission, the doctor is eligible to access the EHR.

A. Problem statement

Even though various EHR sharing systems cantered on blockchain technology were introduced, several drawbacks are present in the prevailing techniques, which are enlisted further,

- ✓ The leakage of transaction privacy is the blockchain's chief vulnerability.
- ✓ In the existing system, confidentiality will be lower in distributed healthcare networks; in addition, integrity is a main concern.
- ✓ In IoT data storage, the main challenges are how to share and protect sensitive data.

By analysing these drawbacks, the proposed model aims in developing a secure EHR sharing system that provides integrity, confidentiality, privacy, and security.

The paper's structure is systemized as: Section 2 evaluates the related works; Section 3 explicates the proposed technique; Section 4 elucidates the proposed system's outcomes; lastly, the paper is winded up in section 5.

2. Related Work

Tanwar et al., 2020 [11] examined a blockchain-centric EHR system for healthcare applications. An access control policy approach was developed by the model, which also employed the Hyperledger-centric EHR sharing system. The presented model attained enhanced outcomes on throughput. Nevertheless, owing to the developed consensus mechanism, some of the nodes in the network failed to respond.
Li et al., 2022 [12] propounded a blockchain-cantered EHR system (EHRChain), which utilized an attribute-centric homomorphic cryptosystem. Here, the Semi-Policy Hiding and Dynamic Permission Changing for Partial Ciphertext-Policy Attribute-Based Encryption were presented. The experiment validated that the presented model outperformed other EHR-sharing systems. However, the developed model took more time for achieving security with the usage of large keys.

Antwi et al., 2021 [13] explored Hyper Ledger Fabric (HLF) as the blockchain solution for healthcare applications. Testing scenarios were created by the strategy, which further tested the HLF on such scenarios for analyzing the blockchain-enabled security criteria. The assessment exposed the private blockchain technologies' propitious benefits regarding security. Nevertheless, the investigated HLF models lacked confidentiality.

Shuaib et al., 2022 [14] recommended a secure decentralized EHR sharing system grounded on the blockchain. The model was grounded on the Istanbul Byzantine Fault Tolerant (IBFT) consensus approach as well as IPFS. The outcomes displayed that when analogized with the prevailing Blockchain-centric techniques, the model performed well with better security. However, the traffic in the blockchain mounted when a sheer number of transactions were performed.

Kaur et al., 2021 [15] explored a blockchain-centric system for secured storage, sharing, along with querying of EHR. The developed blockchain model deployed HLF; also, the EHR was stored on the DataBase (CouchDB). As per the outcomes, the presented system exhibited superior efficacy. However, CouchDB could not be reliable as it took large space than other databases.

Mani et al., 2021 [16] presented hyperactive ledger Healthchain for the patient-centric IPFS-centric storage of health records. In HLF, the health record hashes were stored as health record chains; also, the encrypted file was stored on the IPFS. In collaborating and sharing health records, stakeholders were provided with high confidence by this model. However, when the low-activity node participated in the consensus, the consensus mechanism stagnated.

Marangappanavar & Kiran, 2020 [17] probed into an IPFS-enabled blockchain solution for securing the PHR. To ensure security, smart contracts that utilized the give View Permission and revoke View Permission functions were utilized in this model. The experiential outcomes signified the strength of the investigated model. However, the model to secure and retrieve the PHR. consumed more energy and resources.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Z. Sun et al., 2022 [18] employed a blockchain-cantered secure storage methodology for medical information. The secure storage was cantered on the HLF as well as on the Attribute-based access control framework. The experiential outcomes demonstrated that security and integrity were provided by the combination of access-control attributes and blockchain. However, owing to the consensus algorithm, scalability could not be attained.

3. EHR Storage On IPFS With Ledger Blockchain Methodologies

EHRs, which hold numerous personal medical information, are highly privacy sensitive. Thus, the sharing of EHR securely is a concern in recent days. During the sharing of EHRs, blockchain technology provides security and privacy. Hence, an HLB technology-based secure sharing of EHR on IPFS utilizing the QI-BM-ECC algorithm is proposed. Figure 1 depicts the proposed techniques.

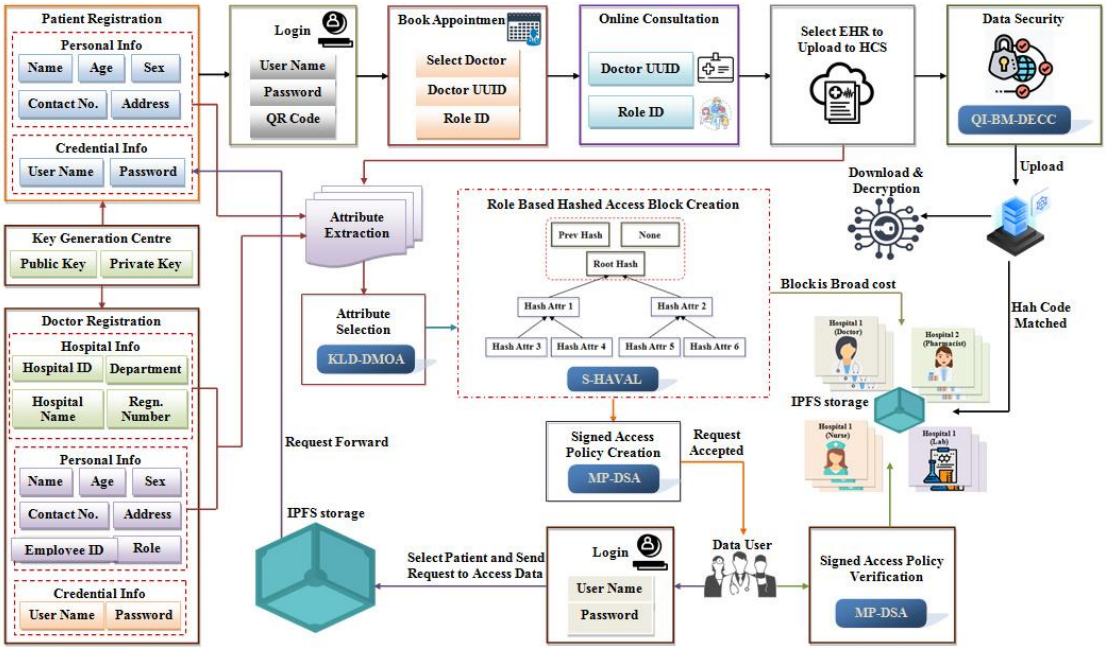

Figure 1: Framework of the proposed secure EHR sharing model

A. Registration

Primarily, in the proposed framework, the doctors and the patients register their details in the network and obtain login credentials.

Patients: The patients register on the hospital website by providing their personal info, such as names, ages, sex, contact numbers, and address credential information, such as usernames and passwords. During registration, a unique QR code will be generated for users, which can be wielded for a secure login procedure.

