Cluster Computing (2024) 27:1047–1061
https://doi.org/10.1007/s10586-023-04038-2

Attribute-based access control scheme for secure storage and sharing of EHRs using blockchain and IPFS

Jasleen Kaur[1] · Rinkle Rani[1] · Nidhi Kalra[1]

Received: 15 July 2022 / Revised: 22 April 2023 / Accepted: 17 May 2023 / Published online: 9 June 2023

Abstract
Medical records are one of the crucial documents and a significant asset for anyone seeking treatment. Electronic health records (EHRs) have made a dynamic shift by making them easier to manage, facilitate and share among various stakeholders such as doctors, lab technicians, and insurance agents. EHRs are vulnerable to hacker, cybercriminal attacks, and data breaches. Once compromised, health records cannot be retrieved. As a result, patients must have control over who gets their EHRs, when they get them, and where they get them. To address the aforementioned issue, this paper proposes a blockchain-based secure record-keeping and trustworthy sharing system. In order to do this, a distributed off-chain storage architecture for large-scale medical data storage is developed, which overcomes the drawbacks of on-chain data storage and enhances scalability. The distributed storage, i.e., InterPlanetary File System, is a content-addressable storage that ensures the integrity of the content such that a slight modification in the stored EHR records results in a change in the obtained hash value. Furthermore, a Ciphertext Policy Attribute-Based Encryption (CP-ABE) algorithm integrated with blockchain technology is designed for fine-grained access control, allowing only authorized users to access specific EHR data based on their attributes. The combination of CP-ABE with blockchain technology provides a tamper-proof and verifiable audit trail of all data access and updations made to EHRs. This enhances accountability and ensures that the patients or owners can track and verify all actions taken on the data. To implement the proposed system, the Remix-Ethereum IDE is used. Smart contracts (SCs) are designed with access permissions so patients have complete control over their records. The scalability and immutability of the system is ensured by storing the hash of the encrypted EHRs on the blockchain and the actual encrypted records on IPFS. The security analysis of the proposed system is carried out by evaluating its resistance to various attacks. Additionally, potential security flaws in the proposed SCs are investigated using the Oyente tool. Different test cases are presented to demonstrate the functionality and cost analysis of the proposed system.

Keywords Blockchain · Electronic Health Record · InterPlanetary file system · Ethereum · Smart Contract · Attribute Based Encryption

1 Introduction

Health Records are valuable assets for every patient. A health record consists of the medical history of the patient, such as disease information, prescriptions, lab test reports, vitals (body temperature, pulse rate, blood pressure), financial information (medical bills, debit/credit card, and bank account details) along with personal information such as name, age, physical signs (height and weight). Though it should be kept personal, but sharing becomes inevitable such that appropriate treatment can be given.

✉ Jasleen Kaur
jkaur_phd18@thapar.edu

Rinkle Rani
raggarwal@thapar.edu

Nidhi Kalra
nidhi.kalra@thapar.edu

[1] Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

Physical health records are difficult to coordinate among various stakeholders and prone to lose or damage.

Electronic Health Record (EHR) systems came into the picture with the advancements in technology. EHRs are much more dynamic and are easier to manage, facilitate and share among various stakeholders such as doctors, patients, etc. For storing and managing patient's medical data, these systems have been utilized by several hospitals, medical institutes, and dispensaries. The report shows a rise in accepting EHR systems from 9.4 to 83.8% across non-federal acute care hospitals from 2008 to 2015 [1]. The American Hospital Association (AHA) found that approx. 96% of non-federal acute care hospitals and nearly four out of five (approx. 78%) office-based physicians have adopted certified EHR systems in a subsequent survey done in 2021. This is a significant improvement over 2011, when only 28% of hospitals and 34% of physicians used EHR systems [2]. Compared to the traditional manual paper-based medical records, EHR systems are more efficient, organized, and less error-prone.

Hospitals or third-party organisations are primarily in charge of running and maintaining the EHR systems. Sharing EHRs outside the hospitals (to other hospitals) and clinics is still not easy due to strict regulations and policies, as EHRs contain sensitive and confidential information [3]. Also, a patient visits various hospitals, doctors, clinics, and laboratories during treatment, due to which medical data is scattered at different places. In some cases, patient is required to undertake the same tests that have previously been performed elsewhere. As a result, better clinical decisions can be made by combining patient data from many sources into a single, unified view. The medical records of patients must be accurate, complete, and up-to-date, to ensure timely and high-quality healthcare services.

