Cluster Computing (2023) 26:2217–2244
https://doi.org/10.1007/s10586-022-03652-w

MediBlocks: secure exchanging of electronic health records (EHRs) using trust-based blockchain network with privacy concerns

Erukala Suresh Babu[1] · B. V. Ram Naresh Yadav[2] · A. Kousar Nikhath[3] · Soumya Ranjan Nayak[4] · Waleed Alnumay[5]

Received: 25 November 2021 / Revised: 7 May 2022 / Accepted: 17 June 2022 / Published online: 17 August 2022

Abstract

Healthcare is a sensitive domain, and it is necessary to design an efficient solution in the healthcare system. Electronic Medical Records often have high privacy-sensitive data; unauthorized access to these patient data could hurt patients' reputations and cause financial losses to the organization. Most of the existing healthcare systems are based on a centralized architecture, which is more vulnerable such as hacking the healthcare system, arbitrary modification attacks, and single-point-of-failure. Also, these systems increase the cost due to various factors like—establishing a trusted network, cost per transaction, and limited access to patient health data, which are the problems that remain open until now. The blockchain is a general-purpose technology that seems to be an innovative and interesting technology that can improve the Health care domain by solving the above issues with a secure architecture. This paper address addresses the challenging problems. (1) How to provide secure data exchange and anonymity. (2) How to preserve the personal data privacy of the patient health records. The proposed solution leverages the Hyperledger Fabric, a permissioned blockchain that establishes a secured and trusted network to all stakeholders, ensuring the integrity of protected health information and providing authenticity and health access control. Further, the entire supply chain can be traced, decreasing duplicate tests and unnecessary service, increasing accountability, preserving the crucial documents, limiting the unauthorized sharing of the EHRs documents, and offering lower costs across the care continuum.

Keywords Health records · Security · Privacy · Blockchain

1 Introduction

In the last decade, a tremendous improvement in Information Technology (IT) brought many benefits to the healthcare area. The advancement of IT has produced large databases for digital storage of patients' health records and tools for tracking the health data for disease prevention. These advancements in IT and health care would nurture transformation in health IT [1]. However, this transformative growth comes with extreme pressure to regulate costs while providing high quality-of-care to the patients. Building a safer health system can facilitate quality of care

✉ Soumya Ranjan Nayak
nayak.soumya17@gmail.com

Erukala Suresh Babu
esbabu@nitw.ac.in

B. V. Ram Naresh Yadav
bvramnaresh@gmail.com

A. Kousar Nikhath
kousarnikhath@vnrvjiet.in

Waleed Alnumay
wnumay@ksu.edu.sa

[1] Department of Computer Science and Engineering, NIT Warangal, Hanamkonda, Telangana, India

[2] Department of CSE, JNTUH Hyderabad, Hyderabad, Telangana, India

[3] Department of CSE, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, Hyderabad, Telangana, India

[4] Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India

[5] Riyadh Community College, CS Department, King Saud University, Riyadh, Saudi Arabia

by easily and timely retrieving accurate important information for better diagnosis, easy navigation, clinical reasoning, decision-making, and efficient communication among patients and health care professionals. However, the diagnosis in the health care system is complex, requires a collaborative effort, and often involves Intra and inter-professional teamwork, as well as with patients and their families. Electronic Health Records (EHR) help us build a safer and quality health care system [2, 3] that presents the opportunity to share completely accurate information to and between healthcare providers. The EHR is the systematic collection of clinical care longitudinal patient information (e.g., medical notes in the form of prescriptions, lab tests, results, procedures, medication, imaging, and the diagnosis, etc.), which potentially not only improves the quality of care through the quantity of clinical data but also improves the diagnostic performance and reduces the diagnostic errors. These EHRs provide various advantages such as Integration with the clinical decision support system, access to knowledge bases, data aggregation, effective communication with patients, improved communication between providers, support for multiple views, better analyses and diagnosis, and accessibility from various locations. However, healthcare organizations use third-party providers to use cloud computing for EHRs and Personal Health Record (PHR) services due to the high cost of data centers. While healthcare providers maintain the EHRs, PHRs are preserved by the patients themselves, who can manage their records and usually have no control over the data stored in healthcare providers' databases. Furthermore, the health organizations store all patients' health records in databases that will have restricted access to internal health professionals. In many cases, healthcare providers/ organizations do not share their patients' data because of privacy concerns and are unwilling to send the information, which gives others a competitive advantage. Thus, EHRs lack trust in technologies usage, which results in a lack of utilization of a large amount of high-dimensionality distinct medical data among healthcare organizations; the collected records may be systematically misleading and contradictory, and EHR data is missing largely in the form lack of visits, patient relocation, and recording mistakes, etc.

In current public health information technology systems, EHRs often have high privacy-sensitive data [4, 5]. Unauthorized access to these patient data could hurt patients' reputations and cause financial losses to the organization. Moreover, most existing healthcare systems are more vulnerable such as hacking the healthcare system, arbitrary modification attacks, and single-point-of-failure. In addition, these systems also increase the cost due to the various factors like—establishing a trusted network, cost per transaction, limited access to patient health data, and varying data standards-interoperability, which are the problems that remain open until now. Some of the challenging issues need to be addressed. (1) How to preserve personal data privacy. (2) How to provide open access to sensible health data and successful healthcare data exchange (Interoperability). (3) How to ensure anonymity and avoid data being misused. Specifically, securing the electronic medical record is challenging; it fails to secure the patient's health record, has legal and financial consequences, and impacts patient care. Secure sharing of healthcare data [6, 7] needs to be enabled between healthcare providers, researchers, and patients. Moreover, Patients will also have complete control of their health data. Furthermore, privacy concerns also need to be addressed because patient data is also an asset to the institution, and patient privacy is not only an ethical responsibility. But a legal mandate which impacts any healthcare system. Hence, Healthcare is a robust domain. Changing the existing system is complicated, particularly relevant to conflicts between the accessibility of data, privacy, and security are natural to occur. But there is a growing demand for healthcare services that need to address information technology solutions to reduce costs, interoperability and provide security & privacy. The blockchain [8] is a general-purpose technology that seems to be an innovative and interesting technology, which has the potential to improve the health care domain by solving the above issues with a decentralized architecture. This paper address the privacy and secure data exchange issue on Electronic Health Records (EHRs) [9, 10] using a blockchain [11, 12] shared platform that ensures the integrity of protected health information, authenticity, health data ensuring access control across systems and organizations, which decrease duplicate tests, unnecessary service, and offers the lower costs across the care continuum.

Our Contribution This paper address addresses the following challenging issues. (1) Challenge-1: How to provide secure data exchange and anonymity. Issue: Securing the electronic medical record is challenging; failing to secure the patient health record has legal and financial consequences and impacts patient care. Secure sharing of healthcare data needs to be enabled between healthcare providers, researchers, and patients. Moreover, Patients should also have complete control of their health data. (2) Challenge-2: How to preserve the personal data privacy of the Patient Health Record. Issue: The privacy concerns need to be addressed because patient data is also an asset to the institution, and patient privacy is an ethical responsibility and a legal mandate. To address these issues, the following objectives were proposed:

- We proposed a permissioned-based blockchain framework for securely exchanging information that provides a high assurance and guarantees the integrity of data provenance. The proposed system uses an ECDSA cryptosystem in a healthcare blockchain network that effectively allows the nodes to interact anonymously and securely to share healthcare information within a data-sharing network. In addition, this healthcare system is a tamperproof database that secures all the health records and is added to it and replicated across a collection of nodes connected as a peer-to-peer network. Each health record is a unique event that contains a timestamp and assigns a hash of the record created by a cryptographic hash algorithm.
- Data privacy issues are the main impact of any healthcare system; we integrated the privacy-preserving framework in the proposed healthcare blockchain system, which uses the Online/Offline framework. This framework is mainly used for two purposes: to speed up the process of verification and to secure health record storage. Specifically, The proposed blockchain system can store two types of information. (1) "Online-Chain" data is directly stored on the blockchain network since online-chain does not require any heavy computations and stores less record information that speeds up the block process. (2) "Offline-Chain" data are stored in local databases, and the links/index of offline data are stored on the blockchain. Hence, the Online/Offline framework provide authenticity, fine-grained data access control with confidentiality (i.e., all health records of all the customer remain fully confidential), signer anonymity, and public verifiability. In addition, this mechanism is mainly designed to prove ownership of customers/ health providers with a full public key on the healthcare blockchain, which prevents information exchanges to unauthorized users or health providers and does not disclose any patient information about its customers.

Organization of the Paper: The rest of the paper is organized as follows. The related work is discussed in Sect. 2. The Sect. 3 presents the preliminaries, building blocks, System Design, and proposed system framework. Section 4 describes System Setup and Performance Analysis. The conclusion is discussed in Sect. 6.

2 Related work

This section presents the related work of many researchers who contributed significantly to various mechanisms and approaches to secure EHRs in the eHealth care domain. Recently, Governments and Healthcare authorities have been undertaking several initiatives to boost the health IT system. Electronic Health Records are one such initiative, and the next step is to ensure a seamless flow of health information across stakeholders that will enable better decision-making. The ability of health IT systems to share this information and use that information for better decision making. However, the absence of interoperability creates several issues like restricted data sharing, and the unavailability of complete data restricts the lack of meaningful insights, i.e., the benefits of data mining. Several challenges have to be faced while enabling EHR Interoperability, like data security and integrity, government regulations, etc. In [13], Mike Miliard presented a blockchain-based EHR that could unlock the conditions of interoperability and security issues. This paper mainly focuses on interoperability, security, and smart contracts in healthcare systems. They discussed three models of interoperability among medical data. One is the Push model is used for sending health information to providers. Second, the Pull model is used to query information from a provider. The third is the View model, which can be used to see the information inside another's record. In [14], the authors presented the Ancile framework and its unique type of smart contracts to operate the EHRs using the Blockchain System. Ancile contains three principal units: Ethereum-go client, database manager, and cipher manager. Framework architecture for several operations and the steps involved in it are mentioned. Ancile uses Ethereum tools for making a system that uses blockchain technology and is cost-effective.

In [15], Elmisery et al. proposed an open platform to provide smart health services for managing chronic diseases using the distributed object group framework on PHR. The author showed the smart health network environment makes performance of distributed network is very efficient distributed agent system can be built. In [16] Li et al., discussed the privacy challenges in the IoT and how blockchain will provide the solutions by examining 61 papers systematically. Further, the authors showed that the blockchain could dominate the IoT restrictions by providing a beneficial guarantee for the privacy and security of IoT users. In [17] Liang et al., provide a systematic study of privacy and security in blockchain systems suitable to solve image retrieval problems, cancer datasets and patient healthcare records, and financial information in IoT applications and smart environments. In [18] Kanwal et al., discussed the privacy preservation models and techniques on EHR and presented the most relevant privacy techniques deployed in the cloud environment.

In [19], Holbl et al. provided various fundamental concepts of blockchain, consensus mechanisms, how blockchain was introduced to the public and reviewed research questions and selected studies and references. Several

research questions are discussed and answered. The fundamental tasks in the system involve getting connected to the network, updating the ledger, verifying transactions, forwarding only valid transactions to the network, and making new blocks whenever required. The data in health records need not be modified or deleted; the immutable property of blockchain satisfies this condition. In [20], da Conceicao et al. presented how smart contracts help deal with data integrity, management, and interoperability. The paper focuses mainly on data privacy, which gives ideas about who is authorized to access data and data accessibility, showing the extent to which data is available. In [21], Theodouli, Anastasia, et al. proposed blockchain technology ensures secure data sharing without intermediaries. This paper provides information about how smart contracts are used in blockchain technology to share data securely. The smart contracts consist of a logic that provides accessibility to patients' health data to the right person at the right place by ensuring safe and secure sharing of patients' health records. In [22], Alexaki et al. presented a Smart Contracts-based blockchain that will be used to ensure privacy and integrity for medical data. The process of achieving privacy and security [23] is presented using blockchain that has the specific feature of providing a decentralized system, in which recorded transactions cannot be changed, Capability to recover from difficult situations, thus not affecting the stored data and unauthorized access to health information that is stored. In [24], Katuwal et al. addressed important use cases of blockchain technology in the health care sector like management of health information, research, management of prescriptions provided by doctors, billings are reviewed, and their proposed work eliminates the need for a third party. However, most of the existing proposed Electronic Healthcare Records (EHRs) mechanism is based on the public permissionless blockchain network [25]. The following Table 1 summarizes the related and proposed work of EHRs using a blockchain system.