Doctors: The doctors register on the same website by providing information similar to the patients, but additionally personal information of Employee ID and designation also the hospital information, namely Hospital ID, department, hospital name, and registration should be given.

B. Keys generation for patient and doctor

By utilizing the QI-BM-DECC algorithm, the public and private keys are generated for patients and doctors during the registration process. The prevailing Elliptic Curve Cryptography (ECC) is vulnerable to attacks by exploiting the public parameters of ECC. Thus, in the proposed encryption, the keys are generated grounded on the Brownian Motion (BM); also, the secret number can be generated utilizing the Quadratic Interpolation (QI) function [19-20]. Therefore, key generation utilizing the QI-BM-DECC algorithm is given as; an elliptic curve $\mathcal{E}$ of the QI-BM-DECC is the set of all the points, whose coordinates satisfy the polynomial equation,

$$e^2 = t^2 + \upsilon t + \zeta \qquad (1)$$

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Where, $e,t \in E$ are the variables and υ,ζ represents the constants; also, these values can be real, imaginary integers, complex, or any other forms. The doctor and the patient sides agree on a point $\varpi(e,t)$ in $\varepsilon(e,t)$. Here, the agreement point was computed with the BM rather than utilizing a prime number as a private key (ϖ).

$$\varpi = \frac{\aleph \times \ell - e}{sd} \tag{2}$$

Where, $\aleph$ elucidates the drift rate of the data point e,t, sd exemplifies the standard deviation of the data dimensions, and ℓ explicates the time step. After that, a random integer ϑ_{ri} is chosen, then calculates $(\vartheta_{ri}\varpi)$ at the patient's side, and sends it to the doctor's side. The doctor's side chooses a random integer ϑ_{ir}, then computes $(\vartheta_{ir}\varpi)$ and sends it to the patient's side.

After that, the public key (P_k) of the doctor and patient side is generated with the shared $(\vartheta_{ri}\varpi)$ and $(\vartheta_{ir}\varpi)$ as,

$$P_{patient} = \vartheta_{ri}(\vartheta_{ir}\varpi) \tag{3}$$

$$P_{doctot} = \vartheta_{ir}(\vartheta_{ri}\varpi) \tag{4}$$

Where, $P_{patient}, P_{doctor}$ portrays the same public key for the patient and the doctor sides, which is signified as P_{key}. With the keys generated, the secret key is computed, which is shared between the patient and the doctor's side utilizing the QI formula as,

$$sk = \zeta_0 + \zeta_1(P_{patient} - P_{doctor}) \tag{5}$$

Where, sk represents the secret key, ζ_0, ζ_1 specifies different constant values.

C. Login

After successful registration, if a patient $(patient)$ wishes to consult a doctor to upload EHR files and to accept the access request, the patient should log in to the website by inserting the user name, password QR code, and the purpose of login. If the given details matched the registered information, the user is allowed to access the service.

D. Online consultation

The user books an appointment with the doctor $(doctor)$ after successful login. At that time, the Universally Unique Identifier (UUID) will be generated and shared with the selected $doctor$ and patient for every appointment time with their user id. After that, by inserting the user id and password, the $doctor$ logs in and fixes the appointment time. After the appointment is booked, the patient consults with the $doctor$; also, with that time, the EHR should be securely shared with the $doctor$. The EHR is mathematically specified as E.

E. EHR sharing

In the proposed model, the EHR utilized is the PCOS dataset, which contains information that is more private; hence, this EHR should be shared securely. To securely store the EHR, the EHR was encrypted with the QI-BM-DECC algorithm and securely stored on the IPFS. Meanwhile, a digital signature access policy is created and verified to access the EHR in the HLB.

IPFS: The IPFS, which could store large files and could be efficiently managed, is a distributed file storage system. Here, the EHR of the patient who is present in the blockchain is encrypted and stored in the secure off-blockchain IPFS. In the IPFS storage, the EHR file is uploaded securely by encrypting the file utilizing the QI-BM-DECC algorithm, and the $doctor$ can decrypt the file utilizing the public keys provided to them. Therefore, the encryption and decryption of the EHR file are given as,

Encryption: While encryption time, the original data would be encrypted utilizing the receiver's first public key with a secret key and again it will be encrypted utilizing the second public key as well as the secret key [21-22]. Here, the secret key will be multiplied by the encryption formula. Thus, the encryption is mathematically expressed as,

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

$$cip_1 = \lambda * P_{key1}$$
$$cip_2 = E * cip_1 * sk \quad (6)$$

$$cip_3 = cip_2 + P_{key2}$$
$$cip_4 = E + cip_3 * sk \quad (7)$$

Where, $cip_1, cip_2, cip_3, cip_4$ specifies the obtained cipher texts of patient EHR during encryption, and P_{key1}, P_{key2} represents the first and the second public key generated in each iteration [23]. This data is uploaded in the ledger HLB. Meanwhile, to access the EHR on the IPFS, an efficient role-based access policy is created.

a. Hyper-Ledger Blockchain (HLB)

Hyperledger, which was created for supporting the distributed blockchain technology developed by linux, is an open-source project. The HLB comprises two states: (1) the world state, which connects the blockchain to the IPFS, and (2) the transaction logs, which are for access policy verification.

Smart contracts: The predefined conditions that run in the HLB to verify the transactions are named smart contracts. For EHR sharing, the smart contract can be written with the terms agreed upon by both parties (patient and doctor side) and written to the ledger. Here, hashed access policy is created for smart contracts. The hashed access policy is created by the *patient* in HLB and *patient* decides which *doctor* in the hospital can access the hashcode of the EHR, as *patient* is the owner of the EHR shared; he/she has the authority to give different access rights and permissions to the selected doctors. Once the policies are written, the signatures attached to the transactions are validated as if the signatures fulfil the governance agreed to access policy created by the hyperactive ledger. Figure 2 depicts the pictorial representation of the HLB.

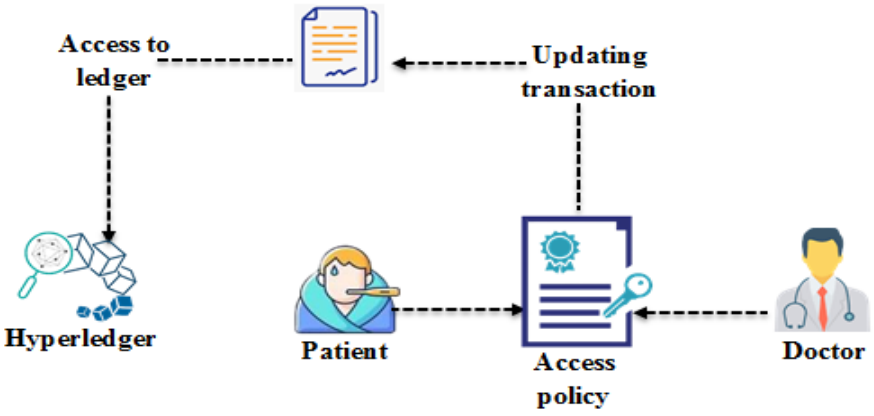

Figure 2: Proposed HLB access.