Healthcare data contain confidential and private information, susceptible to security menaces such as information theft, unauthorized disclosure, tempered or shared without the patient's consent. As of now, EHRs are prone to various attacks and data breaches by crackers and cybercriminals. One can change the leaked ATM pin or close the bank account, but private or confidential information pertaining to a patient's health records cannot be entirely retrieved or taken back after it has been disclosed. Even if the content is removed from the original source, it may have already been distributed, copied, or accessed by unauthorized users. According to the Health Insurance Portability and Accountability Act (HIPPA) medical records breach report, approximately 1,322,211 records were disclosed impermissibly or stolen in July 2020 [4]. So, there arise critical security and privacy issues. Thus, an efficient smart health care system needs to securely store, fetch, and share medical records with patient consent. Various researchers proposed many solutions [5]. Xhafa et al. [6] proposed privacy-aware health care records (PHR) system using cloud computing. Outsourcing confidential data is always at risk. Thus, the authors [7] gave directions related to the privacy & security of records to healthcare and cloud service providers. Moreover, encryption algorithms or public-key cryptography can be used to provide confidentially, integrity, and data access control. Attribute-based encryption (ABE) extends the concept of encrypting data for a particular user. In this, the user can only access data if the private key and the attributes of the cipher-text match. The idea of ABE was first presented by Sahai and Waters [8] and by Goyal et al. [9]. ABE is mainly classified into Key Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In KP-ABE, private keys are produced based on an access tree or policy that specifies the permissions associated with the interested user, and data is encoded/ encrypted using set of attributes. CP-ABE utilizes access trees/policies for encrypting data, and private keys of users are created using attributes. Recently, many authors used these encryption techniques to secure private healthcare data sharing. Mubarakali [10] proposed an attribute-based and blockchain-based approach for storing patient's health data on cloud servers and then transmitting it to doctors and insurance agents based on policies defined to make the system secure and robust. Similarly, Li et al. [11] suggested a patient-centric, secure, and scalable PHR data-sharing architecture using both KP-ABE and CP-ABE approaches. For outsourcing data and providing fine-grained access control, authors use ABE with proxy re-encryption technique [12].

As EHRs are valuable assets for patients thus, sharing them with other stakeholders on demand must be in the hands of patients. Patients must have complete control over whom, when, and what they share. They can observe all the activities anywhere, anytime, instead of relying on any centralized infrastructure. Thus the trend shifts to decentralized technologies such as blockchain, which is gaining popularity nowadays and widely adopted in different areas [13]. Blockchain is a distributed ledger with the attractive properties of immutability, tamper-resistance, and traceability of the log, which records all CRUD operations performed on the data as transactions. Transactions are stored in cryptographically linked blocks to form a chain of blocks. In addition, the emergence of various blockchain platforms such as Ethereum, Hyperledger Fabric, Sawtooth, and smart contracts [14] has improved blockchain so that it is used in various fields, especially in healthcare [15]. Simply Vital Health, Inc. (SVH) [16] originated as a service provider for building decentralized applications on the Health Nexus protocol. They developed a platform called connecting care for sharing patient records among different stakeholders.

Another example is EncrypGen [17], which securely stores and shares genomic data using blockchain technology with various researchers. The authors [18] proposed a framework named Bheem for effective storage and management of EHRs. In their work, they also investigate the privacy and security considerations in healthcare 4.0, and storing voluminous data on the blockchain increases cost. In a similar work, the authors [19] utilize the Ethereum Platform for Blockchain implementation. They proposed a model for secure storage of EHRs at distributed storage called InterPlanetary File System (IPFS). In contrast, the work done by authors [20] uses Hyperldger, a permissioned based blockchain platform along with proxy re-encryption algorithm, for providing privacy protection to PHR data. Similarly, for preserving privacy, a Hashed based access authentication scheme is suggested by the authors [21]. Thus, several studies utilizes a system based on blockchain to create, access, and manage EHRs incorporating various cryptographic techniques [22–24]. Ali et al. [25], put forward a secure search and keyword-based access approach for the healthcare system. To achieve security and privacy in patient healthcare networks with higher efficiency and lower cost, a blockchain-based framework is proposed utilizing both, Ethereum and Hyperledger platforms [26]. The authors of [27] implemented improved CP-ABE based authentication scheme to securely collect, upload and download EHR data from cloud. The authors of [28, 29] also proposed a light-weight, hybrid, deep learning based privacy preserving system for Internet of Medical Things (IoMT)-based Cyber-Physical Systems (CPS). Table 1 describes the summary of the related work.

This paper presents a blockchain model built on the Ethereum platform that integrates IPFS for efficient record storage and secure sharing with access control. CP-ABE algorithm integrated with blockchain technology is designed for fine-grained access control,that allows only authorized users to access specific EHR data based on their attributes. The combination of CP-ABE with blockchain technology provides a tamper-proof and verifiable audit trail of all data access and updations made to EHRs. This enhances accountability, non-repudiation and ensures that the patients or owners can track and verify all actions taken on the data. As a result, the proposed system eludes the risk of single-point failure as in existing centralized systems by using decentralized platforms, Ethereum and IPFS. Smart contracts are designed with access permissions so patients have full control over their records. CP-ABE-based encryption/decryption helps securely store and share records with other stakeholders. The scalability of the system and immutability of the records is also achieved by storing the hash of the encoded EHRs in the blockchain and actual encoded records in the distributed off-chain storage, i.e., IPFS. Thus, the proposed system addresses the challenges of data integrity, security, scalability, and immutability.

Table 2 presents the feature-based comparison of the proposed work with the related work. The features included are the patient-centric model for making patients the data owners, content addressable distributed storage for storing a large amount of EHR data, and additional cryptographic technique for providing privacy protection and fine-grained access control. It also includes analyzing the implemented code for vulnerability assessment to prevent from various attacks, performance analysis to check the effectiveness and feasibility of the proposed system, and the blockchain system security analysis to check the scheme is resistant to various attacks.