3 Overview of the proposed system

The proposed system is a permissioned blockchain network that acts as the trust infrastructure by authenticating the users /actors and ensuring trust between various actors like-patients, doctors, pharmaceuticals, researchers, and healthcare organizations. Nevertheless, also validate the user's identity with a high level of security. The proposed e-Healthcare application is implemented in Hyperledger fabric [26] that makes use of the Elliptic Curve Digital Signature Algorithm (ECDSA) mechanism [27, 28] for the security and privacy of healthcare records over public networks. To secure the patient health record, every member of the blockchain network must follow the

Table 1 Summary of related work of EHRs using a blockchain system

S.No	Title	Identity Management	Secure Exchange	User Authentication	Integrity	Privacy	Secure Storage	Scalability
1	Alexaki, Sofia, et al. [1]	✗	✗	✓	✓	✓	✓	✗
2	Anastasia Theodouli.et.al [2]	✗	✓	✓	✗	✗	✗	✗
3	Gajendra J. Katuwal.et.al [4]	✗	✓	✓	✗	✗	✗	✗
4	da Conceição et al.[10]	✗	✗	✗	✓	✓	✗	✗
5	Mike Miliard.et.al [25]	✗	✓	✓	✓	✗	✗	✗
6	Gaby G. Dagher et al. [26]	✗	✓	✗	✗	✗	✓	✗
7	Nitish Mittal.et.al [27]	✗	✗	✓	✓	✗	✗	✗
8	B Nguyen.et.al [28]	✓	✗	✓	✓	✗	✗	✗
9	A Shahnaz. et al. [29]	✗	✗	✗	✗	✓	✓	✓
10	Qiuli Qin.et.al [30]	✗	✓	✓	✓	✓	✓	✗
11	RJ Krawiec.et.al [31]	✗	✓	✓	✓	✗	✗	✗
12	**Our Contribution**	✓	✓	✓	✓	✓	✓	✓

endorsement policy. Any unauthorized access to information can be restricted using security measures and services. The proposed work uses blockchain servers that use Transport Layer Security (TLS) protection to prevent individuals from accessing any information they are not authorized to access the data. The client data will be encrypted using his private key, generated at registration to the blockchain network using the Elliptic Curve Digital Signature Algorithm (ECDSA). To ensure the privacy of stored health records, all the members of this proposed blockchain network must follow the procedures related to privacy for protecting personal EHR data. The proposed architecture uses two types of storage to ensure the data's privacy. One is online storage, and the other is offline storage. Online storage refers to the ledger of the blockchain network that stores the personal data about the clients and hash values of encrypted records and transactions. The indexed key of the corresponding user data is stored in

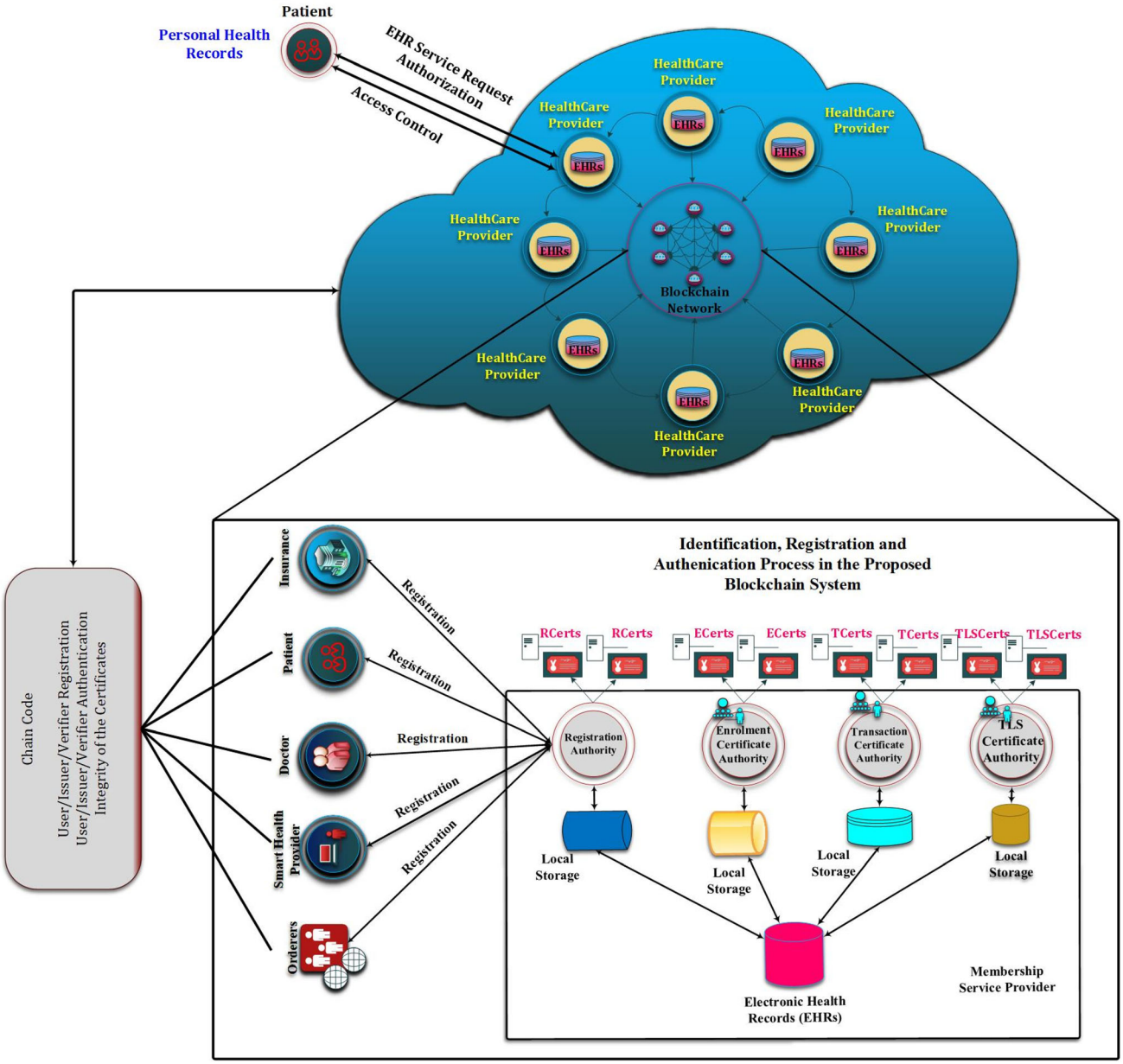


Fig. 1 Proposed framework of secure data exchange of EHRs and privacy-preserving using permissioned blockchain system

offline storage. While offline storage is mainly used to store the actual records of patients in encrypted forms as key-value pairs and the indexed key of corresponding data is available in online storage. The Fig. 1 shows the proposed Framework of Secure data exchange of EHRs and privacy-preserving using a permissioned blockchain system [29–31]

3.1 Background

This section presents the background of the proposed work. The digital healthcare marketplace has recently experienced transformative growth in the information technology sector. But there is a need to design a solution in current health information technology that supports secure collaboration among health care providers, secure sharing of health data, accountability, and auditing, ensuring the authenticity of health data and user data, immutability, integrity, and trust. Currently, the healthcare industry is used to manually storing medical records, which are not efficient, secure, organized, tamperproof, and redundant. Many hospitals worldwide have used the EHR systems that include complete information ranging from patients' appointment management to billing and lab tests. In n this paper, we suggest decentralized Blockchain-enabled health IT systems change the current health information technology and gain greater efficiency in private and public health care systems.

Characteristics of blockchain technology

(a) Transparency: Any new transaction can be added to the ledger only when most nodes accept it as valid, ensuring transparency.
(b) Enhanced Security: The need for central authority is eliminated. Security can be enhanced since there is no possibility that only one person can change any network characteristics for their benefit.
(c) Decentralized technology: It is decentralized because there will not be any central authority that governs the entire network.
(d) Consensus: Blockchain technology has consensus mechanisms that include algorithms that help the network make decisions.
(e) Distributed ledgers: Ledgers will be present with all the system users distributing computational power across the computers to provide a good outcome.

The proposed method can provide technological solutions that help address the above challenging issues such as security and privacy of user-owned data, integrity, and health data interoperability. Blockchain is a digital ledger technology that can potentially transform the health care system. Specifically, it enables the longitudinal health care records to solve the health data interoperability issues for the specific healthcare domain, online patient access, dramatically improves data accuracy, provides privacy and security, and reduces maintenance costs. Cybersecurity in

Table 2 Key terms used in the proposed work

Abbreviation	Purpose
BID_{Issuer}	Blockchain identifier of issuer
K_{Pu}^{Issuer}	The public key of the issuer
K_{Pr}^{Issuer}	The private key of the issuer
$Sign_{Pr}^{Issuer}$	Signature of the issuer
$BID_{Receiver}$	Blockchain identifier of receiver
$K_{Pu}^{Receiver}$	The public key of the receiver
$K_{Pr}^{Receiver}$	The private key of the receiver
EHR-ID	Identifier of requested electronic health record
S_{Key}	Symmetric key of patient
UID	Unique identification number (Aadhaar, SSN, etc..)
RCert	Certificate issued by registration certificate authority
Enrollment CA	Enrollment certificate authority
ECert	Certificate issued by enrollment certificate authority
ECAcert	Certificate of enrollment CA used for verification
TLS-CA	Transport layer security certificate authority
TLScert	Certificate issued by TLS-CA
TLSCAcert	Certificate of TLS-CA used for verification
Transaction CA	Transaction certificate authority
Tcert	Certificate issued by transaction certificate authority
TCAcert	Certificate of transaction CA used for verification

the IT sector faces significant threats- ransomware, Distributed Denial of service (DDOS) attacks, etc., which affect the healthcare industry. The blockchain is a highly disruptive technology that can significantly enhance cybersecurity features- like enabling secure data exchange, immutability via File Integrity, enable peer-to-peer interoperability via Collaborative Version Control among participants within transactions, and Data Access Management.

3.2 Preliminaries and building blocks

The section discusses cryptographic and distributed consensus protocol preliminaries used in the proposed healthcare system. The hyper ledger fabric is a permissioned blockchain system that uses Elliptic Curve Digital Signature Algorithm (ECDSA). This ECDSA [28] is a public key cryptographic algorithm the proposed blockchain network uses to ensure that EHRs information will only be available to the rightful owners. The ECDSA algorithm uses the private key and public key. A private key is a secret number known to the user, who has generated it, and the public key is also a number but does not need to be kept secret and corresponds to a private key. Some of the following Notation and Conventions used in the proposed work are shown in the Table 2.

(A) ECDSA Signature Scheme To describe the ECDSA in the abstract form, Consider an elliptic curve $E: y^2 + ax + b$ over the finite field Z_p, Where $a, b \in Z_p$ and $4a^3 + 27b^2 \neq 0$ of integers modulo a prime 'p'and let cyclic group G which is a generator of prime order 'q' where $G \in E(Z_p)$; The ECDSA algorithm uses the hash function $H: M \rightarrow Z_p$ that used for embedding messages into the field with 'p' ECDSA is an Elliptic Curve Digital Signature Scheme which has triple tuples of efficient randomized algorithms (Key-Gen, Sign, Verify) such that.

1. ***Key-Gen Algorithm*** (1^k) **:** ***Key-Gen*** is the Key Generator algorithm that takes the input as security parameter $Key - Gen(1^k)$ and produces the output as key pair (P, k) Where P is the public key and k is the secret key of the signature scheme. The proposed application uses Koblitz elliptic curve $E: y^2 + x^3 + 7$ called as secp256k1 (P256); which has special properties to perform the group operation very efficiently, Where
   - $p = \{$FFFFFFFF$l. \ldots \ldots \ldots$ FFFFFFFF FFFFFFFE FFFFFC2F$\}$ that contains 256 prime numbers;
   - $E(1, 7)$ is the base point
   - $E \cup \mathcal{O}$ has cardinality
     $$n = \left\{ \begin{array}{c} \text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE} \\ \text{BAAEDCE6 AF48A03B BFD25E8C D0364141} \end{array} \right\}$$
     is the multiplicative order of $E(1, 7)$;
   - The private key 'k' is a random integer that can choose from the set $k \in \{1, 2, 3, \ldots \ldots, n-1\}$ and
   - The public key kP, Where
     $P = \{x, y\}$;
     $$x = \left\{ \begin{array}{c} \text{79BE667E F9DCBBAC 55A06295 CE870B07} \\ \text{029BFCDB 2DCE28D9 59F2815B 16F81798} \end{array} \right\}$$
     and
     $$y = \left\{ \begin{array}{c} \text{483ADA77 26A3C465 5DA4FBFC 0E1108A8} \\ \text{FD17B448 A6855419 9C47D08F FB10D4B8} \end{array} \right\}$$
   - **Output** (P, k)
2. ***Sign Algorithm*** $(k \in Z_p; m \in \{0,1\}^*)$**:** Sign is the signing algorithm that takes the input as secret key 'k' and message 'm' and produces the digital signature 'σ' as output m
   (a) $n \leftarrow$ 256-bit integer
   (b) **Compute** $H(m)$ = SHA-256(SHA-256('m').
   (c) Uniformly choose a **random integer** called **instance key** 'j'; $j \leftarrow Z_p$
   (d) **Calculate** $'jP' = (x, y)$
   (e) **Calculate** $r = xmodn; ifr = 0 thenGotoStepc.$
   (f) **Calculate** $s = j^{-1}(H(m) + k.r)modn;$ $ifs = 0 thenGotoStepc.$
   (g) **Output** $\boldsymbol{\sigma} := (\boldsymbol{r}, \boldsymbol{s})$ is the digital signature with 512 bits long for the message 'm'
3. ***Verify Algorithm*** $(kP \in Z_p; m, \sigma := (r, s))$ **Verify** is the verification algorithm that takes the input as a public keykP, and the message m and verifies the $\sigma := (r, s)$ is a proper signature or not
   (a) **Compute** $H(m)$ = SHA-256(SHA-256('m').
   (b) **Parse** $\sigma := (r, s)$ by calculating
   $j_1 = H(m) * s^{-1} modn$ and $j_2 = r * s^{-1} modn$
   (c) **Calculate** the Point $V = j_1 * P + j_2 * (kP)$; $V(x, y) \in Z_p$
   (d) if $\boldsymbol{V} = 1; ifandonlyif(xmodn == r)$ then
   The **Signature** is **Valid.**
   else.
   The **Signature** is **Invalid.**

Fabric-CA uses the above ECDSA Signature Scheme to authenticate the proposed work members by generating *RCert, ECert, TCert, and TLScert* Certificates. The notion of security for signature schemes was formally defined in [27] in various flavors.