In the proposed hyperactive ledger model, the Merkle root MR is created with the attribute-centric hashing algorithm, which is broadcasted to the HLB block. The MR fingerprint (hashed value) is stored in the HLB since IPFS could use the fingerprint of the file to locate its address so that EHR recovery can also be performed easily.

The HLB is a private permission blockchain, here the access control policies are stored and enhanced the integrity of the network. The HLB contains the MR in header and time stamp, hashcode, previous hash value, and transactions in the body. The HLB performs consensus mechanisms to ensure the security of the transactions.

Consensus mechanisms: Consensus in HLB is a process where the nodes participate in the network providing an ordering of transactions and validating those blocks of transactions that need to be presented in the ledger of HLB. Unlike other blockchain technologies, various sorts of consensus, namely Proof of Stake (PoS), PBFT, Proof of Work (PoW), IBFT, et cetera are utilized by the HLB. However, those consensus mechanisms are consuming processes; however, the HLB chain stores millions of data, which made the consensus mechanisms unreliable. Thus, instead of the consensus mechanism, the Directed Acyclic Graph (DAG) mechanism is wielded in the proposed HLB. Using the DAG, each new transaction submitted to a block requires the confirmation of at least two earlier transactions before it is successfully recorded onto the network. Moreover, DAG does not require miners for authenticating block transactions. The functions performed with DAG in the HLB are as follows,

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

- ✓ The parent transactions are determined by the DAG system.
- ✓ After that, the system signs their hashes and encloses the next transactions.
- ✓ Afterward, a Merkle-tree structure of transactions is formed in which every single transaction would be considered, confirmed, along with unchanged.

Thus, by employing DAG, secure and energy-efficient transactions can be performed in HLB. Figure 3 elucidates the EHR uploaded by the proposed technique in HLB,

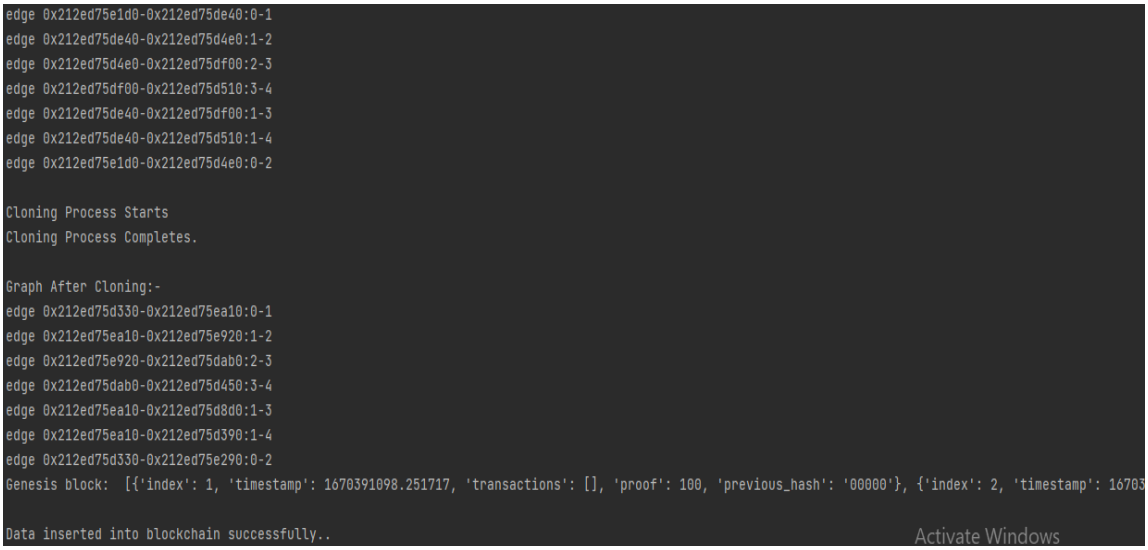

Figure 3: HLB transactions in the proposed framework

b. Attribute-based hashed access policy

The Access Policy Creation (APC) and verification for the transaction in the HLB are illustrated in the following sub-sections,

Attribute extraction

The attributes of the EHR, attributes of the patient personal information, and attributes of the $doctor$, namely hospital information and personal information of $doctor$ are extracted to create an attribute-based access policy. Hence, the extracted attributes are represented as,

$$A = \{a_1, a_2, ..., a_m\} or\, a_x, x = 1,2,...,m \quad (8)$$

Where, A specifies the variable set and a_m signifies the extracted m^{th} attribute.

Attribute selection

After the attribute extraction, the vital attributes are selected for creating an access policy in the HLB. Here, the attributes are selected with the Kullback Libler Divergence Optimization Algorithm (KLD-DMOA). The Dwarf Mongoose (DM) constructs a sleeping mound in which a rich source of food was found. Hence, the Kullback Libler Divergence (KLD) is employed in the prevailing DM Optimization Algorithm (DMOA) to enhance the sleeping mound. The working procedure of KLD-DMOA is given further,

Population initialization

In the KLD-DMOA approach, the DM population is stochastically generated. Here, the initial population is the extracted features. Thus, the population initialization is expressed as,

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

$$A=\begin{bmatrix} a_{1,1}\, a_{1,2} \cdots a_{1,n-1}\, a_{1,n} \\ a_{2,1}\, a_{2,2} \cdots a_{2,n-1}\, a_{2,n} \\ \vdots \quad \vdots \qquad \vdots \quad \vdots \\ a_{m,1}\, a_{m,2} \cdots a_{m,n-1}\, a_{m,n} \end{bmatrix} or\ a_{x,y}, y=1,2,..,n \tag{9}$$

Where, $a_{m,n}$ implies the position of the DM m in the problem dimension n, A signifies the population set, and m specifies the size of the population. The distribution of $a_{x,y}$ within the dimension n is represented as,

$$a_{x,y}=\Re\left(L_b, U_b, n\right) \tag{10}$$

Here, L_b, U_b exemplifies the lower bound and upper bound of the dimension n, and $\Re(\ )$ is the uniform random distribution. Moreover, the DM is divided into three groups; the alpha group, the scout group, and the babysitters.

Alpha female selection

The alpha female (F) is the controller of the DM family. Therefore, the female alpha can be selected from the DM population utilizing the fitness function as,

$$F=\frac{f(x)}{\sum_{x=1}^{m} f(x)} \tag{11}$$

Where, $f(\ )$ elucidates the fitness function. Here, fitness is considered as a role-centered attribute. The number of F in a group is obtained by subtracting the number of babysitters (ω) from the DM group $(m-\omega)$.