1.1 Our contribution

- A decentralized and distributed system for secure healthcare data storage and sharing using blockchain technology and IPFS is proposed. The proposed EHR administration design incorporates a wide range of stakeholders from the healthcare sector.
- The proposed system utilizes Cipher-text Policy Attribute-Based Encryption (CP-ABE) algorithm and designed Smart Contracts for fine-grained access control, that allows only authorized users to access specific EHR records based on their attributes. By assuring user validity and a verifiable audit trail of every data access and update made to EHRs, it will aid in making the system tamper-proof and confidential.
- To enhance the system's scalability, IPFS distributed storage is employed to store encrypted records of patients and hashes on blockchain.
- Qualitative assessment of the proposed system is performed by considering different scenarios. Different test cases are presented to show the proper functioning of the system. In the end, cost analysis and security analysis is performed.

The remaining paper is structured as follows. Section 2 represents the preliminaries used for the execution of the proposed system. A concise explanation of the designed architecture is introduced in Sect. 3. Section 4 presents the outcomes and discussions. Section 5 concludes the paper.

2 Preliminaries and background

This section describes the details of the proposed system along with the preliminaries used, such as Ethereum, InterPlanetary File System (IPFS) and CP-ABE.

Table 1 Summary of the related work

Reference (s)	Year	Objective (s)	Platform	Storage	Additional cryptographic technique applied	Access control mechanism	Performance evaluation parameter (s)
[18]	2018	To provide efficient storage and maintenance of EHRs using blockchain	Ethereum	Offchain	Not Mentioned	Implemented using Smart Contracts	Not Mentioned
[19]	2019	To design a system using blockchain technology for EHR and provide secure storage to EHRs	Ethereum	Off-chain (IPFS)	Not Mentioned	Role based smart contracts are implemented	Execution time, Throughput, and Latency
[20]	2019	To present a blockchain model for preserving the privacy of PHR data	Hyperledger fabric	Cloud	Proxy re-encryption algorithm and ABE	Access list and their metadata is defined for access control	Time taken by various cryptographic algorithms considering varying size of data, and users
[10]	2020	To monitor the healthcare services in Cloud and securely transfer the data from patients to Cloud Storage by the proposed model named SRHB (Secure Robust Healthcare Blockchain).	Implemented using NetBeans 8.2, and Jelastic cloud environment	Cloud	ABE	ABE based access contol	Success rate, average delay, and execution time
[21]	2020	To preserve privacy of PHR data, Hashed based access authentication scheme is proposed	–	Cloud	Hashed Based CP-ABE	Attribute based access policy	Encryption/ decryption time
[22]	2021	To propose lightweight deep learning-based privacy preservation model for IIoMT	Ethereum	Cloud	ABE	Implemented using attribute based access control	Analysis of access policies w.r.t attributes and certificate authority, etc
[23]	2021	To achieve the secure storage, trustworthy sharing, access management, and safeguarding privacy of healthcare records, an EHRchain named blockchain-based system is proposed	Hyperledger Fabric	Offchain (IPFS)	Attribute-based and homomorphic encryption	Attribute based access control policy	The time taken by by various proposed algorithms are discussed
[26]	2021	To achieve security & privacy in patient healthcare networks with higher efficiency at lower cost, a blockchain based framework is proposed	Hyperledger and Ethereum	Cloud	Ring Signature + ABE	Implemented using Smart Contracts	Execution time
[24]	2022	To propose a Blockchain and CP-ABE based secure storage and access control along with revocation process	Java Based Blockchain network Design	Cloud	CP-ABE	Implemented using CP-ABE	Time Taken by key generation, encryption algorithms and etc
[25]	2022	To propose a novel approach based on the neural network and group theory that efficiently detects invasions in the IoT network. Further, a secure keyword-based search using blockchain and homomorphic techniques is applied	Hyperledger Fabric	Cloud	Homomorphic Encryption	Implemented using attributes and features	Encryption/ decryption time, latency, and throughput

Table 1 (continued)

Reference (s)	Year	Objective (s)	Platform	Storage	Additional cryptographic technique applied	Access control mechanism	Performance evaluation parameter (s)
[27]	2022	To provide secure collection, download and upload EHRs from cloud is proposed in this work	–	Cloud	CP-ABE	Improved CP-ABE based access control Authentication schemes	Encryption / Decryption execution time w.r.t varying file sizes
Proposed System	2023	To propose an attribute based access control for secure storage and sharing of EHRs using Blockchain and IPFS	Ethereum	IPFS	CP-ABE	CP-ABE and smart contracts based access control	Encryption and Decryption Execution time w.r.t varying number of attributes, file sizes

Table 2 Feature-based comparison of existing and proposed work

Features → Reference(s)↓	Patient centric	Content addressable Distributed storage	Additional privacy preservation policy	Performance analysis performed	Code vulnerability analysis	Blockchain security analysis (attack resistant)
[18]	*Yes*	*No*	*Yes*	*No*	*No*	*No*
[19]	*Yes*	*Yes*	*No*	*Yes*	*No*	*No*
[20]	*Yes*	*No*	*Yes*	*Yes*	*No*	*No*
[21]	*Yes*	*No*	*Yes*	*Yes*	*No*	*No*
[24]	*No*	*No*	*Yes*	Yes	*No*	*Yes*
[27]	*No*	*No*	*Yes*	*Yes*	*No*	*No*
Proposed system	*Yes*	*Yes* (IPFS)	*Yes*	*Yes*	*Yes*	*Yes*

2.1 Ethereum

Ethereum, a distributed blockchain network, was officially introduced in 2015 by Vitalik Buterin [30]. He uses the concept of blockchain which was recently used in the well known digital currency Bitcoin [31] by Santoshi Nakamoto. The thought behind Ethereum was to design a smart contract (SC) for the trustless system that would be open source and has a feature for programmable blockchain. For customizing own blockchain and writing SCs, a language named Solidity is provided by Ethereum. SCs can be defined as autonomous and immutable programs for verifying and executing an agreement's terms or set of rules. They are autonomous, thus reducing the cost of creating and maintaining a centralized database.