(B) Consensus Mechanism The proposed healthcare system uses permissioned blockchain network infrastructure, implemented in Hyperledger fabric [26] that provides one unique characteristic: *Consensus Mechanism*. The fabric architecture relies on a deterministic consensus mechanism that performs four steps *execute, order, validate and persist*, as shown in the Fig. 2. The ordering service of the consensus process is mainly used for the administration point of the blockchain network, which proposes the new blocks, seeks consensus, and ensures all the transactions of the block within the network are executed in order based on the agreed value. The transaction will be written to a block after the block is validated by the peer using an ordering service guaranteed to be final and correct. Finally, update that block position in the ledger, immutably assured. The proposed work uses Raft's [32] consensus mechanism that supports crash and byzantine fault-tolerant ordering service. The Raft provides a decentralized approach to Kafka and Solo mechanisms that allow different organizations to contribute nodes to the distributed ordering service with high availability strategy. Raft protocol uses a "leader and follower" conceptual model in which RAFT nodes are always in one of three states- leader, candidate, and the follower. The leader is elected dynamically per channel among the ordering nodes, and its decision messages are replicated to all the follower nodes. The proposed application uses all the doctors as the leader node; patients, other doctors, pharmaceuticals, researchers, and healthcare organizations are the followers. The leader node will wait to commit its decision until most follower nodes have successfully written their decisions. Whenever the peers or applications send the endorsed transactions, the RAFT consensus mechanism automatically routes these endorsed transactions to the ordering nodes that receive these endorsements to the current leader of that channel. However, the applications or peers do not know the current leader at any time.

The proposed blockchain system uses ECDSA digital signatures [28] to verify raft peers that securely distribute each peer's public keys and signed certificates. These signed certificates are used primarily to create and validate the messages securely. When the peers join the blockchain network and those peers must be added to the system channel, and these nodes must follow the Raft consensus protocol. Raft protocol supports crash and byzantine fault-tolerant nodes, which can easily perform to impersonate a Raft node or masquerade as other nodes. To protect the raft nodes, every peer in the network identifies with each other using TLS Certificate. Without a valid TLS configuration, it is tough to execute the Raft peers by the attackers. In addition, the attacker needs to obtain the TLS certificates' private key, which is very difficult to impersonate the nodes. The Raft leader will get the requests to add the EHRs message into the block from the EHRs issuing peers of the network. The raft leader collects the incoming EHRs messages and forms a Merkle tree after ordering these messages. Finally, the leader signs the root of the Merkle tree using its private keys. The resulting EHRs messages are of the form $< BID,\ EHR\text{-}ID,\ EHR,\ RLID,\ RMT,\ Sign >$ where *BID* is the Peer Blockchain identity, *EHR-ID* is the *EHRs* messages identity, *EHR* is the EHRs messages, *RLID* is the Raft leader identity, *RMT* is the root of the Merkle root, $Sign: = Create\text{-}Signature_{Pr\text{-}Key}\ (RLID,\ RMT)$ is the signature of the root Merkle root from the Raft leader RLID. This mechanism is efficient because only one signature is used for all EHRs messages stored in the Merkle tree. All the communicating nodes will be signed to prevent replays and spoofing attacks. When the raft node sends the message to the other network members, the received nodes will securely confirm the identity of the sent raft node.

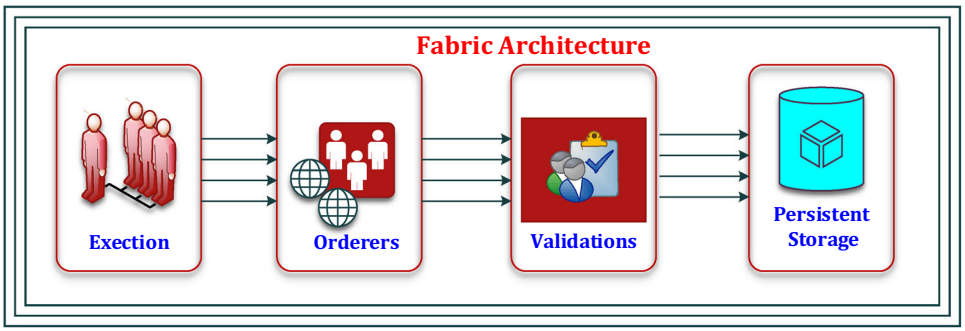


Fig. 2 Hyperledger fabric architecture transaction flow

3.3 System design and framework

This section presents the System Design and Framework of the proposed e-Healthcare application implemented in Hyperledger fabric that uses the Elliptic Curve Digital Signature Algorithm (ECDSA) mechanism for the security and privacy of Healthcare records over public networks. The following subsection discusses the system model of the proposed work, Registration and Identification of members, Authentication of the members in the Blockchain Network, Secure sharing and storing of Health Records in the Permissioned Blockchain Network.

3.3.1 System model

This section presents the system model that consists of three components- (A) A Permissioned Fabric Blockchain Network, (B) An offline storage for storing the EHRs data, and (C) An interactive platform for the users to access the blockchain network.

(A) **Permissioned Blockchain Network** The first component is the Hyperledger fabric provides a distributed and permissioned blockchain, which ensures that all the participants are trusted and known to the clients/user. Once the participant, like doctors, pharmaceuticals, or researchers, are registered on the blockchain network, the network is provided with a Unique ID generated by the Membership Service Provider (MSP). The transactions (EHRs) performed are stored on the blockchain network in an immutable manner and can't be modified once stored in the network. It ensures that the EHRs are secure and tamperproof. All the participants are trusted, and the chaincode (smart contract) allows the patient to make sure that his data is not misused. The proposed application work is implemented on hyperledger fabric [33], a major open-source enterprise blockchain platform hosted by the Linux Foundation. Specifically, this fabric is a permissioned blockchain framework that is always open to all/registered participants who wish to participate in the network. The Fabric blockchain architecture [33] is shown in Fig. 3 and runs on user-defined smart contracts that support scalability, flexibility, identity, resiliency, and strong security features- confidentiality, authentication, integrity, etc. Further, the fabric also gives the impression of executing the application on a single globally-distributed blockchain computer. In other words, it is a distributed operating system for permissioned blockchains that forms a distributed ledger network, which executes distributed applications and allows multiple identified nodes/parties to participate and execute consistently across many nodes/parties. The Fabric blockchain architecture maintains the distributed ledger that records all transactions between the parties/Nodes. Each transaction of the replicated ledger is append-only across all the parties/nodes. The proposed work is a permissioned blockchain network infrastructure, which is implemented in a hyperledger fabric that provides various services like a Membership service provider (MSP), Ordering service(s), Distributed Ledgers, and channel (s) and Chaincode(s). This fabric provides better functionalities like—Efficient parallelism and concurrency, multiple transaction executions, efficient commitments of the transaction into the ledger, etc., compared to other blockchain tools like Ethereum, Corda, etc. The MSP service is used to sign and issue the certificates and create the public and private keys for authorization and verification of users through the Certificate Authority (CA). This MSP is responsible for registering various entities and verifying those entities to the proposed applications. The proposed blockchain network permits multiple users of this application, such as the patients, doctors, hospitals, and other hospital entities such as laboratories and pharmacies, to enroll through a trusted MSP. The configuration file for this MSP service contains information about the channel(s) within the blockchain network. The file also includes the membership information (certificates) for all the channel members within the organization and the channel's policies within the network. The Ordering service(s) are used for ordering the transactions. Specifically, the ordering service is responsible for collecting the various transactions using consensus mechanisms—the whole process is shown in Algorithm 1.

(B) **MongoDB** The second component is MongoDB, a document-oriented database program and one of the classifications of NoSQL databases. The EHR data will be uploaded and stored in a 64-bit format in MongoDB. The Hash value of EHR data will be generated using the MD5 hash algorithm. The hash value and Patient-ID will be stored in the world state database. As the world state database keeps track of updated EHR data and transactions, the hash value of the EHR will be stored in the blockchain. Hence, using this scheme improves storage, speed, and performance.

(C) **User interface** The third component is a web-based application, which helps users communicate with the blockchain network that allows all the participants to exploit the properties of the blockchain. Once all the actors and the patients can register to the network.

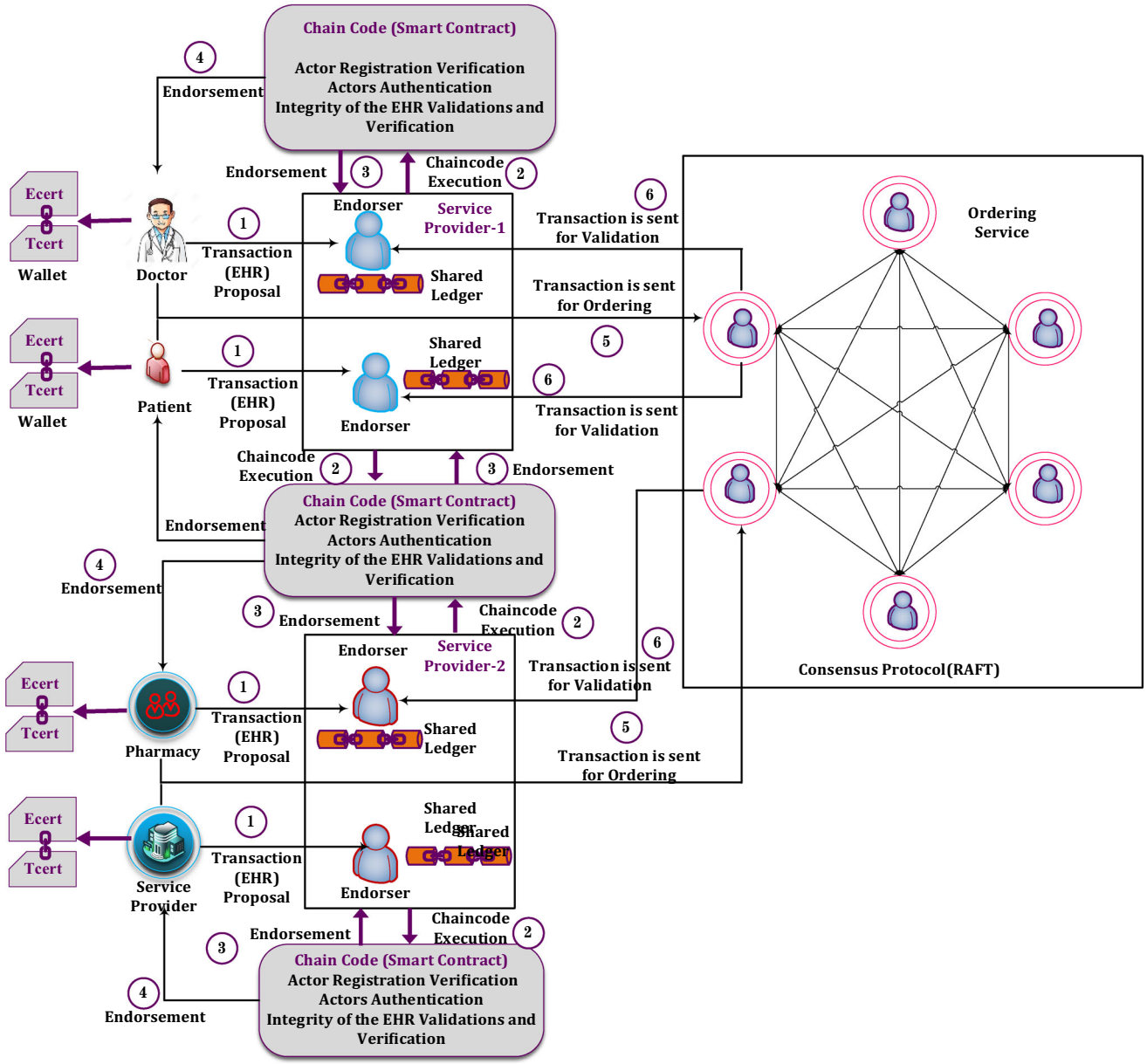


Fig. 3 Hyperledger fabric blockchain components

The Doctors will look into the EHR record and upload it into the network (or) the doctors can request the patient to permit access to their EHR record. The patient can also grant access to other parties such as researchers, companies, and pharmacies.