Exploitation

The F creates a peep sound termed $\Im$ to keep the DM population on the right track toward the food source. The position of the sleeping mound is determined by the abundant food position. The best position (abundant food source position) obtained by the F DM x is formulated as,

$$a_x^{It+1}=a_x^{It}+\Im\times\rho \tag{12}$$

Where, a_x^{It+1}, a_x^{It} is the position of the F DM x in the iteration $It+1$ and It and ρ specifies a random uniformly distributed number. After that, the fitness for the position a_x^{It+1} is evaluated. If $f\left(a_x^{I+1}\right)$ is better than the $f\left(a_x^{It}\right)$, then update a_x^{It+1} as the best position and construct the sleeping mound. Here, to enhance the sleeping mound, the KLD is wielded to evaluate fitness as,

$$M_x^{It+1}=\sum_{x=1}^{m} f\left(a_x^{It+1}\right)\log\frac{f\left(a_x^{It+1}\right)-f\left(a_x^{It}\right)}{\max\left[f\left(a_x^{It}\right), f\left(a_x^{It+1}\right)\right]} \tag{13}$$

Where, the sleeping mound obtained DM x is notated as M. After the sleeping mound is found, the average value of the sleeping mound is given, which is computed as,

$$\alpha_{It+1}=\frac{\sum_{x=1}^{m} M_x^{It+1}}{m} \tag{14}$$

Here, α elucidates the average value given to the babysitters. If the time step $S \geq P$, P indicates the babysitter exchange parameter exchange babysitters.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Exploration

After the exchange criterium is obtained, scouting, which analyzes the next sleeping mound determined by the scouters, is the next stage. Scouters' next position is updated as,

$$a_x^{It+1} = \begin{cases} a_x^{It} - cc * \rho * ra * \left(a_x - \vec{Z}\right) if \left(\alpha_{It+1} > \alpha_{It}\right) \\ a_x^{It} + cc * \rho * ra * \left(a_x - \vec{Z}\right) else \end{cases} \quad (15)$$

Where, $cc = \left(1 - \frac{It}{It_{\max}}\right)^{\left(2 \times \frac{It}{It_{\max}}\right)}$ specifies the mongoose group's collective-volatile movement that linearly abates during iterations, $It_{\max}$ indicates the maximum iteration, ra represents a random number, and $\vec{Z}$ indicates the motion vector of the scout group, which is symbolized as, $\vec{Z} = M_x^{It+1}$. The best position a_x^{It+1} gives the optimally selected attribute values. The optimal attribute selected is represented as,

$$Y = \{y_1, y_2, ..., y_q\} or\ y_\beta, \beta = 1,2,..,q \quad (16)$$

Where, Y exemplifies the selected attribute set and y_q implies the q^{th} optimal attribute. The pseudocode of the proposed KLD-DMOA is as follows,

---

Input: Extracted attributes
Output: Optimal attributes

Begin

Initialize DM population, $\Im, \omega, m$, $It_{\max}$

Set $m = m - \omega, P, It = 1$

While $\left(It \le It_{\max}\right)$ **do**

Evaluate DM fitness

Find alpha

Determine food position a_x^{It+1}

Evaluate fitness of a_x^{It+1}

Evaluate sleeping mound with KLD

If $\left(S > P\right)$ **then**

Exchange babysitters

End If

Evaluate next position of DM

$$a_x^{It+1} = \begin{cases} a_x^{It} - cc * \rho * ra * \left(a_x - \vec{Z}\right) if \left(\alpha_{It+1} > \alpha_{It}\right) \\ a_x^{It} + cc * \rho * ra * \left(a_x - \vec{Z}\right) else \end{cases}$$

Update best positions

End While

Return optimal values

End

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

S-HAVAL hashing

To create a hashed access policy for the blockchain, the selected attributes are then hashed utilizing a role-based hashing Successive Hashing Variable Length (S-HAVAL) algorithm. The HAVAL, which converts long message bits into a fingerprint of 128, 160, 192, 224, or 256 bits, is a one-way hashing algorithm. Here, the fingerprints are considered as the transactions in the HLB. The representation of S-HAVAL is given in figure 4,

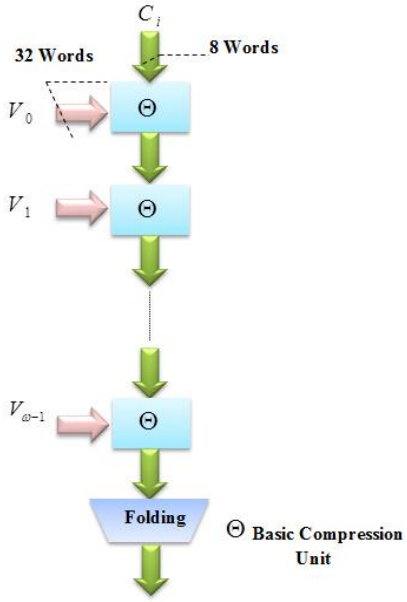

Figure 4: Hashing with S-HAVAL

To compress y_β utilizing the S-HAVAL, the input message y_β extended (padded) to a multiple of 1024 bits. The last block of the padded message comprises the information on the number of bits in the unpadded y_β, the required length of the fingerprint, and the number of passes each input message block is processed. Let the padded blocks can be expressed as $V_{w-1}, V_{w-2}, ..., V_0 \; or V_i$, where each block is a 1024-bit block. The HAVAL starts from V_0 and 256-bit (8-word) consonant string $C_i = C_{0,7}, C_{0,6}, ..., C_{0,0}$, which is taken from the fraction value of Pi, and compress the $V_{w-1}, V_{w-2}, ..., V_0$ by repeatedly computing,

$$C_{i+1} = \Theta(C_i, V_i), i = 0,1,.....w-1 \qquad (17)$$

Where, Θ exemplifies the basic compression function, and the compression operation in HAVAL is expressed further,

The basic compression Θ applies three to five passes, which is signified as $\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5$. The output of $\Theta\ (C_{out})$is determined from the input C_i in the following way,

$$\begin{aligned} N_0 &= C_i \\ N_1 &= \Theta_1(N_0, V_i) \\ N_2 &= \Theta_2(N_1, V_i) \\ N_3 &= \Theta_3(N_2, V_i) \\ N_4 &= \Theta_4(N_3, V_i) \quad if \;\; pass = 4,5 \\ N_5 &= \Theta_5(N_4, V_i) \quad if \;\; pass = 5 \end{aligned} \qquad (18)$$

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

$$C_{out} = \begin{cases} N_3 \oplus N_0 \ if \ \ pass = 3 \\ N_4 \oplus N_0 \ if \ \ pass = 4 \\ N_5 \oplus N_0 \ if \ \ pass = 5 \end{cases} \quad (19)$$