Ethereum virtual machines (EVMs) are used for the execution of SCs. Ether (ETH) is a cryptocurrency given as an incentive to miners in proof of work (PoW) based systems for their computation work and adding blocks to the blockchain. Gas is defined as the unit of measure of computation work required for executing functions on the Ethereum blockchain network. The gas unit is Wei where $1Wei = 10^{-18}$ ETH. The transaction cost is measured as Ether = gas used * gas price. There are two kinds of Ethereum accounts available: the first one is an externally owned account which is controlled externally by using private keys, and the second is a contract code account where SCs are deployed to the network and managed by contract code.

2.2 Distributed storage

IPFS, known as InterPlanetary File System, is a peer to peer (P2P) distributed storage used as an off-chain repository for keeping extensive medical data. Content-based addressing is used for storing files like images, pdfs and videos [32].

2.3 Ciphertext policy attribute- based encryption (CP-ABE)

Amit Sahai and Brent Waters initially suggested the concept of attribute-based encryption (ABE) [8]. Attributes determine a user's private/secret key and ciphertext. Decryption is allowed if the attributes of the secret key match with the ciphertext attributes. This work utilizes CP-ABE architecture proposed by John Bethencourt et al. [33] in 2007. It mainly contains four functions: *Setup(), KeyGen(), Encryption(), and Decryption()*. In this, the Master Public Key is same for every user. Secret keys are generated by executing the KeyGen() algorithm, and for each user, these are related to the set of attributes. During Encryption of documents, access policies are embedded and described over attributes using logical operators [20].

For specifying access policies, a tree access structure is defined such that the internal nodes of trees represent logical operators and external nodes represent different attributes. The decryption of a ciphertext is possible only if user attributes specified during KeyGen() satisfy the policy defined. Access policies are logical formulae created using attributes from an attribute list merged with logical operators. Logical operators are AND (n of n), OR (1 of n) and k of n threshold gates that must match k out of n attributes in the access policy.

3 Proposed system

In this section, the architectural and implementation details of the proposed system are presented and discussed. The following symbols are used to describe the system depicted in Table 3.

3.1 Proposed system architecture

Figure 1 presents the proposed system architecture. The proposed approach uses the Ethereum platform for developing blockchain and IPFS storage for securely keeping EHRs. All the entities, such as patients, doctors, etc., must be registered on the Ethereum blockchain network except the distributed storage. Following are the main components.

1. **Regulatory Authority:** A government body or a trusted public authority responsible for registering users such as doctors, patients, and Third-Party Administrators (TPA). It initiates the system by deploying smart contracts and executing the *Setup()* function. After initialization, users request RA for registration with their attributes such as role, name, age, date of birth, location, etc. They are assigned a unique address when they are registered on their request. The authority returns private keys to users corresponding to attributes submitted by executing the *KeyGen()* function.
2. **Patients:** Patients are a crucial part of our system. The cipher policy attribute-based encryption and smart contracts are utilized to design access control functions to make patients the real owners of their data. These functions such as *grantPermission()*, *create &UploadRecords()* and *revokePermission()* are executed in response to a request submitted by doctors/viewers to patients. Thus, in addition to having access to their healthcare records, they also control the security issues of unfair data sharing and data-stealing among various users by managing access attempts through smart contracts.
3. **Doctors:** Records generated after treatment or consultation are uploaded to IPFS by the doctor. The records are first encrypted by invoking the *Encryption()* function, including the access policy (AP) satisfied by patients and then uploaded to IPFS. Finally, the resulting hash of added encrypted EHR is put on the blockchain and later verified by the patient for data integrity.
4. **Viewers:** Viewers are the requester's requesting to view patients' medical history or records. They may be registered doctors, clinicians, lab technicians, or insurance agents. The patients send the hash of the encrypted records by defining the access policy after authentication. Only the viewers satisfying the procedure can down and decrypt them locally using the *Decryption()* function.
5. **Distributed Off-chain Storage:** Encrypted EHRs are placed at IPFS to prevent unauthorised access and data leakage. Records are encrypted before they are placed on IPFS using CP-ABE.

Table 3 List of symbols used

Symbol(s)	Description
Γ_E	Unique Ethereum blockchain address of the Entity E, where E=RA, P, D, V
$\Gamma_{RA}, \Gamma_P, \Gamma_D, \Gamma_V$	Unique Ethereum blockchain address of the Regulatory Authority (RA), Patient (P), Doctor (D), and Viewer (V)
msg.sender	Ethereum Address of the user who currently interacts with the Smart Contract (SC)