Algorithm-1: Managing and Creating the Orderer Nodes in Healthcare Organizations

```
{
   While (Create the Multiple Healthcare Organizations)
   {
        Type of the Orderers
        Creating and managing Orderer peer nodes
        Creating the Orderers with their Domains < org1.example.com >
              Hostname: Doctor; Hostname: Pharmacy
        Orderer Domain and Host Addresses
        Enable the Orderers Nodes
        Display the Maximum Number of user accounts in each organization
        Block-Size ← Number of Transactions in each block + Maximum Number of Bytes
        Maximize-Size ← Maximum of Bytes allowed for Serialize the Block
        Initialize the Timeout of each Block;
        Connect the Orderer through Broker
    }
  End While
}
```

3.4 Registration and identification of members in hyperledger fabric framework

The proposed healthcare blockchain system completely records the health information securely and easily accessible by keeping all the information in one place and efficiently verifying the EHRs, ensuring the privacy of the EHR data. The proposed blockchain network has multiple peers classified as default and client peers. These peers must register on the network to invoke the chaincode and perform the transactions. The default peers are the core peers part of the blockchain network, such as endorser, orderer, and anchor peers. At the same time, client peers are the participating peer who can perform the transaction on the network. Every participating peer will have an identity in the fabric blockchain network provided by the fabric's membership service provider (MSP) module. This MSP module contains security infrastructure that issues and maintains the identities of the clients and verifiers the endorsed peers and orderers in the blockchain system at the time of the registration process. These credentials are mainly used for identification, authentication, and authorization purposes. The registration of nodes and key management is part of the MSP service, which uses standard PKI methods to perform the digital signatures on the transaction, authenticate the peers, and provide the certification authorities (CAs). The orderer's and endorsed peers' credentials are also generated by a fabric-CA of the MSP module and distributed to all the peers in the network.

The hyperledger fabric provides a special peer called an admin, who has administrative control, is registered with the Fabric Certificate Authority (CA), and is responsible for enrolling or removing the entities in the network. The algorithm for enrolling the admin is shown in Algorithm 1. The Fabric-CA sends the enrolment call to the root server to retrieve the enrollment certificate (ECert) for this admin user. The Fabric-CA root server invokes a certificate signing request (CSR) in the ECert of this Admin, which stores the ECert key material in the blockchain network. The identity keys are stored in the wallet for storing the keys and the client's identity information. The admin is mainly used to register and enroll a new client, endorser peers, and Orderers into the network. This admin peer is also used to install the chain code on every endorsing peer that has been enrolled. The various clients use the NodeSDK, which can access the API for performing the user identity registration and enrollment process using the membership service process (MSP) and connects to the network to submit transactions for querying or updating the distributed ledger (updating details or storing records metadata or hash). Thus, the registration and identification process helps the client store the user identity and public and private key details in the wallet, which will be used for the authentication for submitting all the transactions.

Algorithm-2: Chaincode for Enrollment of Administrator and Clients

```
{
  Input: Enrollment of Admin
  Output: Registered into Blockchain Network
  Fabric-CA-Services← Fabric-CA-Client;
  Admin-Wallet ←Fabric-Network;
  Crypto-System ← Fabric-Network-Crypto;
  Connect-Path ← Blockchain-Network (Connection-1, Connection-2…….Connection-n);
  Create a New CA client for interacting with the Fabric-CA.
        Identity-Information ← Certificate-Authorities ('ca.org1.health-provider.com')
        Generate TLSCACerts ← Admin/Clients;
        CA ← FabricCAServices (TrustedRoots: caTLSCACerts, verify: True, 'Domain-Name')
  Create a new file system-based wallet for managing identities.
       wallet ← FileSystemWallet(Admin);
       adminExists = await wallet.exists('admin');
       if (adminExists)
        {
          Display('An identity for the admin user "admin" already exists in the wallet');
          return;
        }
       Else-if (Enrollment)      Enroll the admin user and import the new identity into the wallet.
        {
          Enrollment ← Await-CA-Enroll(EnrollmentID: 'admin', enrollmentSecret: 'Password' });
          Identity ← X509-Wallet.CreateIdentity('Org1MSP', enrollment.certificate, enrollment.key);
          Wallet ← Import ('admin', identity);
        }
        Display ('Successfully enrolled admin user "admin" and imported it into the wallet');
       } Endif
  Display (`Failed to enroll admin user);
} End of the Algorithm
```

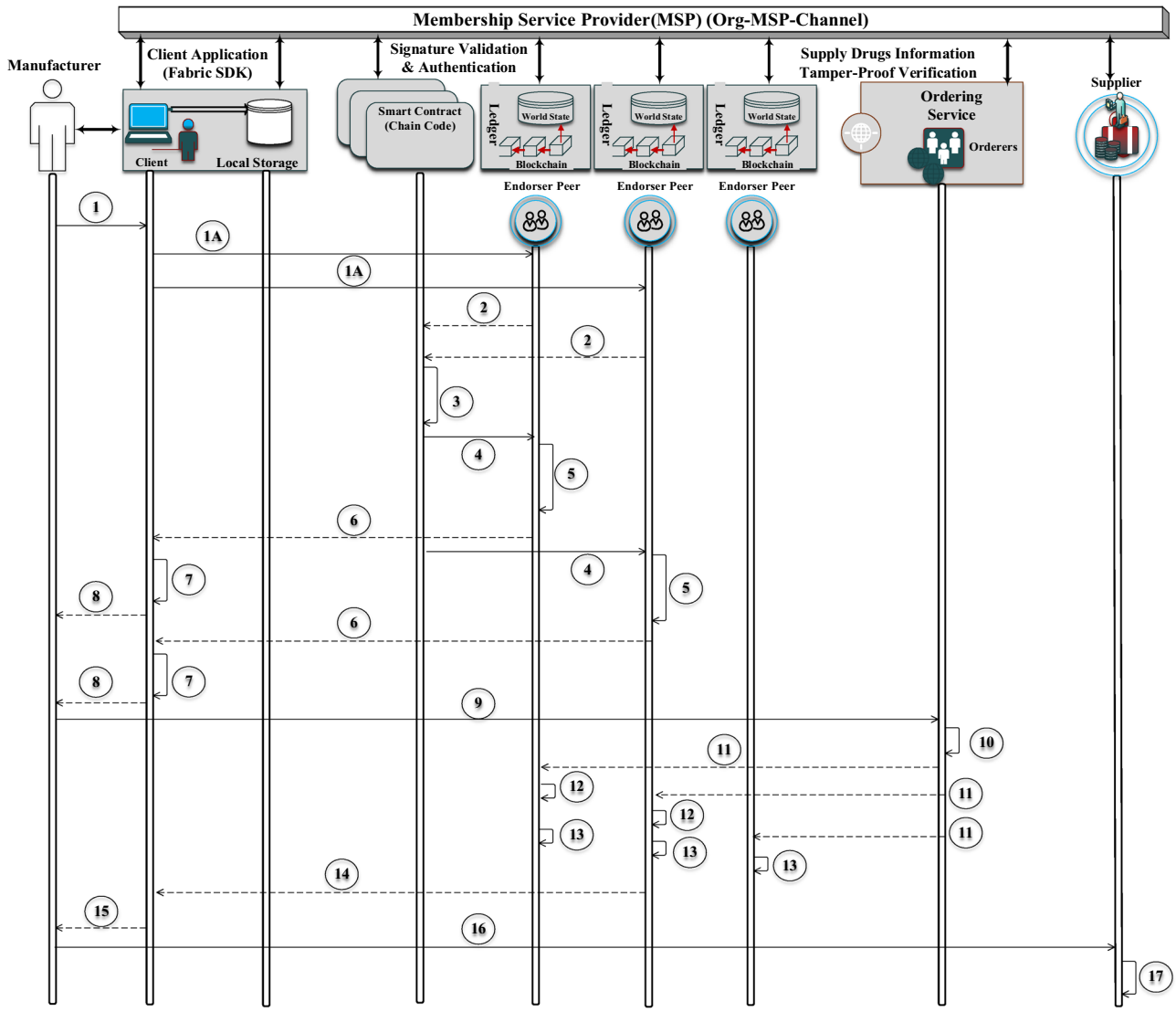
The User-Interface (UI) for the clients is developed using React-Redux that can be used by doctors, patients, hospitals, laboratories, pharmacies, researchers, and insurance companies to register onto the network. The proposed application uses NodeSDK (software development kit), which acts as a gateway for interacting and accessing the blockchain ledger and the endorser peers of the network. The clients make the REST API calls to the server running using the Node SDK to connect a gateway to the network. This SDK is also responsible for enrolling the users, issuing the TLS certificates, storing the keys, and generating identities generated in a wallet. The end-user can interact with this gateway and the blockchain network through the React framework, which is used to invoke the REST APIs. The patient details such as Username, Registration-No, or the Medical-Registration-ID are sent with the POST API call to create the Blockchain ID (BID) for all the clients who need to register on the network. Once the client is registered, they can join the network by logging in through their respective portals and submitting the transactions signed by the secret key stored in the wallet for every entity. Moreover, this API performs the GET and POST requests to the blockchain network, enabling client applications to invoke chaincode functions and permits submitting the request to either send or retrieve information from the network. The blockchain network will evaluate the transactions, order the transactions, and commit the transaction into the distributed ledger that either update the world state or fetches data from the world state. The distributed ledger stores the EHRs information and other related documents in base64 format, and the equivalent hash value is sent back to the network. Whenever the patient/doctor accesses or retrieves the EHR documents. First, the network validates the generated hash value with the document's hash value stored in the mongo-DB, which should be matched to ensure the EHR data cannot be tampered with by the adversary node. Finally, the successful notifications will be sent back to the client application.

Algorithm-3: Enrollment of Creating Patient in Chaincode

```
{  Input : Enrollment of Patient
   Output : Reponses to the Enrollment of Patient
    If (PatientExists)
        Return (Patient with username already exists);
    Else
     {
      if (!patientExists)
      {
        New-Patient ← Create-Patient (First-Name, Last-Name, Address, ID-No, DOB, Gender, Blood-Group, User-
                                      Name, Phone, Password);
        New-Patient-Details←Create-Credentials (Permissioned-ID, Emergency-Contacts, EHRs, Requesters, Bills-
                                      Generation, Medicine-Receipts, Lab-Records, Appointments);
        Response ← New Patient with Username is updated in the Blockchain world state Ledger;
        Return(Response to the New Patient);
      } End of inner if
     } End of Outer if.
} End of the Algorithm
```

The following steps are used to register and identify the user in hyperledger fabric; one must follow the sequential steps shown in Fig. 4. The number in the circle represents the step number. A detailed description of each step is given below.

1. The client requests Blockchain Identity (BID) through SDK API to the Fabric-CA/Admin of the blockchain network. The BID is the cryptographic identity of the client that is used to authenticate itself in the messages.
2. The fabric-CA requests the Registration-CA (RCA) to process the request from the client user. Here, the Registration-CA is one of the trusted intermediate authorities that validity and identity and authenticates new users, endorser peers, and orderers who want to participate in the proposed permissioned blockchain.
   (a) Once the request is received from fabric-CA, then
   (b) RCA creates registration credentials such as user account in the form of Blockchain User Identity (BID) and Password needed for enrollment and transaction purposes and supports the evidence as user's identity (BID) in response to the submitted user's request and stores the BID and password in its local database.
3. Further, RCA requests the trust anchor TLS_{CA} certificate for processing the BID and Password of the client user.
   (a) TLS certificate is a self-signed X.509 certificate trusted by the blockchain network for securely provisioned TLS communications between the endorser peers, orderers, and local client systems.
   (b) These TLS certificates validate the user's registration and enrollment credentials and allow them to use its blockchain network.
   (c) This certificate mainly carries the user's identity and is used for network-level security.
   (d) The TLS-CA will process the client's BID and Password request and create the certificate TLS-CERTTLS-CA by signing using his private key and storing the TLS-Cert $(BID, PWD)_{TLS\text{-}CA}$ in its local database and returns TLS-Cert $(BID, PWD)_{TLS\text{-}CA}$ to the Registration-CA.
   (e) The RCA will be subsumed as part of the fabric-CA that returns the BID, Password TLS-CERTTLS-CA to the Fabric CA.
   (f) The fabric CA issues digital certificates in ECert, TCert, and TLSCert to the new user.
4. Next, the client user will enroll with BID and Password to the client system. The client user connects to its local client system for enrolling in the BID and password.
5. On behalf of the client user, the local client system requests Enroll-Register (Public Key, BID, and Password) to the Enrolment-CA (ECA), which is part of the fabric PKI framework the blockchain system.
6. The ECA verifies the Enroll-Register Request (BID and Password), which already exists in its co-located local database.
   (a) The Enrollment Certificates (ECerts) are mainly used by Enrollment Certificate Authority (ECA) for validating the registration credentials provided by the client user.
   (b) The ECA also certifies and ensures that the client user has the right to submit the enrollment public key.



1 Request (Transaction Proposal SDI) 1A Transaction Proposal-Invoke Chaincode 2 Execute the Chaincode

3 Chain code Validates the Identity Credentials & Chain code Verify and Generates the Recipients Credentials as Per Endorsement Policy

4 Endorsement is Successful 5 TX-Proposal Hash(EHR) & Signed the TX-Proposal using his Private Key

6 Transaction Proposal Response (Endorsement, Public-Key) 7 Verify the Tx-Proposal (Endorsement) 8 Signed EHR is Verified

9 Request the Transaction(Signed EHR) are Ordered ; Encapsulate the Both Endorse Signed EHR Tx and Signature of Endorse Peer1 and Endorse Peer2.