Where, $N_0,...N_5$ are the internal computation parameters, The C_{out} is obtained after the folding technique that gives the desired lengths of fingerprint length, which can be expressed as,

$$\delta = \mathrm{mod}\nabla \sum_{i=0}^{7} C_i \quad (20)$$

Where, δ indicates the folding operation and ∇ signifies the bit length. Moreover, the $\Theta_1,\Theta_2,\Theta_3,\Theta_4,\Theta_5$ performs the round operation of different Boolean functions on the input message block (V_i). Here, five functions $\xi_1(\ ),\xi_2(\ ),\xi_3(\ ),\xi_4(\ ),\xi_5(\ )$ are employed by $\Theta_1,\Theta_2,\Theta_3,\Theta_4,\Theta_5$ to perform a bit-wise operation utilizing the proposed successive technique as,

$$\xi_1(l_6,l_5,l_4,l_2,l_1,l_0) = l_1l_4 \oplus l_3l_2 \oplus l_5l_6 \oplus l_0l_1 \oplus l_0 \quad (21)$$

$$\xi_2(l_6,l_5,l_4,l_2,l_1,l_0) = l_1l_2l_3 \oplus l_3l_4l_1 \oplus l_5l_6 \oplus l_3l_4 \oplus l_1l_2 \oplus l_5l_4 \oplus l_3l_6 \oplus l_0l_2 \oplus l_0 \quad (22)$$

$$\xi_3(l_6,l_5,l_4,l_2,l_1,l_0) = l_1l_2l_3 \oplus l_3l_4 \oplus l_5l_6 \oplus l_3l_4 \oplus l_0l_2 \oplus l_0 \quad (23)$$

$$\xi_4(l_6,l_5,l_4,l_2,l_1,l_0) = l_1l_2l_3 \oplus l_3l_4l_5 \oplus l_5l_6l_1 \oplus l_1l_2 \oplus l_3l_4 \oplus l_5l_6 \oplus l_3l_4 \oplus l_1l_6 \oplus l_5l_2 \oplus l_3l_6 \oplus l_0l_4 + l_0 \quad (24)$$

$$\xi_5(l_6,l_5,l_4,l_2,l_1,l_0) = l_1l_4 \oplus l_3l_6 \oplus l_5l_2 \oplus l_3l_4l_6l_1 \oplus l_0l_5 \oplus l_0 \quad (25)$$

Here, $l_0,l_1.l_2,l_3,l_4,l_5,l_6$ represents the bits belonging to V_i, and $\oplus$ describes the modulo-bit addition. Thus, the hashed value of y_β is represented as,

$$C_{out} = \{h_1h_2....h_{nn}\} \quad (26)$$

Where, nn value can be 128, 160,192,224, or 256 bits. The pseudocode for the S-HAVAL is given as,

---

Input: Selected attributes
Output: hashed value

Begin

Initialize hash length, C_i, V_i, number of passes

For each y_b **do**

Perform padding and get V_i

Perform $\xi_1(\ ),\xi_2(\ ),\xi_3(\ ),\xi_4(\ ),\xi_5(\ )$

Compress V_i by $C_{i+1} = \Theta(C_i,V_i), i = 0,1,.....w-1$

Perform folding

End For

Return C_{out}

End

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Afterward, this hashed value is converted to the Merkle format of the blockchain. In the Merkle tree format, the Merkle root comes from a hashing transaction and pairing two hashing transactions (child nodes) to create the parent tree node (MR), which is expressed as,

$$MR = C_{out}(y_\beta) \| C_{out}(y_\beta + 1) \quad (27)$$

Here, $C_{out}(y_\beta), C_{out}(y_{\beta+1})$ illustrates the fingerprint obtained for corresponding attributes, and $\|$ symbolizes the hashing function of two fingerprints to create a single hash value (Merkle root). The Merkle root MR is stored in the block of the HLB.

Signed access policy creation

After that, grounded on the Mean Public keys- Digital Signature Algorithm (MP-DSA) scheme, the hashed access policy is signed. However, the keys are engendered randomly in the prevailing DSA. In the MP-DSA, using the private key and the public key, the signature is encrypted and decrypted, correspondingly. Utilizing the mean values of the public keys utilized in the QI-BM-DECC algorithm, the key is engendered in the proposed model and utilized for the signing and verification process.

Key generation: Primarily, the keys are generated with the hash function h_r, the key length (Γ), and the chosen two prime numbers pp, pn of different lengths. Moreover, choose an integer $\hat{\lambda}$, from $(2,....,pn-2)$, and the private key (χ) is generated utilizing,

$$\chi = \hat{\lambda}^{(pn-1/pp * \bmod(pn))} \quad (28)$$

After the private key generation, the public key is generated as,

$$\mu = \hat{\chi}(h_r) * \bmod(pn) \quad (29)$$

Where, μ is the public key wielded in the MP-DSA approach,

Key distribution: The key is distributed to the user side after the key generation. During key distribution, the private key is kept a secret by the signer, whereas the public key is shared with the $doctor$ side without any secret mechanisms.

Signature generation: In the signature generation phase, the hashed function h_r is given as input to the signing function, which gives two variables T, K as output. After that, the value of T, K is computed as,

$$T = \left[(rn-1)(\hat{\lambda} + h_r.K) \bmod pp\right] + \sigma_{key} \quad (30)$$

$$K = (\chi.rn. \bmod pn) \quad (31)$$

Where, rn specifies a random integer, such that $0 < rn < pp$. Also, the key value generated with the mean public keys in the QI-BM-DECC algorithm utilized to sign the signature as,

$$\sigma_{key} = \frac{P_{patient} - P_{doctor}}{2} \quad (32)$$

Then, the signature package is specified as $\{T, K\}$. After that, the message and signature are sent to the $doctor$ side as $\{E, T, K\}$.

F. Downloading EHR

When the doctor requests to access the EHR stored in the HLB, the signature is verified utilizing the MP-DSA approach. After successful verification, the EHR will be decrypted and downloaded.

Signature verification: The $doctor$ side employs a hash value, and gives it to the verification function. During verification, the other variables were taken as parameters, which can be computed as,

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

A parameter j is estimated such that,

$$T * j \bmod pp - \sigma_{key} = 1 \quad (33)$$

Moreover, other two parameters (φ_1, φ_2) are estimated, which are expressed as,

$$\varphi_1 = \lambda * j \bmod pp \quad (34)$$

$$\varphi_2 = K * j \bmod pp \quad (35)$$

Lastly, from the estimated parameters, the final verification is performed utilizing the verification component (ψ) as,

$$\psi = \left\lfloor \left(\chi . \varphi_1 \times \sigma_{key} . \varphi_2 \right) \bmod pn \right\rfloor \bmod pp \quad (36)$$

If the ψ matches the signed signature, the transaction is verified by the HLB and $doctor$ was eligible to access the EHR in the HLB. After that, the access request is sent to the patient. The patient then logs in to the network and accepts the request, after that only, the $doctor$ will be able to download the file. If the patient did not accept, the EHR could not be downloaded.