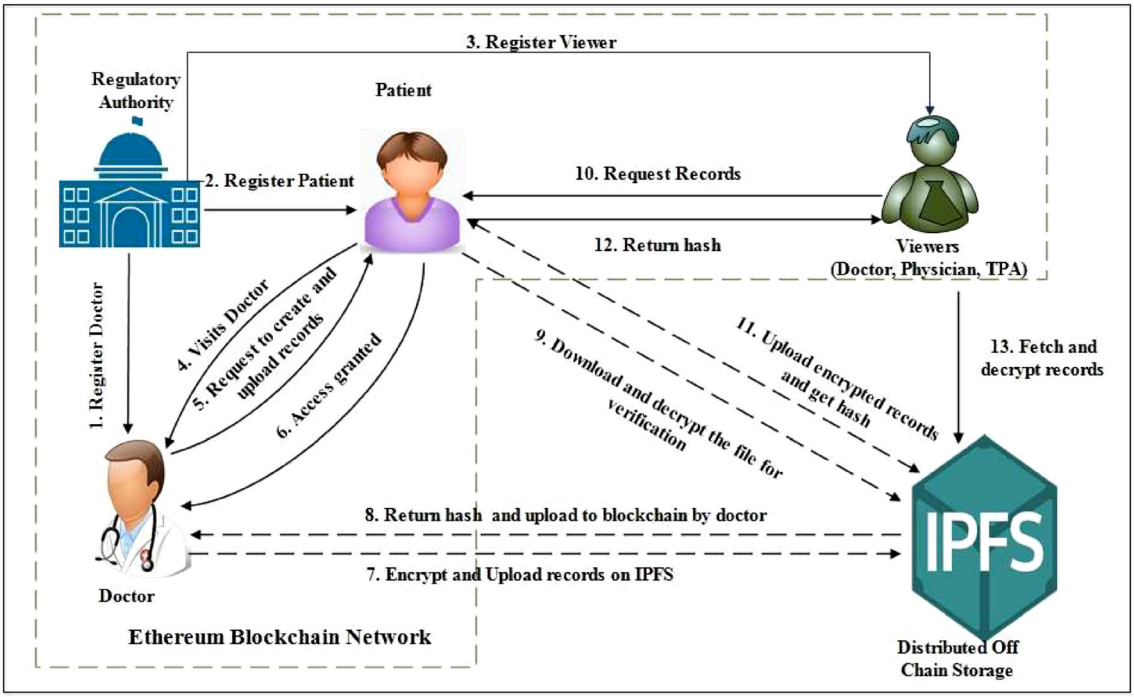

Fig. 1 Architecture of the proposed system

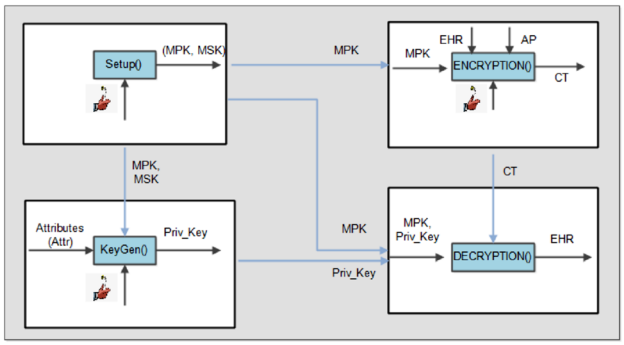

Fig. 2 Workflow of CP-ABE algorithm

3.2 Ciphertext policy attribute based encryption (CP-ABE)

In this work, CP-ABE, proposed by Bethecourt et al. [33] is employed. This approach uses attributes to define user credentials, and specifying policy by encrypter determines who can decrypt the data. The Charm-Crypto tool [34] is used for implementation containing the following four functions. Figure 2 depicts the work-flow of the CP-ABE algorithm.

1. **Setup() ⟶ MPK, MSK:** The ***Setup()*** function initiated by the RA takes only implicit parameter as an input and generates master public key MPK and a master secret key MSK as an output. First, the algorithm selects bilinear group G_0 of prime order p with generator g. Then select random integers $\gamma, \delta such that \gamma, \delta \epsilon Z_p where Z_p$ is integer group. Then finally compute $MPK = G_0, g, h = g^{\gamma}, e(g,g)^{\gamma}$ and $MSK = (\delta, g^{\gamma})$
2. **KeyGen(MPK, MSK, Attr) ⟶ Priv_Key:** The ***KeyGen()*** function takes MPK, MSK and Attr (attributes set) as an input. It outputs a secret key Priv_Key for each user w.r.t Attr as follows: $Priv\_Key = (K = g^{(\gamma+i)\div\delta}, \forall a \epsilon Attr : K_a = g^i.H(a)^{i_a}, K'_a = g^{i_a})$ where the algorithm randomly selects $i \epsilon Z_p$,and the $i_a \epsilon Z_p, \forall a \epsilon Attr$.
3. **Encryption(MPK, EHR, AP) ⟶ ct:** The ***Encryption()*** function takes MPK, a message EHR and access policy AP as an input. The algorithm encrypts the EHR and produces a ciphertext *ct*, allowing only users with a set of attributes satisfying the access policy can decrypt the EHR. A polynomial q_x is selected in a top-down way, beginning from root node of the tree AP, $\forall x \epsilon AP, x$ is node. A degree d_x of the polynomial q_x is assigned such that is equal to k_{x-1},where k_x is threshold value of node. For root node, $q^{(0)}_{root} = n$, a random number $n \epsilon Z_p$ is picked, and d_{root} is chosen to define polynomial q_{root} entirely. For a node x in the tree, $q^{(0)}_x$ is defined as: $q^{(0)}_x = q^{(0)}_{parent}$. Similarly, chooses d_x to define polynomial q_x. Ciphertext ct is computed as follows: $ct = (AP, \check{C} = EHR.e(g,g)^{\gamma n}, C = h^n \quad \forall k \epsilon K : C_k =$

$g^{q_{k0}}, \grave{C}_k = H(attr_{(k)})^{q_{k0}})$ where k is leaf nodes containing Attributes in AP.