10 Both Transaction are Ordered by per the Consensus Meachanism & Transactions are placed in Blocks

11 Send Blocks (Transactions(EHR)) placed Blocks for Validation 12 Update the World State and Link the Block into Blockchain

13 Validate and Verify the Block 14 Committed Notification of Endorsement Transaction) 15 Signed EHR is Generated

16 Nofity and Send Signed Encrypted JSON File (EJF) to the Supplier 17 Verify the Signature using Manufacturer Public Key & Decrypt the EHR Using his/her Private Key

Fig. 4 Registration and identification of user in hyperledger fabric

(c) The Enrolment-CA (ECA) creates the signature with an enrollment certificate (ECert) containing the Client User Public Key and generates self-signed ECA-CertECA needed to further process and prove user enrollment.

7. After generating the ECert certificate, the ECA responds ECert (Client Public Key) and ECA-$Cert_{ECA}$ to the client user.
8. The client user verifies the submitted public key, and the ECert public key is the same. The ECert is decrypted using ECA Public Key to retrieve his (client) public key.
9. Next, the client user sends a registration request for TLS Certificate (TLS-Cert (TLS-Client public key)) to the TLS-CA.
10. Once the TLS-CA receives the request, it verifies the client user credentials already existed in its co-located local database. Here, the (TLS-Cert (TLS-Client public key)) mainly carries the identity (TLS Public Key) of the client in a blockchain network.

    (a) If it exists, then the TLS-CA creates the signed TLS Certificate (TLS-Cert), which contains the client user TLS public key and generates self-signed TLS-$CERT_{TLS\text{-}CA}$ needed for processing the transaction proposed by the endorser peer.

11. After generating the TLS-Cert certificate, the TLS-CA responds TLS-Cert (Client TLS public key), TLS-$CERT_{TLS\text{-}CA}$ to the client user.
12. The client user verifies whether the submitted public key and the TLS-Cert public key are the same.

    (a) The TLS-Cert is decrypted using TLS-CA Public Key to retrieve his (client) TLS public key.
    (b) If it matches, then it stores in its local database. Note that the Enrollment process will be the same for the new users, endorser's peers, orderers, and verifying institutions.

13. Next, the client user requests for transaction certificate (TCert (ECert (Client Private Key))) to the Transaction-CA (TCA).
14. The Transaction Certificates (TCerts) are mainly used by the Transaction Certificate Authority (TCA) for validating the enrollment credentials provided by the client user.
15. Once the TCA receives the request, it issues the transaction certificates (TCert) by pseudonymous authorizing the client user to submit the transaction by generating TCert(Client Private Key), self-signed TCA-CERTCA using TCA private key, and response to the request (TCert, TCert (Generate-DF Key), TCA-CERTCA) back to the client user.

    (a) The TCA returns TCerts sets that contain the Key-DF Key (Key-Derivation-Function Key). This Key-DF_Key is mainly used to derive the client user's private key.
    (b) The TCerts helps the client user prepare and sign the transaction proposal, process, and store the Digital Academic Certificate (DAC) in the network.

16. The client user verifies TCert (Generate-DF Key) using decrypting with the public key of TCA and invokes the DF-Key method, which generates the private client key of the user.

    (a) Finally, the client user stores ECert (Client Public Key), TLSCert (TLS Public Key), and TCert (Generated Private Key) in its local database.

The whole process of the proposed work is performed using the customized cryptogen tool that generates a file called crypto-config. yaml. This file generates cryptographic material such as a set of X.509 certificates and signing keys for the blockchain network. These certificates will serve as the identities to authenticate peers' transact and communication.

3.5 Authentication of the members in the blockchain network

The blockchain network is responsible for checking the authenticity of the registered users. The blockchain network administrator monitors all the identities stored in the wallet. Suppose the blockchain network believes that the identities of the registered user are fraud or are no longer licensed. In that case, the administrator removes those identities from the wallet, and they no longer be able to perform any transaction. Moreover, the administrator keeps monitoring all the registered entities and working properly according to the policy of the proposed application. The registered client can't access the offline database as it is not linked with the end-user application and can be accessed only blockchain network via the Medical Register Association. Only the verified users will be allowed to keep submitting the transaction. The admin always monitors all the entities present via User-Interface (UI) and can remove the entities by making an API call to remove the user identity.

3.6 Secure sharing and storing of health records into permissioned blockchain network

This subsection presents (A) Hospital Appointment and Doctor Assignment by the Health Care Provider. (B) Generation of EHR Document by the Doctor. (C) To secure the Patient Health Records. (D) Secure sharing of health records within Blockchain Network and (E) Secure sharing of health records between Doctors, Patients, and healthcare providers. (F) Secure storing of health records in a distributed manner.

(A) **Hospital Appointment and Doctor Assignment by the Health Care Provider** The patients can invoke the chaincode exposed through the NodeSDK using the React framework as the interactive User Interface (UI). Whenever a patient wants to book an appointment, the patient searches for the nearby hospitals and inputs the time and the ailment the patient feels after selecting the hospital. After requesting an appointment, the React framework does a POST call to the blockchain network. The network receives a book appointment call from the patient user. Then, first, the network checks whether the patient's identity exists in the network or not. Otherwise, the user should register first to book the appointment. The whole process is presented in Algorithm 4. After verifying the existence of the patient's identity, a gateway is created that connects to the network, and the chaincode is fetched. After that, the chaincode is invoked, and the parameters are passed to book an appointment in the respective hospital chosen by the patient. When the chaincode is invoked by the gateway, the endorser peer executes the chaincode, and the patient's credentials are verified. Once the endorsement is successful, the ledger for that peer is updated, and the endorsing peers return a proposal to the client. The endorsed transaction proposals are then ordered into blocks and distributed to all peers for final validation and commitment. Once the transaction is committed, the hospital gets a new notification about the new appointment request. Based on the ailment mentioned, the hospital selects the appropriate doctor from the doctors working there. After choosing the doctor, the hospital makes a POST call to assign the doctor for that appointment. On receiving the post-call to assign a doctor, the server first checks for the existence of the hospital's identity through the registration number of the hospital. Once the identity is verified, the gateway created by the hospital identity checks in the connection profile for the peers and the certificates of the hospital. After that, the contract is invoked, which is already deployed on the various peers in the network. Once the chaincode is invoked, the endorsing peers execute the chaincode with the parameters passed to the chaincode. The ledger for the user is updated, and the endorsement is sent back as a proposal to the gateway, which is ordered into blocks by the orderers in the gateway. Once the world state is updated, the doctor's and other peer's ledgers are updated. Once the doctor is assigned the appointment call and updated, a notification is sent to that doctor informing about the coming patient.

Algorithm-4: Register the Patient in the Blockchain Through the Gateway

```
{
  Input :  Register the Patient
  Output : Registration is Completed/Failed to Register
  Router ← Install (Router);
  X509.Wallet ← Require (Fabric-Network);
  Gateway ← Require (Fabric-Network);
  Connect-Path ← Blockchain-Network (Connection-1, Connection-2…….Connection-n);
  Blockchain-Network ← Require (Access-EHR-Document-Database);
  Path-Setup ( X509.Wallet, Gateway, 'Blockchain-Network');
  Router-Post (Request and Response)
  {
    //Create a new file system based wallet for managing identities.
    wallet ← FileSystemWallet(User-Name);
    User-Name-Exists = await wallet.exists('User-Name');
    // Check to see if we've already enrolled the user
    if (User-Name-Exists)
    {
      Display("Candidate has been already registered");
      return;
    }
    // Check to see if we've already enrolled the admin user.
    if (!adminExists)
    {
      Send("Admin is not currently enrolled. Please wait for some time...");
      return;
    }
    // Create a new gateway for connecting to the Peer node.
     Connect-Gateway (wallet, identity);
    // Get the CA client object from the gateway for interacting with the CA.
    CA←  Gateway (Client-Get-CertificateAuthority());
    Admin-Identity ← Gateway (Get-Current-Identity());
    // Register the User, Enroll the user, and Import the new identity into the wallet.
    Secret-Key ← CA-Register (Affiliation-Domain, Enrollment-ID, Role of the Client, Admin-Identity);
    Enrollment ← CA-Enroll (Enrollment-ID (User-Name), Enrollment-Secret (Secret-Key));
    User-Identity←  X509Wallet-Create-Identity  ('Organisation-Name-MSP',  Enrollment-Certificate,  Enrollment-
                                                Key);
    Response← Import-Wallet(User-Name, User-Identity);
    //Getting the information from Blockchain State Ledger
    Response ← Register-In-Ledger(Request)
    {
      // Get the blockchain network channel for the proposed contract deployed.
       Blockchain-Network ← Gateway-Get-Network ('Channel');
       // Get the contract from the network.
      Contract ← Blockchain-Network-Get-Contract ('EHR');
      // Submit the specified transaction.
      If (Contract)
      {
        Response← Contract-Submit-Transaction ('Create-Patient);
        Return (Response)
      }
      Else
        Return("Failed to submit Transaction to the ledger");
    }
    Registered-User ← Database-Handler-Register-New-User-Request (User-Name Patient Credentials');
    Response← send("Correct");
  }
  Else
  {
   Return (`Failed to register user);
   //Response to the User
   Response ← Send("Failed to register candidate");
  }
} End of the Algorithm
```

(B) Generation of EHR Document by the Doctor Once the completion of treatment of the patient by the allotted doctor, the doctor generates the EHR for the patient. The doctor selects the EHR of the patient and uploads it on the web-app. Once the document is uploaded, based on the requirement of the laboratory or pharmacy, the doctor chooses the laboratory or pharmacy present in the hospital. The doctor makes a POST REST API call to upload the document. The document is sent to the blockchain network. The whole process is presented in Algorithm 5. When the request is received, the multi-service handler grips the file and prepares it for upload into the database ledger. The offline database ledger of this application is mongo DB, which is used to store various non-essential file details uploaded by doctors, hospitals, laboratories, and pharmacies. The grid-fs-storage package of mongo can store the files in chunks as one separate collection that stores the metadata along with the hash (MD5) of those files. The file is stored in Base64 format and divided into chunks to facilitate the database's distributed nature. The file's name is then encrypted using a random function and stored in a different collection utilizing the patient's username. This new random encrypted file and the document's hash value are uploaded and sent back to the doctor, who signs the transaction with his identity. The hash value and the file name are used to create an EHR document in the world state. The ordering services broadcast the block to all the participating nodes in the network once the endorsement is completed after the EHR document is stored in the world state. The file is updated in the patient's world state, and the access is completely transferred to the patient single-handedly. Based upon the request, access is given to various other clients. The hash value of the stored EHR document will be used not only to ensure the integrity of the record that is not tampered with but also to verify the hash value for gaining access to the data.

(C). Securing the Patient Health Records To secure the patient health record, every member of the blockchain network must follow specific guidelines related to securing the data. Any unauthorized access to information can be restricted using security measures and services. The proposed work uses blockchain servers that use Transport Layer Security (TLS) protection to prevent individuals from accessing any information they are not authorized to access the data. The client data will be encrypted using his private key, which is generated at the time of registration to the blockchain network using the Elliptic Curve Digital Signature Algorithm (ECDSA) $K_{Pu}^{Issuer}, K_{Pr}^{Issuer} = Generate\_Keys\_ECDSA(BID_{User})$ and $Encrypt(EHR_{ID}, K_{Pr}^{Issuer})$. The EHR data will be stored in offline storage in encrypted form. It is very difficult for the attackers to get the required keys and decrypt the EHR data. Moreover, the sharing of keys completely depends on the patient's choice; keys stored in blockchain will not be compromised easily. The proposed system combines the keys with smart contracts (chaincode) that prevents unauthorized parties from adding, deleting, and updating the information to a patient's health records, including outsiders trying to tamper with data for malicious or self-serving purposes. Finally, the encrypted EHR data will be hashed using the hash generating algorithm SHA-256, and the hash value will be stored in the blockchain. $Blockchain_{Network} = H_{256}(Encrypt(EHR_{ID}, K_{Pr}^{Issuer}))$. The storing of EHR data only will be done after encrypting the data. Hence, securing clients' EHR data can be ensured.

(D) Secure Sharing of Health Records within Blockchain Network The following process presents the secure sharing of patient health records that can be ensured using the Advanced Encryption Standard (AES) symmetric encryption key.

1. When user registration and public and private keys, the patient will also have a symmetric key with which their health records get encrypted and stored in a database.
2. When a patient's health records are requested to be provided by any healthcare provider, it's completely the patient's choice to provide access or revoke access to his health records.
3. If a patient wants to provide access to the EHRs to the healthcare provider, the patient can share a symmetric key encrypting the key with the healthcare provider's public key.
4. If the symmetric key is compromised, the patient can generate a new one by running a pseudo-random generator algorithm.
5. After getting symmetric key details from the patient, the healthcare provider decrypts the symmetric key using his private key and retrieves the records from the blockchain database.
6. The healthcare provider decrypts the record using a symmetric key and performs respective operations.