Decryption: Next, on the $doctor$'s side, the encrypted data will be decrypted by utilizing the receiver's second private key with a secret key and again it will be decrypted by utilizing the receiver's first private key with a secret key wielded in the QI-BM-DECC encryption. Here, the secret key is divided by the decryption formula.

$$E_{partial} = \frac{sk}{cip_4 - \varpi_2 * cip_3} \quad (37)$$

$$E = \frac{sk}{cip_2 - \varpi_2 * cip_1} \quad (38)$$

Here, $E_{partial}$ specifies the partially decrypted EHR, and ϖ_1, ϖ_2 elucidated the first and the second private keys in the corresponding runs. Hence, double encryption and decryption mounted the EHR's security.

G. Attack detection

In the proposed model, after implementing blockchain for securing EHRs, it can detect and prevent the Double Spending (DS) attack, which is explicated as follows. After the EHR was shared successfully, if the patient tries to share the same EHR file with a different doctor or sends the same file multiple times, it is considered as a DS attack. If the DS attack was detected, the proposed model displayed "sharing multiple times is not allowed".

4. Results and Discussions

Here, the proposed EHR sharing system's outcomes are comparatively examined and discussed. In the working platform of Python, the experiments were executed.

A. Dataset description

In the proposed system, the EHR shared is the publically available Poly Cystic Ovary Syndrome (PCOS) dataset. For determining PCOS along with infertility-related problems, the dataset comprises all the clinical and physical parameters. The parameters of 541 patients are encompassed in the dataset; also, the data in the dataset are gathered from 10 hospitals across Kerala, a state in India.

B. Performance analysis

In this, the proposed system's performance is examined in 2 segments, such as attribute selection and data security.

a. Performance analysis of attribute selection

Here, regarding fitness values along with attributes selection time, the proposed attribute selection algorithm KLD-DMOA's performance is comparatively examined with the conventional attribute selection algorithms, namely Emperor Penguin Optimization Algorithm (EPOA), Locust Swarm Optimization Algorithm (LSOA), Sooty Tern Optimization Algorithm (STOA), and DMOA.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

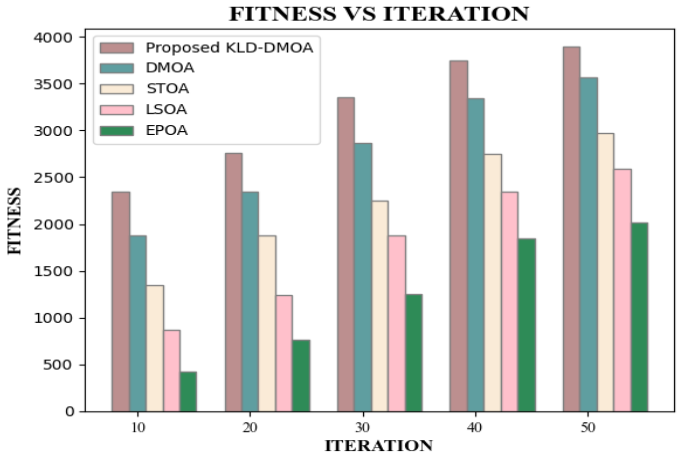

Figure 5: Fitness vs iteration analysis

How close the chosen attribute is to the best role-centric attribute in each iteration was evaluated by the fitness function. Figure 5 displays that the fitness of the proposed KLD-DMOA increases from 2347-3892 when the number of iterations increased from 10-50. Although the fitness of prevailing DMOA, STOA, LSOA, and EPOA approaches increased, still they could not reach the proposed KLD-DMOA fitness. Hence, KLD-DMOA has enhanced fitness value displays that the most significant role-centric attributes are selected with KLD-DMOA over the other approaches.

Table 1: Comparative analysis of attribute selection time

Algorithms	**Attribute selection Time (ms)**
Proposed KLD-DMOA	1784
DMOA	2365
STOA	2865
LSOA	3365
EPOA	3865

In table 1, the attribute selection time of the proposed along with the prevailing attribute selection algorithms are depicted. For the proposed KLD-DMOA, the attribute selection time is 1784ms, which is 601ms, 1081ms, 1581ms, and 2081ms faster when analogized with the baseline DMOA, STOA, LSOA, and EPOA, correspondingly. Hence, utilizing the proposed attribute selection algorithm, only less time is taken to select the optimal features.

b. Performance analysis of data security

Here, centered on memory usage and time during the encryption and decryption, key generation time, throughput, and average latency concerning throughput, the proposed data security algorithm QI-BM-DECC's performance is examined in comparison with the baseline ECC, RSA, Diffie Hellman, and ElGamal algorithms.

Table 2: Memory usage during encryption and decryption

Algorithms	**Memory usage on encryption (kB)**	**Memory usage on decryption (kB)**
Proposed QI-BM-DECC	138767324	139568321
ECC	157092164	160274363
RSA	183470231	189542701

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Diffie Hellman	201894326	225698214
ElGamal	249654214	248569021

The CPU memory wielded by the proposed QI-BM-DECC algorithm and the conventional cryptographic techniques, namely ECC, RSA, Diffie Hellman, and ElGamal during the encryption and decryption of EHR is illustrated in table 2. When analogized with the existing ECC, RSA, Diffie Hellman, and ElGamal techniques, the memory utilized by the proposed algorithm during encryption and decryption is of less storage, which is 138767324kB and 139568321kB, respectively. Hence, the proposed system obtains less storage when compared to other algorithms.

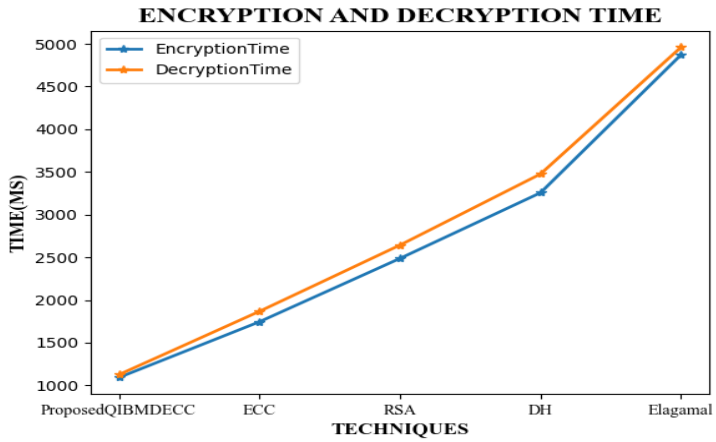

Figure 6: Time taken during encryption and decryption of EHR

In figure 6, the graphical representation of the time taken to encrypt and decrypt the PCOS dataset is displayed. It shows that to encrypt and decrypt the PCOS dataset, the proposed algorithm takes less time. Here, when contrasted with traditional approaches, the proposed QI-BM-DECC algorithm takes a lower encryption time of 1092ms for the PCOS dataset. In addition, QI-BM-DECC obtains 741ms, 1517ms, and 3835ms lesser than the ECC, RSA, and ElGamal techniques, respectively. Thus, for the proposed EHR encryption and decryption, the QI-BM-DECC approach is made more suitable.