4. **Decryption(MPK, Priv_Key, ct) ⟶ EHR:** The ***Decryption()*** function takes MPK, Priv_Key and *ct*, which contains AP as an input. Only the requester satisfies the policy can decrypt ct and get EHR as an output. This procedure starts simply from root node. If set of attributes s satisfies policy then A = Decrypt_node(ct, Priv_Key, root) = $e(g,g)^{rq^{(0)}_{root}} = e(g,g)^{rs}$, and decryption function as below [33].

$$\check{C}/(e(C,D)/A) = \check{C}/(e(h^s, g^{(\gamma+r)/\delta})/e(g,g)^{rs}) = EHR$$

3.3 Working of the proposed approach

Figure 3 shows the working of the proposed approach as an interaction diagram. The figure describes the interaction between different entities involved in the network for securely storing and sharing EHRs after deployment of the Smart Contract. The following are the steps followed:

- **Step 1–3:** Initially, the regulatory authority registers all entities, i.e., patients, doctors, and viewers, participating in the network by executing functions defined in SC and storing their information in the Blockchain. Each registered entity has a unique address having 40 hexadecimal characters. The address has the prefix "0x" followed by the 20bytes of the Keccak-256 hash of the public key.
- **Step 4:** The patient visits a doctor for treatment or consultation and executes a function to inform the doctor about the patient's visit.
- **Step 5–6:** After treatment, the doctor requests permission to create and upload EHRs via SC, and then the patient grants invoking *grantPermission()* present in Algorithm 1.
- **Step 7–8:** The doctor encrypts EHRs using *Encyption()* function and uploads the encrypted EHRs to IPFS in Step 7. Then the hash value $H_{(sha256)}(EHRs)$ of 32 bytes corresponding to the submitted EHR is generated by IPFS and return to the doctor. The doctor updates the patient with this hash value and Record_info such as file id, file description, date, and design (doctor_sign) by calling *create &UploadRecords()* Algorithm 2 in Step 8. The transaction is recorded into the blockchain network.

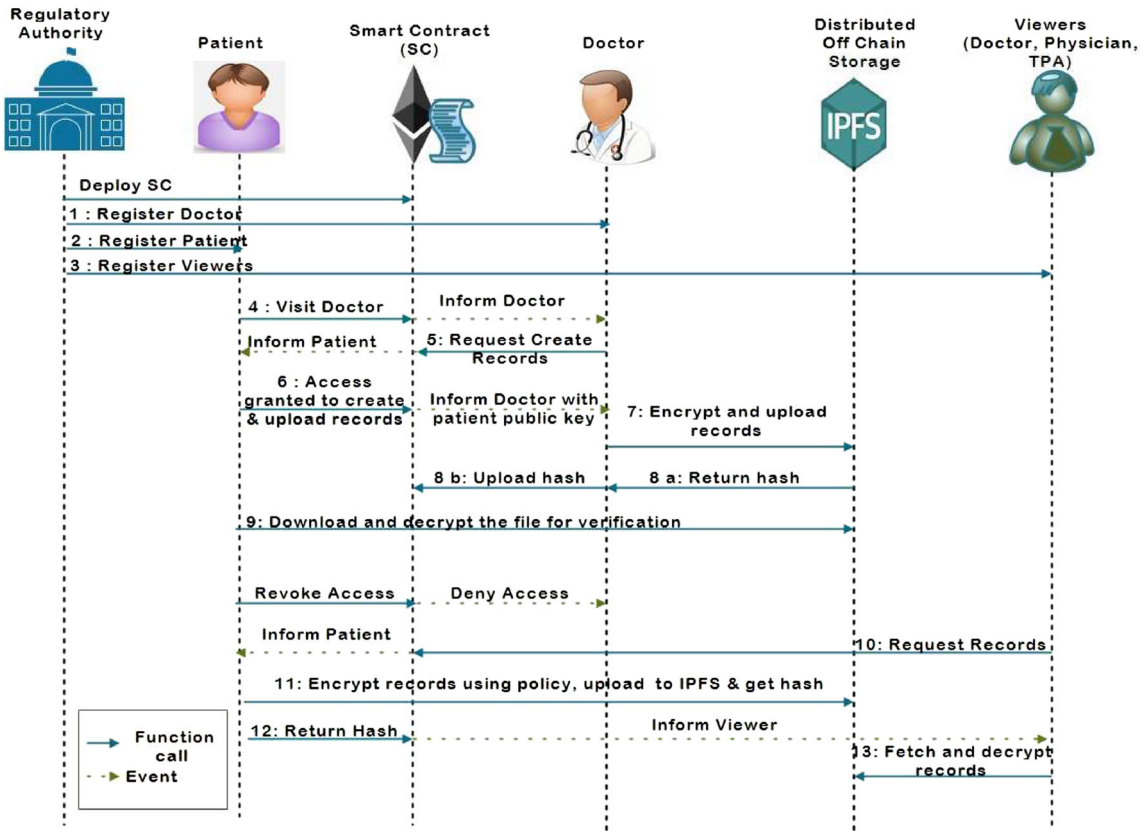

Fig. 3 Workflow of CP-ABE Algorithm

Algorithm 1 grantPermission( )