Algorithm-5: Generation (Creation, Updation, and Deletion) of Patient EHR

```
{  Input: Creation of Patient EHR
   Output: Responses to the Patient EHR Creation
   //Creation of Patient EHR
     If (EHR-Exists)
     {
        Return ( Hospital-Registration-ID,  EHR-Patient-ID,  Doctor-ID←Medical-Registration-No,  EHR-Record,
                 Appointment-ID, EHR-Time);
     }
     Else
     {
        Response ← Create-EHR(EHR-Record);
        EHR-Record ← EHR( EHR-ID, Patient-ID, EHR-Doctor-ID, Hospital-ID, Date, Time);
        Return(EHR-Record);
     }
   //Update the EHR in the list of the patient EHRs into the World State of Blockchain Ledger
      If  (Patient-ID-Exists)
      {
         Verify (Patient-ID);
         If (EHR-Exists)
         {
            Verify Appointments-Index(Patient-Appointments, Appointment-ID);
            World-State ← Push(EHR-Record, EHR-ID); //An EHR is updated and stored in the world state
         }
         Return(World-State);
      }
}End of the Algorithm
```

(E) Secure Sharing of Health Records between Doctor and Patient The following process presents the secure sharing of patient health records to the doctor.

1. The doctor requests the patient access to his health record with an identifier EHR ID and sends the Doctor's ID along with the request. Here the patient will be the Issuer, and the doctor will be the receiver.

   $$Request - Patient\,(EHR - ID,\ BID - Issuer;\ BID - Receiver)$$

2. The patient will now receive a notification and can either accept or reject the request sent by the doctor. If the patient clicks ACCEPT, BID-Receiver (Doctor) will be added to the Access Control List (ACL).

   $$Add - ACL - Issuer\,(BID - Receiver)$$

3. The patient initiates the transaction of key sharing by encrypting the symmetric key (S_{Key}) and EHR-ID with the public key of the doctor (receiver), Note: EHR-ID is already encrypted with the symmetric key (S-Key) and sends it to the doctor.

   $$Send - to - Doctor\left(Encrypt\left(S_{Key}(EHR - ID);\ Receiver_{Pu}\right)\right)$$

4. The doctor will now receive the encrypted key as M and Decrypts it using their Private key (private key of the receiver)

   $$Decrypt\,(M,\ Receiver_{Pr})\,M:\ S_{Key}(EHR - ID)$$

5. The transaction is added to the blockchain network.
6. A doctor can now retrieve the encrypted health record from the database using EHR-ID and decrypt using the S_{Key} and access the records.

   $$EHR - ID\ =\ Retrieve - Record\left(M, S_{Key}\right)$$

(F) Secure Storing of Health Records in Distributed Manner The proposed permissioned blockchain is a decentralized system in which each member in the network contains a copy of the ledger and contributes to the validation and certification of transactions. The health records are stored in a distributed manner within the network.

1. EHRs data will be fragmented into chunks and distributed across the network rather than having complete data in a single place.

2. These chunks are encrypted and uploaded onto the blockchain.
3. All the chunks are distributed and available even if part of the network is down. This is also known as redundancy.
4. The EHRs will be added to a shared ledger by network participants. Once the EHRs are added to the ledger, they cannot be altered.
5. All the participants will have a copy of the ledger, and any changes can be made only when the majority of the peers need to approve it. Hence, a single user can't alter the data.

All the Health records will be stored in MongoDB in 64-bit format. The Hash value of these records will be generated using the SHA-256 hash algorithm. The Hash value and Blockchain ID, and Patient-ID will be stored in the world state database as a key-value pair but not in the blockchain network. The world state of the ledger keeps track of the current state of the EHRs of the blockchain. Only a limited amount of data is stored in a blockchain network that reduces the computation overhead, and processing time will be efficient.

3.7 Privacy of stored health records

The proposed system also ensures the privacy of stored patient health records. To ensure privacy, all the blockchain network members must follow specific procedures related to privacy for protecting personal data. The proposed architecture uses two types of storage to ensure the data's privacy. One is online storage, and the other is offline storage. Online storage refers to the blockchain network's ledger that stores the clients' data, hash values of encrypted records, and transactions. The indexed key of the corresponding user data is stored in offline storage. While offline storage is mainly used to store the actual records of patients in encrypted forms as key-value pairs and the indexed key of corresponding data is available in online storage. The Couch DB is used as offline storage. The clients cannot access offline storage directly without the permission of the blockchain network. The health records can be viewed only when a patient, based on the access policy of the blockchain network, grants access.

1. When a third party or health provider requests the access handler to provide access to patient EHR data using his $< BID,\ EHR\text{-}ID >$. The access handler sends BID to the policy repository in the blockchain to verify whether the third party or health provider can be granted access.
2. The access policy repository verifies the BID of the requester and sends an Access Verification Response (AVR) to Access Handler.
   (a) If Access Verification is valid, then Access Handler provides Access Response, using which a third party can access patient EHR data.
   (b) If Access Verification is not valid, then Access Handler queries patients to provide access to the third party or health provider.
   (c) If the patient agrees to provide access, then Access Request Handler sends a new access policy to Access Policy Repository and requests to verify again.
   (d) After verification, it sends the result to Access Request Handler, which sends a response to the third party or health provider to access the resource.
3. If the patient denies granting access to the third party, a rejected request notification will be sent to the third party by Access Request Handler.

3.7.1 Grant/revoke access of EHRs document from the blockchain system

The proposed application allows only registered users that the network admin authenticates. These authenticated users can access the network and do transaction operations. However, users or patients can share the details with the various client peers, who can only view the details but cannot access the data. The registered client details are stored in couch DB in a key-value manner, an online blockchain database that the key holder can only access through the gateway established using the NodeSDK. The various users can request access to the medical documents from the patient based on the need. For instance, Insurance companies want to check medical claims; the hospitals might want to verify bills; the doctors might want to study the previous medical history of the patient, etc. All these requests are only provided to the requester by the patient, depending on the type of request. For instance, the doctor requires the history of the patients' previous treatment and will request using the patient's identity to access such documents. Subsequently, the network asks permission from the patient to access the EHR documents discussed in the following.

1. If any client users want the access the patient's EHR document, then the user requests through UI to access the documents.

2. Once the network receives the request from the requester for accessing the document, who doesn't have access to those documents.
3. The network verifies the credentials of existing permissions rights that the user owns. To do this,
   (a) The POST call is sent to the network along with the patient's identity, requester type, and registration ID.
   (b) The network verifies the user identity by verifying the credentials of the client node.
   (c) If the identity's existence is present, it creates a gateway to invoke the chaincode.
   (d) The endorser peer in the connection profiles in the network will invoke the chaincode execution by updating the user request in the patient world state and sending the endorsement to the client through a gateway.
4. Next, the client will send the request to the orderer for ordering service for inserting the request into blocks.
5. Once the transaction (request) is verified and added to the blockchain and the block is broadcasted to all the nodes, the patient receives the notification about the requester requesting the EHR documents.
6. The patient can either ignore the request or grant access to some documents only depending upon the need.
7. After successfully selecting the list of the documents, the patient makes a POST call to the network.
8. The network receives the parameters having the requester id, patient id, and the list of documents the requester can access.
9. Once again, the identity is checked and confirmed. The same flow for the transaction endorsement is followed, and when everything is verified, the transaction is submitted with the digital signature of the patient.
10. The requesters are added to the permissioned IDS, and the patient is also updated in the world state of the requester.
11. Finally, the block with this transaction is broadcasted to all the nodes in the network. The whole process is presented in Algorithm 6.

Algorithm-6: Request the Access to EHRs from Patient

```
{   Input : Request the Access of Patient EHR
    Output : Patient EHRs is Granted/Rejected
    // Process in Blockchain and Verify the Credentials
    if (Requester-ID-Exists && Patient-ID-Exists)
    {
        //Get the patient and update the request for that patient
        Push(Requester-ID, Requesters-Index);
        Put-State (Patient-User-Name);
        Response ←"Request to access the EHRs Documents has been submitted with the patient";
        Return(Response);
    }
      // Grant-Access Request Using Blockchain
     if (Requesters-Index > -1)
    {
        //Remove the requester from the Requester Block  and
         // Put the Requester into the permissioned IDs with the list of all the EHRs-ID that particular can access
        Put-State( Requesters-index(1), Permissioned-ID←Assign(Requesters-ID), Patient-ID);
        //Update the patient in the Patient Block for the requester
        Response ←"Access has been provided to the requester  with the Requester-ID";
        Return(Response);
    }
       else
    {
       Return("No such requester  Exist") ;
    }
}
    else
{
    Return("This requester-ID or the patient-ID doesn't exist`);
}
}End of the Algorithm
```

4 System setup and performance analysis

The proposed system comprises the following components that have been implemented in Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz, 16 GB RAM Ubuntu 18.04 LTS. Hyperledger Fabric-1.4 version is used to run the proposed system. This proposed fabric network is implemented in hyperledger fabric consisting of Fabric Certificate Authorities through MSP, Ordering service(s), Endorser peer nodes, Distributed Ledgers, Channel(s), and, Chaincode. The Fabric Certificate Authority (CA) is one of the important services provided to the proposed application, which mainly issues the certificates for the members of the organizations for authentication and identification. The blockchain network permits the various users of the application, such as the patients, doctors, hospitals, and other hospital entities such as laboratories and pharmacies, to enroll in the network through a trusted MSP which is responsible for the creation of signing certificates, issuing the public and private keys for authorization and verification through the Certificate Authority (CA). The ordering service uses the consensus mechanism, which is mainly used for the administration point of the network. This service contains a configuration file for the channel(s) within the blockchain network. Specifically, this configuration file contains the membership information for all the members of the channel within the organization and the channel's policies within the blockchain network. This fabric provides better functionalities like—Efficient parallelism and concurrency, multiple transaction executions, efficient commitments of the transaction into the ledger, etc., compared to other blockchain tools like Ethereum.

The End-user applications are placed outside the blockchain network that uses Software Development Kit (SDK) to interact with the blockchain ledger by connecting the endorsed peers when they need to store and access ledgers (EHRs Documents) chaincode. This SDK is also responsible for enrolling the users, issuing the TLS certificates, and storing the keys and the identity generated in a wallet. The End user can interact with the gateway and the server through the React framework used to invoke the REST APIs and perform the GET and POST requests to the server. Specifically, the SDK API enables the client applications to invoke the chaincodes to generate the transactions, connect the network peers, submit the EHRs documents to the network, order the transactions, and commit the transaction into the distributed ledger. Once completed, the successful notifications will be sent back to the end-user. The EHRs and other similar documents are first stored in the distributed database in base64 format, and their hash value is sent to the blockchain system. To retrieve the EHR document, the hash value should be matched with the hash value of the EHR document stored, ensuring the system's tamperproof.

We tested the network's overload related to proposed applications that invoke transactions and update the ledger (storing the EHRs) to test the proposed system. The overload of the network for uploading EHRs applications, Accessing and Granting the Access is important to assess the system performance because the requesting application has to wait until the confirmation message is received from the blockchain system. Specifically, the blockchain system for requesting applications performs write transactions that receive the information from Write-Method invoked by the chaincode, Generates the hash value of the EHRs, validates based consensus mechanism, and updates the ledger. The experiment analysis and evaluation of the proposed system have been performed to test the performance using Hyperledger Caliper [34]. The admin peer of the blockchain network creates the genesis block, which is initialized with a caliper benchmark configuration file. This new genesis block file contains channel and chaincode for reading and writing the EHRs Documents installed and instantiated. The caliper configuration file is used for both proposed application that delivers the data for analysis. This caliper configuration file contains parameters such as Concurrent Users, Transaction Arrival Rate (TAR), and Blocksize with two performance metrics- throughput and Transaction Latency. The whole process is tested on one host to perform the experiment analysis [35–37].