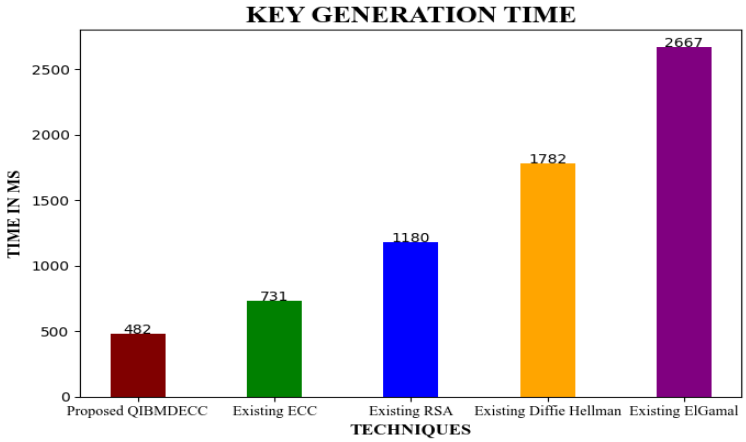

Figure 7: Analysis of key generation time

In figure 7, the time taken to generate the keys (public, private, and secret keys) utilized in the proposed technique utilizing proposed and prevailing techniques is elucidated. The figure exhibits that when analogized with RSA, DH, and ElGamal techniques, the ECC algorithm generates the key in less amount of time. However, key generation time is diminished by 249 ms with the utilization of BM and QI in the ECC algorithm. Hence, it is concluded that when compared with the conventional techniques, the keys are generated in less time with the proposed QI-BM-DECC technique.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

Table 3: Security level of the proposed QI-BM-DECC

Algorithms	**Security level (%)**
Proposed QI-BM-DECC	97.26
ECC	94.76
RSA	91.04
Diffie Hellman	87.34
ElGamal	85.82

Table 3 displays the security level of the proposed along with the existing algorithms. It also exhibits that when analogized with existing ECC, RSA, Diffie Hellman, and ElGamal algorithms, the proposed QI-BM-DECC system is 2.63%, 6.83%, 11.35%, and 13.33% more secure, correspondingly. Hence, the proposed system is more secure against attacks.

c. Performance analysis of the proposed framework

This section examines the overall performance of the proposed Role-Based Hashed Access Policy HLB (RBHAP-HLB) for EHR sharing regarding time, throughput, along with latency.

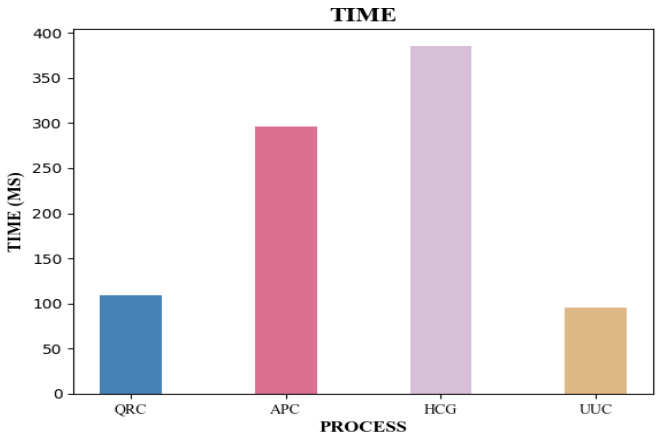

Figure 8: Time taken for QRC, APC, HCG, and UUC

The pictorial representation of the time consumed during Hash Code Generation (HCG), APC, UUID Creation (UUC), and QR code Creation (QRC) is exhibited in figure 8. In the proposed technique, the QRC, APC, HCG, and UUC processes are completed in 109ms, 296ms, 385ms, and 96ms. Thus, HCG, APC, UUC, and QRC processes in the proposed technique are executed in less time.

Table 4: Throughput analysis of the proposed model

Number of users	**Throughput (kBPS)**
100	2392
200	2591
300	2876
400	3028
500	3447

The amount of data transferred from one place to another in the unit of time is the throughput. Here, to analyze the throughput in kilobits/sec, 100-500 users are initialized to perform various activities like uploading and retrieving EHRs, which is depicted in table 4. The table shows that the throughput elevates from 2392kBPS to 3447kBPS as the number of users elevates. This enhancement in throughput displays that more amount of data can be shared through the proposed system.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

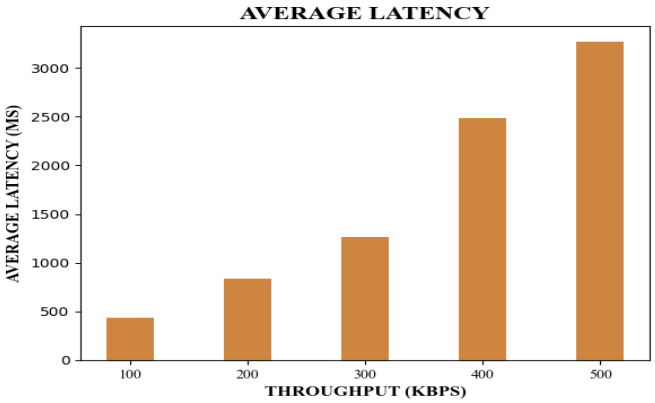

Figure 9: Analysis of the average latency

The delay occurring betwixt when the number of bits is transferred and when the response is attained is named latency. Figure 9 shows that for the proposed work, when 100kBdata is transferred, the lowest latency of 437ms transpired and when 500kB data is transferred, the highest latency of 3266ms was achieved. Hence, in the proposed system, only less delay occurred during the EHR sharing.

C. Comparative analysis with the related research

Here, regarding privacy level, the proposed RBHAP-HLB framework techniques are analogized with the existing studies of [11], [12], and [13].

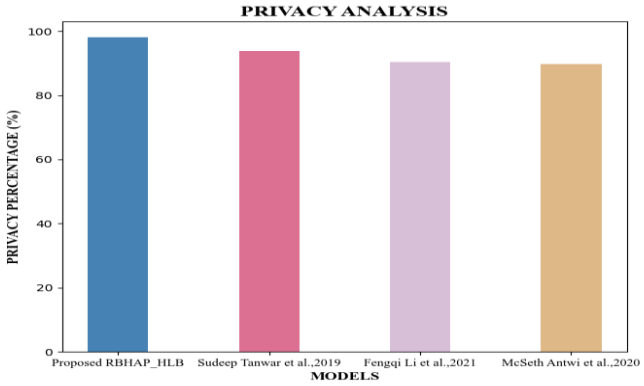

Figure 10: Privacy level comparison with the existing works

In figure 10, the privacy analysis of the proposed framework along with the recent related techniques is represented. Although [13] and [11] utilized HLB for EHR sharing, they do not utilize a secure access policy. Hence, less privacy was attained. Even though [12] have utilized a changed access policy, HLB-based IPFS storage, still it attained 7.83% less privacy than the proposed RBHAP-HLB framework. This shows that more privacy can be preserved with the proposed model than with the prevailing techniques.