Input: Γ_P, Γ_D
Output: Permission Granted to Doctor with Γ_D

1: **Require:** Only patient can give permission i.e $msg.sender == \Gamma_P$
2: check if already has access, if no i.e $canCreate[\Gamma_D] == false$
3: then $canCreate[\Gamma_D] = true$
4: in patients $doctor\_access\_list \leftarrow push\Gamma_D$
5: in doctors $patient\_access\_list \leftarrow push\Gamma_P$
6: **Emit:** inform doctor about grant permission

Algorithm 2 create&UploadRecords ( )

Input: Γ_D address of doctor, Record_ info, EHRs
Output: Permission Granted to Doctor with $\Gamma_D$transaction recorded in the blockchain network, and inform patient about record added

1: **Require:** Only doctor can upload records
2: If $msg.sender == \Gamma_D$, and has permission
3: Then encryptEncrypt(EHR) and upload to IPFS, get hash value
4: push Record_info and hash into EHR as mappings( )
5: push mappings
6: **Emit:** inform the patient about record added

- **Step 9:** Then, the patient download and decrypts the EHRs using the *Decryption()* function for data authentication. After the visit, they can revoke access for creating and uploading EHRs by executing the *revokePermission()* function defined in Algorithm 3.
- **Step 10–12:** A viewer can request records, and then the patient first checks for the viewer's authorization by executing Algorithms 4 and 5. If authorized viewer, the patient uses the Encryption function with policy defined for encrypting and uploading requested EHRs to IPFS and provides him with the hashes as shown in Steps 10, 11, and 12. In the end, the viewer downloads and decrypts the records if they satisfy the policy. Thus, patients can securely share their records with the proposed approach without revealing their private keys. This approach also permits patients to get lab tests done from one hospital and follow-up treatment from another. With the usage of IPFS, the cost associated with storing data on Blockchain is also reduced by only putting hash values in the blockchain network.

Algorithm 3 revokePermission ( )

Input: Γ_D
Output: revoke permission

1: **Require:** Require: Only patient can revoke permission
2: check if already access, if yes i.e $canCreate[\Gamma_D] == true$
3: then $canCreate[\Gamma_D] = false$
4: from patients $doctor\_access\_list \leftarrow$ remove Γ_D
5: from doctors $patient\_access\_list \leftarrow$ remove Γ_P
6: **Emit:** inform the doctor about revoke permission

Algorithm 4 requestRecords ( )

Input: Γ_V, usage

1: **Require:** only viewers can call this function
2: **Emit:** inform the patient about viewer request for records and the purpose of usage

Algorithm 5 Approve&sendHashes ( )

Input: Γ_V, Γ_P, usage
Output: return hash value to requester

1: **Require:** only patient can call this function
2: If $msg.sender == \Gamma_V \&\& (is\_registered)$
3: Then Encrypt EHRsforVi.e.EncryptV(EHR)
4: upload to IPFS, get hash value
5: return hashes
6: **Emit:** return hashes to the viewer

4 Results and discussion

This section demonstrates the effectiveness of the proposed approach by presenting the results of the designed model after the execution of the algorithms. The cost analysis of the algorithms/functions considering the transaction and execution cost consumption, the security analysis, and the performance analysis as well as comparison with related works are also presented. In addition, the proper functioning of the system is shown using appropriate test cases.

4.1 Experimental evaluation and analysis

The proposed approach is experimentally evaluated to test and assess its performance. The system configuration used

for implementation and evaluation was a laptop running Ubuntu OS and an Intel(R) Core(TM) i5 processor running at 2.50GHz. The Smart contract (SC) was implemented in the Remix IDE [35] using the Solidity programming language [36] and was tested within the private network using JavaScript EVM. This helps in testing and optimizing the model before running it on the Ethereum mainnet.

a. **Functionality Evaluation:** Regulatory authority registers users with different roles, such as patients, doctors, and viewers, each with a unique Ethereum address. After registration, the process of storing and sharing the EHRs begins. The various test cases are presented below in Figs. 4, 5, 6, 7, 8 and 9 to show how the system responds to unauthorized and authorized users. Test Case 1: Attempting to register users with the same Ethereum address i.e. (0xAb8483F64d9C6-d1EcF9b849Ae677dD331583-5cb2) Test Case 2: Event log details of Doctor Account request to create and upload records Test Case 3: Non Doctor account request to create and upload records Test Case 4: Request granted to doctor by patient account and non-patient account Test Case 5: Revoke access to the doctor by patient account only Test Case 6: Request records by the viewer, grant/revoke access by the patient.
b. **Cost Analysis:** In the Ethereum platform, there is a transaction fee or cost associated with the execution of every instruction. We test the SC using the Javascript EVM environment on Remix IDE to analyze our SC. The transaction cost and execution cost in terms of gas consumption corresponding to different functions are shown below in Table 4.

The cost consumption of various functions/algorithms is shown in Table 4. There are two kinds of costs associated with blockchain networks. The first is transaction cost, while the second is execution cost. The quantity of gas needed to transmit information to the network is called transaction cost, while the quantity needed to execute the function is called execution cost. The transaction cost is higher than that of the execution cost because it comprises of execution cost plus the cost for transmitting information to the network.