The proposed blockchain network is evaluated with various performance metrics such as Block creation, Average Latency, Maximum Throughput and Timeouts, Certification Generation by MSP, and Number of Participants in the network. These metrics are evaluated for transaction processing power that has a significant bottleneck and impacts any blockchain network. First, throughput refers to the number of transactions performed by the blockchain network. This throughput is measured using

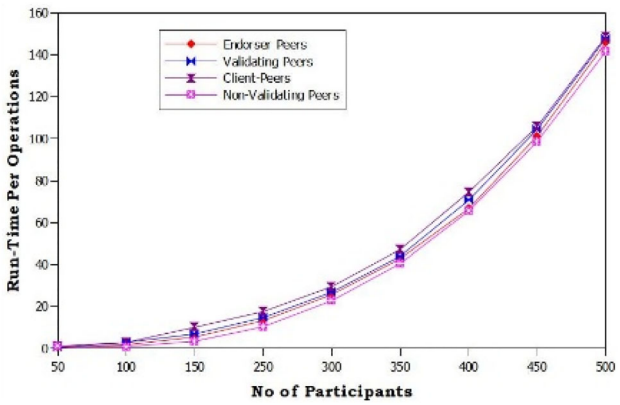


Fig. 5 Running time operation w.r.t number of users

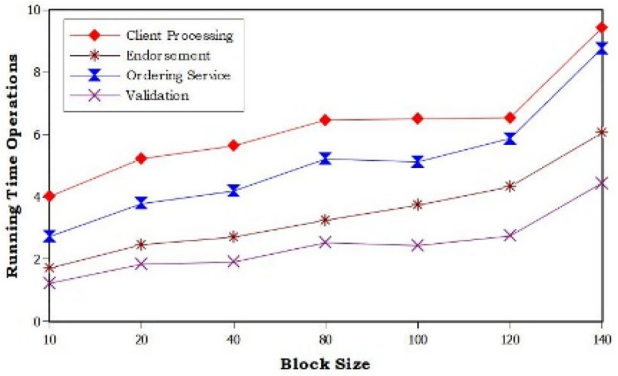


Fig. 6 Running time for proposed application by various peers of the blockchain network

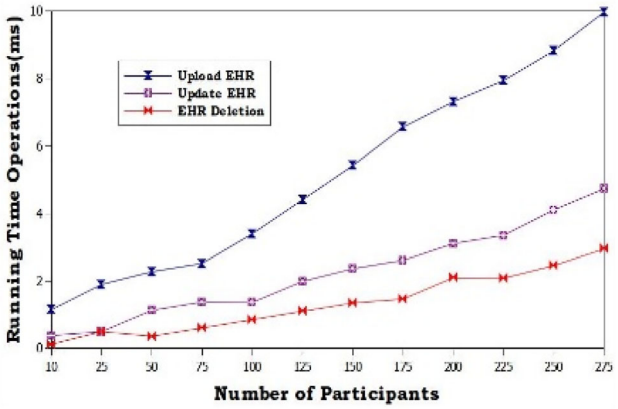


Fig. 8 Operations on EHRs

transactions per second (tps). The client processing throughput performed by the network for the processing EHR applications measures the numbers of EHRs uploading and deleting operations during the specified period. The other is the transaction throughput achieved by the network for the EHRs updating applications that measure the number of ledgers write operations committed during the specified period. The transactions per second (TPS) are measured for the full blockchain network size, i.e., all the nodes of the network will update the committed transaction to the ledger. The Hyperledger fabric can process 300 transactions per second. We set the queue length of the transaction to 1000 to produce stable and reliable results [38, 39].

The Fig. 5 shows the running time of various participants and their enrolment in the network. Each member makes the registered transaction with its new set of generated keys based on its identity (ID). After the members register with the blockchain network with the register transaction, the authentication and communication between the members are achieved. This figure also the time taken by the endorser, orderer, and validator. It is observed from the figure that for the 100 nodes, the time taken is 9 ms on average, and for 500 nodes, the time taken is 152 ms on average. Figure 6 shows the block creation, validation by the peers, and ledger write with the average queuing length of different block sizes. When the queuing length is small, the endorsement phase increases rapidly with an increased arrival rate, while block creation, validation, and ledger write remain stable. While queue length is considerable, there will be less waiting time for the validation/ledger write that generates committing block size transactions [40].

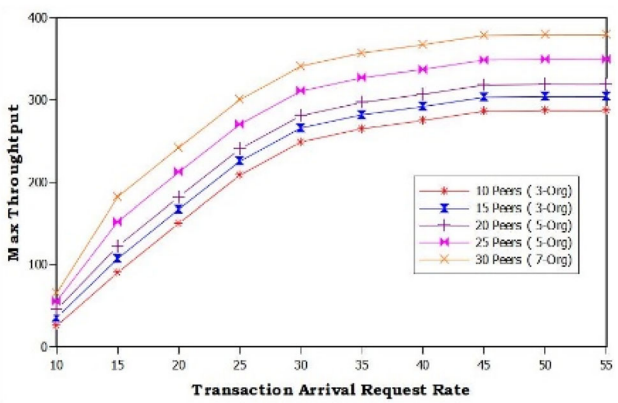


Fig. 7 Maximum throughput

The Fig. 7 shows the relation between the transaction throughput and transaction request rate with varying peers. It is observed that the throughput and transaction request rate have a strong correlation between them. Further, we observed that throughput increases with more peers and larger block sizes. For instance, the throughput linearly increases up to 55 tps of transaction request rate, which can be observed in both the Fig. 7 and of uploading updating and deletion transactions. The maximum reading throughput is reached at $\sim$ 290.4 tps. At the same time, the minimum reading throughput is achieved to $\sim$ 48.06 tps with an increased arrival rate of 55 tps. Hence, the larger blocksize with higher arrival rates results in higher throughput. The Fig. 8 shows another performance metric is transaction latency. The transaction latency is the amount of time the network takes to perform the transaction update across the network. This transaction latency is calculated to upload, update, and delete operations of EHRs transactions and respond to the client. The delay is measured from all the peers' process time of the network. The delay is measured in milliseconds (ms).

The Fig. 8 shows EHRs operation performed in the proposed network. The upload, update and delete operations of EHRs transactions are analyzed with the average execution time of the network. The upload applications spend more time performing the transaction proposal by invoking the transaction requiring endorsement and

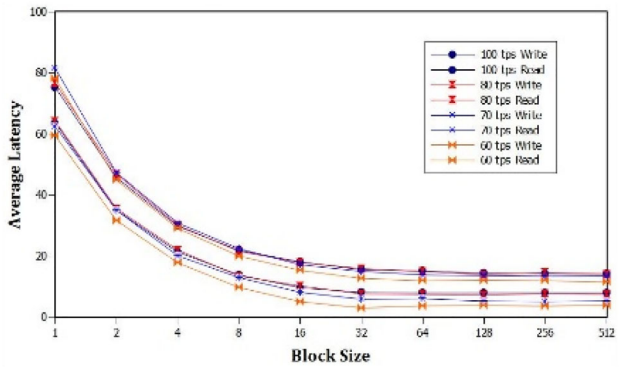


Fig. 9 Average latency

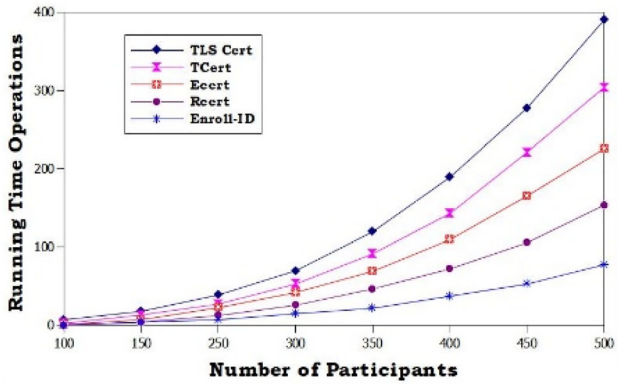


Fig. 10 Running time operation for generating the various certificates

waiting for the transaction response (endorsement) for creating the hash for the EHR documents from the endorser peer. This EHR applications require only 14.045 ms to complete phase one. Next, the same upload applications assemble the endorsement from the peers and send them to the ordering service. When the order peer receives an ordered transaction from the client application, it invokes the Validation System Chaincode (VSCC) to determine its validity. After validation, the EHR will be updated and committed to the ledger. The successful notification will be sent back to the client. This proposed application takes an average transaction execution time of around 3.85 s. The update application takes an average transaction execution time of approximately 1.65 s to update and send successful delivery notifications. Finally, the delete transaction application does not require any endorsement, and updates to the ledger take an average transaction execution time of around 0.34 s to delete the EHRs records.

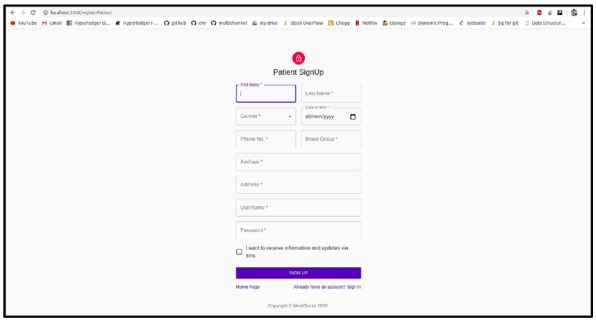
Figure 9 show transaction latency with varying block sizes. It is observed that, with an increase in block size for all arrival rates, the transaction latency is decreased quickly until the block size of 32 transactions per block. It is observed that from the same figure, the average latency for uploading and updating the EHRs transactions is 83.46 ms and 63.15 ms, with an arrival rate of 80 tps and 90 tps, respectively. The throughput reaches the saturation point when the overall latency increases rapidly.

The Fig. 10 shows the generation of certificates with varying participants in the network. It is observed that the time is taken to register, enroll and generate the digital certificates in the form RCert, ECert, TCert, and TLSCert to the participants and to install the chain code on every endorsing peer. It is observed from the figure that for the 100 nodes, the time taken is 12 ms, and for 500 nodes, the time taken is on average 152 ms. The proposed blockchain system takes less processing time to generate the certificates, suitable for the proposed application. The following Fig. 11 shows the abstract view of the proposed healthcare blockchain system.

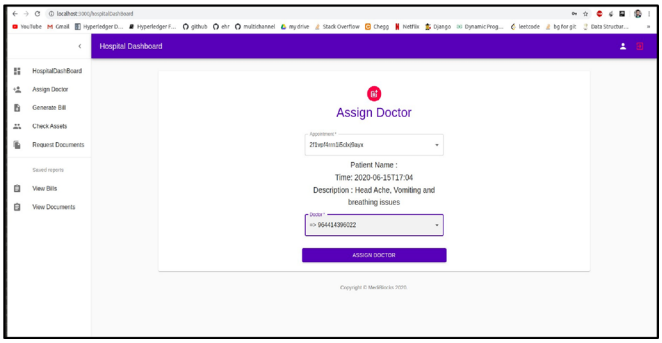
5 Research discussion

The study of e-Health applications will be able to offer the following services such as access to Electronic Health Records (EHRs) that helps the patient handling in an emergency, streaming of medical images, remote patient prescriptions, medication, and routine health check-up reminders, research, and education. However, these services should provide the availability of Electronic Health Records (EHRs) to health care providers, high flexibility and low operational cost, etc.

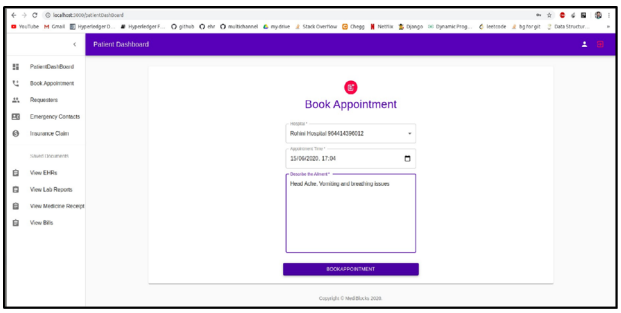
In this work, we addressed the following challenging issues. (1) Challenge-1: How to provide secure data exchange and anonymity. (2) Challenge-2: How to preserve the personal data privacy of the Patient Health Record. The proposed permissioned based blockchain framework for securely exchanging information provides a high assurance and guarantees the integrity of data provenance. The proposed system uses an ECDSA cryptosystem in a healthcare blockchain network that effectively allows the nodes to interact anonymously and securely to share healthcare information within a data-sharing network. Next, Data privacy issues are the main impact of any healthcare system; we integrated the privacy-preserving framework in the proposed healthcare blockchain system, which uses the Online/Offline framework. This framework is mainly used for two purposes: to speed up the verification process and to secure the storage of health records. Specifically, the proposed blockchain system can store two types of information. (1) “Online-Chain” data is directly stored on the blockchain network since online-chain does not require any heavy computations and stores less record information that speeds up the block process. (2) “Offline-Chain” data are stored in local databases, and the links/index of offline data are stored on the blockchain. Hence, the Online/Offline framework provide authenticity, fine-



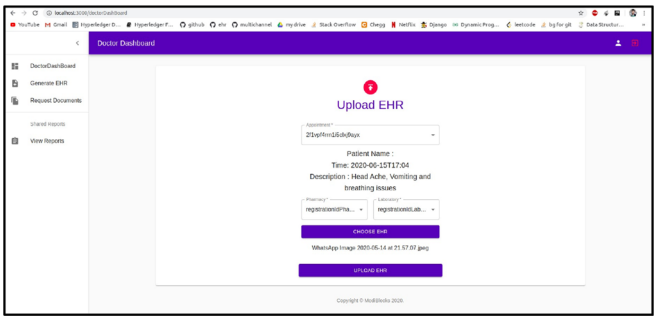
UI for Patient Registration



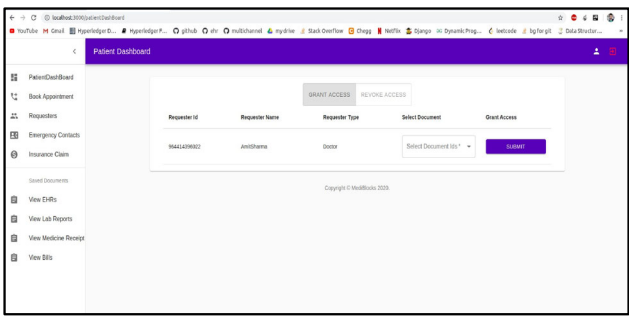
UI for Assigning Doctor



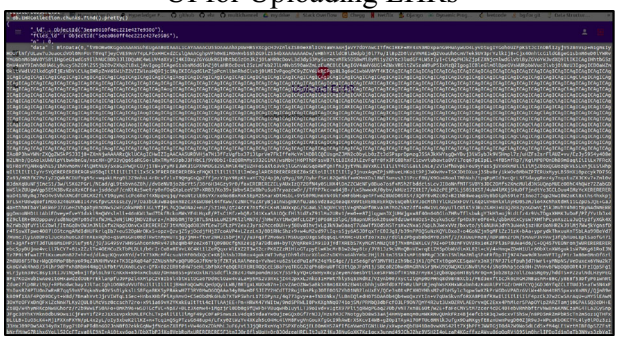
UI for Booking Appointment



UI for Uploading EHRs



UI for Granting Access



Document Stored in Base64 Format in MongoDB

Fig. 11 Abstract view of proposed healthcare blockchain system

grained data access control with confidentiality (i.e., all health records of all the customer remain fully confidential), signer anonymity, and public verifiability. In addition, this mechanism is mainly designed to prove ownership of customers/ health providers with a full public key on the healthcare blockchain, which prevents information exchanges to unauthorized users or health providers and does not disclose any patient information about its customers. Hence the proposed HealthCare Blockchain system is developed in open-source software-Hyper Ledger Fabrics, Composer that can be applied for other applications like pharmaceutical supply chain management, Identity Verification, validation, etc. Further, this work will be useful to health researchers and facilitate the faster discovery of new drugs and treatments.