5. Conclusion

This paper proposes a novel QI-BM-ECC-based secure EHR sharing utilizing HLB with IPFS. Here, the EHR was secured with the QI-BM-DECC algorithm, and an attribute-centric access policy was signed with the MP-DSA approach for creating a signed access policy on the HLB. By utilizing the PCOS dataset as EHR, the proposed mechanism's performance is assessed. The experiential outcomes exhibited that the proposed technique outperformed the other conventional mechanisms. By achieving a security level of 97.26%, the security of the proposed framework is proven. Moreover, the access policy in the HLB is created in 296ms. These analyses display the proposed model's overall superiority for secure EHR sharing in the IPFS grounded on HLB security. This work proposed secure EHR access between the hospital staff and patient only, but not focused on access of EHRs from other legal institutions, such as insurance agencies. Thus, in the future, along with the proposed model, the secure access policy for authenticated other institutions can be introduced.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

References

[1] M. A. Saberi, M. Adda, and H. McHeick, "Break-Glass Conceptual Model for Distributed EHR Management System Based on Blockchain, IPFS and ABAC," *Procedia Computer Science*, vol. 198, pp. 185–192, 2021, doi: 10.1016/j.procs.2021.12.227.

[2] Y. Sharma and B. Balamurugan, "Preserving the Privacy of Electronic Health Records Using Blockchain," *Procedia Computer Science*, vol. 173, pp. 171–180, 2020, doi: 10.1016/j.procs.2020.06.021.

[3] J. Oh et al., "A Secure Personal Health Record Sharing System with Key Aggregate Dynamic Searchable Encryption," *Electronics*, vol. 11, no. 19, pp. 1–24, 2022, doi: 10.3390/electronics11193199.

[4] A. Ghani, A. Zinedine, and M. El Mohajir, "A Blockchain-Based Secure PHR Data Storage and Sharing Framework," in *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, 2020, pp. 162–166.

[5] S. S. R. Krishnan et al., "A Blockchain-Based Credibility Scoring Framework for Electronic Medical Records," in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6, doi: 10.1109/GCWkshps50303.2020.9367459.

[6] G. Subathra, A. Antonidoss, and B. K. Singh, "Decentralized Consensus Blockchain and IPFS-Based Data Aggregation for Efficient Data Storage Scheme," *Security and Communication Networks*, vol. 2022, pp. 1–13, 2022, doi: 10.1155/2022/3167958.

[7] A. Mukherji and N. Ganguli, "Efficient and Scalable Electronic Health Record Management Using Permissioned Blockchain Technology," in *2020 4th International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, 2020, pp. 1–6, doi: 10.1109/IEMENTech51367.2020.9270106.

[8] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain and Edge Computing for Decentralized EMRs Sharing in Federated Healthcare," in *2020 IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9347951.

[9] S. Vardhini, S. N. Dass, Sahana, and R. Chinnaiyan, "A Blockchain-Based Electronic Medical Health Records Framework Using Smart Contracts," in *2021 International Conference on Computer, Communication, and Informatics (ICCCI)*, 2021, pp. 27–30, doi: 10.1109/ICCCI50826.2021.9402689.

[10] W. Zhan et al., "Incentive EMR Sharing System Based on Consortium Blockchain and IPFS," *Healthcare*, vol. 10, no. 10, pp. 1–27, 2022, doi: 10.3390/healthcare10101840.

[11] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-Based Electronic Healthcare Record System for Healthcare 4.0 Applications," *Journal of Information Security and Applications*, vol. 50, p. 102407, 2020, doi: 10.1016/j.jisa.2019.102407.

[12] F. Li et al., "EHRChain: A Blockchain-Based EHR System Using Attribute-Based and Homomorphic Cryptosystem," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2755–2765, 2022, doi: 10.1109/TSC.2021.3078119.

[13] M. Antwi et al., "The Case of HyperLedger Fabric as a Blockchain Solution for Healthcare Applications," *Blockchain Research and Applications*, vol. 2, no. 1, p. 100012, 2021, doi: 10.1016/j.bcra.2021.100012.

[14] K. Shuaib, J. Abdella, F. Sallabi, and M. A. Serhani, "Secure Decentralized Electronic Health Records Sharing System Based on Blockchains," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5045–5058, 2022, doi: 10.1016/j.jksuci.2021.05.002.

[15] J. Kaur, R. Rani, and N. Kalra, "Blockchain-Based Framework for Secured Storage, Sharing, and Querying of Electronic Healthcare Records," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 20, pp. 1–24, 2021, doi: 10.1002/cpe.6369.

[16] V. Mani et al., "Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records," *Electronics*, vol. 10, no. 23, pp. 1–23, 2021, doi: 10.3390/electronics10233003.

[17] R. K. Marangappanavar and K. Kiran, "Inter-Planetary File System Enabled Blockchain Solution for Securing Healthcare Records," in *2020 3rd ISEA International Conference on Security and Privacy (ISEA-ISAP)*, 2020, pp. 171–178, doi: 10.1109/ISEA-ISAP49340.2020.235016.

[18] Z. Sun et al., "A Blockchain-Based Secure Storage Scheme for Medical Information," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, pp. 1–25, 2022, doi: 10.1186/s13638-022-02122-6.

[19] K. Saravanan et al., "WMLP: Web-Based Multi-Layer Protocols for Emergency Data Transmission in Mobile Ad Hoc Network," in *International Conference of Computer Sciences and Renewable Energies (ICCSRE)*, Agadir, Morocco, July 23–24, 2021, doi: 10.1051/e3sconf/202129701065.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025

[20] S. Kumarganesh et al., "A Novel Analytical Framework Developed for Wireless Heterogeneous Networks for Video Streaming Applications," *Journal of Mathematics*, vol. 2022, no. 1, pp. 1–7, 2022, doi: 10.1155/2022/2100883.

[21] K. Saravanan et al., "Power Adjustment Algorithm for Higher Throughput in Mobile Ad Hoc Networks," in *International Conference of Computer Sciences and Renewable Energies (ICCSRE)*, Agadir, Morocco, July 23–24, 2021, doi: 10.1051/e3sconf/202129701064.

[22] N. Sugirtham et al., "Modified Playfair for Text File Encryption and Meticulous Decryption with Arbitrary Fillers by Septenary Quadrate Pattern," *International Journal of Networked and Distributed Computing*, vol. 12, pp. 108–118, 2024, doi: 10.1007/s44227-023-00019-4.

[23] K. Baskar, K. Muthumanickam, P. Vijayalakshmi, et al., "A Strong Password Manager Using Multiple Encryption Techniques," *Journal of The Institution of Engineers (India): Series B*, 2024, doi: 10.1007/s40031-024-01144-6.

DOI: https://doi.org/10.54216/FPA.180214
Received: July 19, 2024 Revised: October 18, 2024 Accepted: January 12, 2025