6 Conclusion

This paper presents a secure architecture for exchanging health information using a permissioned blockchain network that stores electronic health records decentralized. This proposed work provides the complete supply chain for the medical treatment that decentralizes the integrity of protected health information and creates a secure environment for sharing the EHRs documents. The proposed solution leverages the Hyperledger Fabric, a permissioned

blockchain that establishes a secured and trusted network to all stakeholders, ensuring the integrity of protected health information and providing authenticity and health access control. Further, it decreases duplicate tests and unnecessary service, increasing accountability, preserving the crucial documents, limiting the unauthorized sharing of the EHRs documents, and offers lower costs across the care continuum.

Author contributions All authors have equally contributed to this work, read and agreed to the published version of the manuscript.

Funding Waleed Al-Numay acknowledges financial support from the Researchers Supporting Project number (RSP-2021/250), King Saud University, Riyadh, Saudi Arabia.

Data availability The data that support the findings of this study are available on the reasonable request from the first author (E Suresh Babu).

Declarations

Conflict of interest The authors declare that this manuscript has no conflict of interest with any other published source and has not been published previously (partly or in full). No data have been fabricated or manipulated to support our conclusions.

Research involving human participants and/or animals This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Ackerman, A., Chang, A., Diakun-Thibault, N., Forni, L., Landa, F., Mayo, J., van Riezen, R.: Blockchain and health IT: algorithms, privacy and data. Project PharmOrchard of MIT's Experimental Learning "MIT FinTech: Future Commerce.", White Paper August (2016).
2. Angraal, S., Krumholz, H.M., Schulz, W.L.: Blockchain technology: applications in health care. Circulation **10**(9), e003800 (2017)
3. Gordon, W., Wright, A., Landman, A.: Blockchain in health care: decoding the hype. NEJM Catal. **3**(1) (2017).
4. Kuo, T.T., Ohno-Machado, L.: Modelchain: decentralized privacy-preserving Healthcare predictive modeling framework on private blockchain networks (2018). arXiv:1802.01746.
5. Srinivasu, P.N., SivaSai, J.G., Ijaz, M.F., Bhoi, A.K., Kim, W., Kang, J.J.: Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. Sensors **21**(8), 2852 (2021)
6. Liang, X., Zhao, J., Shetty, S., Liu, J., Li, D.: Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1–5. IEEE (2017)
7. Rouhani, S., Butterworth, L., Simmons, A.D., Humphery, D.G., Deters, R.: MediChain TM: a secure decentralized medical data asset management system. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1533–1538. IEEE (2018).
8. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Decent. Bus Rev 21260 (2008).
9. Mandal, M., Singh, P.K., Ijaz, M.F., Shafi, J., Sarkar, R.: A tri-stage wrapper-filter feature selection framework for disease classification. Sensors **21**(16), 5571 (2021)
10. Ijaz, M.F., Attique, M., Son, Y.: Data-driven cervical cancer prediction model with outlier detection and over-sampling methods. Sensors **20**(10), 2809 (2020)
11. Culver, K.: Blockchain Technologies: A Whitepaper Discussing How the Claims Process can be Improved. In: ONC/NIST Use of Blockchain for Healthcare and Research Workshop. ONC/NIST, Gaithersburg, Maryland, United States (2016)
12. Goldwater, J.: The use of a blockchain to foster the development of patient-reported outcome measures. In: ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, MD: ONC/NIST (2016).
13. Miliard, M.: Blockchain use case: electronic health records. Could distributed ledger technology offer the promise of real-time EHR updates, seamless interoperability and protection from ransomware? (2018) https://www.healthcareitnews.com/news/qa-john-halamka-worldwide-trends-ai-blockchain-cloud-and-more.
14. Milojkovic, M.: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology (2018).
15. Elmisery, A.M., Rho, S., Aborizka, M.: A new computing environment for collective privacy protection from constrained healthcare devices to IoT cloud services. Cluster Comput. **22**(1), 1611–1638 (2019)
16. Li, D., Luo, Z., Cao, B.: Blockchain-based federated learning methodologies in smart environments. Cluster Comput (2021). https://doi.org/10.1007/s10586-021-03424-y
17. Liang, W., Ji, N.: Privacy challenges of IoT-based blockchain: a systematic review. Clust. Comput (2021). https://doi.org/10.1007/s10586-021-03260-0
18. Kanwal, T., Anjum, A., Khan, A.: Privacy preservation in e-health cloud: taxonomy, privacy requirements, feasibility analysis, and opportunities. Clust. Comput. **24**(1), 293–317 (2021)
19. Hölbl, M., Kompara, M., Kamišalić, A., Nemec Zlatolas, L.: A systematic review of the use of blockchain in Healthcare. Symmetry **10**(10), 470 (2018)
20. da Conceição, A.F., da Silva, F.S.C., Rocha, V., Locoro, A., Barguil, J.M.: Eletronic health records using blockchain technology (2018). arXiv:1804.10078.
21. Theodouli, A., Arakliotis, S., Moschou, K., Votis, K., Tzovaras, D.: On the design of a blockchain-based system to facilitate healthcare data sharing. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 1374–1379. IEEE (2018).
22. Alexaki, S., Alexandris, G., Katos, V., Petroulakis, N.E.: Blockchain-based electronic patient records for regulated circular healthcare jurisdictions. In: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6. IEEE (2018).
23. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and trustable electronic medical records sharing using

blockchain. In: AMIA Annual Symposium Proceedings, vol. 2017, p. 650. American Medical Informatics Association (2017).
24. Katuwal, G.J., Pandey, S., Hennessey, M., Lamichhane, B.: Applications of blockchain in Healthcare: current landscape & challenges (2018). arXiv:1812.02776.
25. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD), pp. 25–30. IEEE (2016).
26. Official documentation of hyperledger fabric https://www.hyperledger.org/projects/fabric
27. Goldreich, O. (2009). Foundations of Cryptography. Basic Applications, vol. 2. Cambridge university Press, Cambridge (2009).
28. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988)
29. Babu, E.S., Kavati, I., Nayak, S.R., Ghosh, U., Al Numay, W.: Secure and transparent pharmaceutical supply chain using permissioned blockchain network. Int. J. Logist. Res. Appl. 1–28 (2022).
30. Babu, E.S., Dadi, A.K., Singh, K.K., Nayak, S.R., Bhoi, A.K., Singh, A.: A distributed identity-based authentication scheme for internet of things devices using permissioned blockchain system. Expert Syst. e12941 (2022).
31. Babu, E.S., Srinivasarao, B.K.N., Kavati, I., Rao, M.S.: Verifiable authentication and issuance of academic certificates using permissioned blockchain network. Int. J. Inf. Secur. Privacy (IJISP) **16**(1), 1–24 (2022)
32. Hu, J., Liu, K. Raft consensus mechanism and the applications. In: Journal of Physics: Conference Series, vol. 1544, no. 1, p. 012079. IOP Publishing (2020).
33. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: Workshop on distributed cryptocurrencies and consensus ledgers, vol. 310, no. 4, pp. 1–4 (2016).
34. Sukhwani, H., Wang, N., Trivedi, K.S., Rindos, A.: Performance modeling of hyperledger fabric (permissioned blockchain network). In: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA) (pp. 1–8). IEEE (2018).
35. Krawiec, R.J., Housman, D., White, M., Filipova, M., Quarre, F., Barr, D., et al.: Blockchain: opportunities for health care. In: Proceedings of NIST Workshop Blockchain Healthcare, pp. 1–16 .
36. Qin, Q., Jin, B., Liu, Y.: A secure storage and sharing scheme of stroke electronic medical records based on consortium blockchain. BioMed Res. Int. (2021)
37. Shahnaz, A., Qamar, U., Khalid, A.: Using blockchain for electronic health records. IEEE Access **7**, 147782–147795 (2019)
38. Le Nguyen, B., Lydia, E.L., Elhoseny, M., Pustokhina, I., Pustokhin, D.A., Selim, M.M., Shankar, K.: Privacy preserving blockchain technique to achieve secure and reliable sharing of IoT data. Comput. Mater. Continua **65**(1), 87–107 (2020)
39. Mittal N., et al.: Using Blockchain to Address Interoperability Concerns in Healthcare. International Biopharmaceutical Industry (2018)
40. Kim, H., Song, H., Lee, S., Kim, H., Song, I.: A simple approach to share users' own healthcare data with a mobile phone. In: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 453–455). IEEE (2016)

Erukala Suresh Babu working as Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology, Warangal. He obtained his PhD from JNTU Kakinada, specializing in Networking and Security. He secured his M.Tech in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, and B.Tech in Computer Science and Engineering from JNTU Hyderabad. He has 16 years of Research, and Teaching in different levels of Professors, Associate and Assistant Professor. Currently, He works in the area of Blockchain Technology, Internet of Things, IoT Security, Wireless Networks, and Wireless Adhoc Network Security. Some of the areas where he published 65 refereed international publications including Journals and conference and 15 Book Chapters.

B. V. Ram Naresh Yadav is working as Associate Professor in the Department of CSE at JNTUHCES, Sultanpur a constituent college of JNTUH University, Hyderabad. He has 19 Years of Teaching Experience. He is Life member for ISTE Chapter, New Delhi. He held administrative responsibilities as J-hub, TEQIP-III and EDC Coordinator, NSS Coordinator Unit – II at JNTUHCES, Sultanpur a constituent college of JNTUH University, Hyderabad. He attended 06 and conducted 06 Academic Staff College Orientation / Refresher Courses. He had attended 26 and conducted 17 National level workshops. His areas of Interest are Big Data, Natural language processing, Data warehousing and Data mining, Network security, Image processing. He Published 46 papers at various National level and International Conferences and Journals. He visited Bangkok to present a Paper on FCDM approach for feature reduction in intrusion detection organised by International Institute of Engineers, Bangkok, 20-04-2015 to 21-04-2015.

A. Kousar Nikhath is currently working as Associate Professor in department of Computer Science & Engineering with specialization (Artificial Intelligence and Machine Learning and Internet of Things) at VNRVJIET, Hyderabad. She is into teaching profession for the past 18 years. She has done her Ph.D in CSE from KL University. Also, She is certified Professional in Advanced Machine Learning and Artificial Intelligence from IIIT, Hyderabad. She has published nearly 22 papers in reputed indexed national and international Journals/Conferences. Her research interests include

Text mining, Artificial Intelligence and Neural Networks, Machine Learning. She has guided students at PG and UG level.

Soumya Ranjan Nayak is currently working as Assistant Professor at Amity School of Engineering and Technology, Amity University, Noida, India. He received his Ph.D. degree in Computer Science and Engineering under MHRD Govt. of India fellowship from CET, BPUT Rourkela, India; with preceded degree of M. Tech and B. Tech degree in Computer Science and Engineering. He has published over 80 articles in peer-reviewed journals and conferences of international repute like Elsevier, Springer, World Scientific, IOS Press, Taylor & Francis, Hindawi, Inderscience, IGI Global, etc.. Apart from that, 12 Book Chapter, 6 Books and Six Indian patent (two patent granted) and two International patent (two patent granted) under his credit. His current research interests include medical image analysis and classification, machine learning, deep learning, pattern recognition, fractal graphics and computer vision. His publications have more than 600 citations, of h index of 15, and i10 index of 21 (Google Scholar). He serves as a reviewer of many peer-reviewed journals such as Applied Mathematics and Computation, Journal of Applied Remote Sensing, Mathematical Problems in Engineering, International Journal of Light and Electron optics, Journal of Intelligent and Fuzzy Systems, Future Generation Computer Systems, Pattern Recognition Letters, etc. He has also served as Technical Program Committee Member of several conferences of international repute.

Waleed Alnumay received the bachelor's degree in computer science from King Saud University, Riyadh, Saudi Arabia, in 1993, the master's degree in computer science from the University of Atlanta, Atlanta, GA, USA, in 1996, and the Ph.D. degree in computer science from Oklahoma University, Norman, OK, USA, in 2004. He is currently working as an Assistant Professor with the Mobile Networking in Computer Science Department, King Saud University. He has published research articles in reputed international conferences and journals. His main research interests include computer networks and distributed computing that includes but not limited to mobile ad-hoc and sensor networks, information-centric networking, and software-defined networking