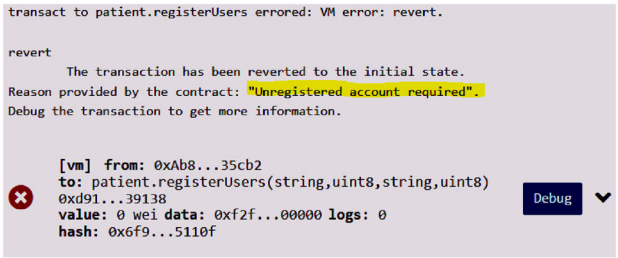

Fig. 4 Attempting to register users with the same Ethereum address

Fig. 5 Event log detail of Doctor Account request to create and upload records

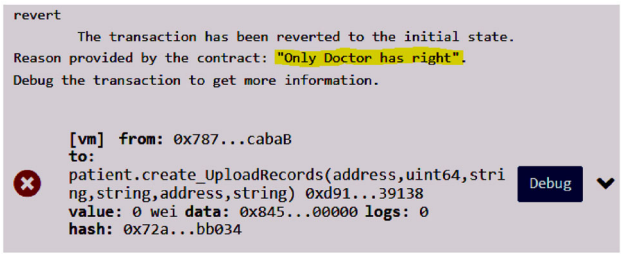

Fig. 6 Non Doctor account request to create and upload records

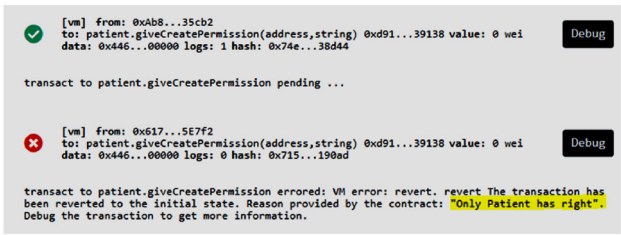

Fig. 7 Request granted to the doctor through patient account and non-patient account

Fig. 8 Revoke access to the doctor by patient account only

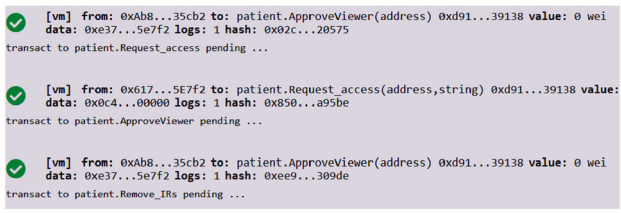

Fig. 9 Request records by the viewer, grant/revoke access by the patient

Table 4 Gas consumption analysis of different functions

Function()	Transaction cost (gas)	Execution cost (gas)
F1: Deploy Contract()	2867007	2150707
F2: Request_createRecords()	28104	5424
F3: grantPermission()	119241	95793
F4: create &UploadRecords()	138562	111018
F5: revoke_Permission()	46328	34472
F6: requestRecords()	27081	3505
F7: Approve &sendHashes()	33405	10725

Different functions consume different amounts of gas. Deploying contracts and calling constructors to consume a large portion of the transaction and execution costs. It is observed in Table 4 that the functions such as F2, F5, F6, and F7 have lesser costs as they involve only initialization and assignment operation. But the functions such as F3 and F4 have a higher cost due to the involvement of expensive checks, loops, and value insertion-deletion using arrays.

4.2 Security analysis

i. **Smart Contract (SC) Security Analysis:** SC is defined as the piece of code /protocol that is executed automatically in accordance with the agreement's condition. SCs are not mature enough and sometimes are prone to security attacks. Therefore, security analysis of smart contracts is necessary before deploying them in a practical scenario. The security analysis helps in identifying various vulnerabilities such as re-entrance vulnerability, timestamp dependency, and call stack depth attack vulnerability [37] in the smart contract. Thus, Oyente [38], an open-source tool for security analysis of the designed smart contract is used in our proposed work. Figure 10 demonstrates that the SCs are well designed and not suspectable to the flaws mentioned above, as the corresponding result is false.

ii. **Security analysis of the proposed system:** The proposed system uses CP-ABE with Blockchain technology to provide attribute-based access control for secure storage and sharing of EHRs. The cryptographic security of the scheme CP-ABE employed in this paper has been demonstrated in the literature [33]. This section shows that the system is resistant to various possible attacks and achieves security features.

```
}root@fa658129b16b:/oyente/oyente# python oyente.py -s PA_.sol
WARNING:root:You are using evm version 1.8.2. The supported version is 1.7.3
WARNING:root:You are using solc version 0.4.21, The latest supported version is
0.4.19
incomplete push instruction at 5910
INFO:root:contract PA_.sol:patient:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                  50.5%
INFO:symExec:     Integer Underflow:                  False
INFO:symExec:     Integer Overflow:                   False
INFO:symExec:     Parity Multisig Bug 2:              False
INFO:symExec:     Callstack Depth Attack Vulnerability:  False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:               False
INFO:symExec:     Re-Entrancy Vulnerability:          False
INFO:symExec:   ====== Analysis Completed ======
```

Fig. 10 Smart contract security analysis

- **Resistance to Masquerade Attack:** An attacker uses a masquerade attack to gain unauthorized access to a system, usually by obtaining credentials and passwords and by discovering program vulnerabilities. The proposed system resists masquerade attacks since our design approach is based on privileges. Each entity must register with the *RegisterUsers()* function before connecting to the network by providing the required data. The system then verifies them. As a result, each entity has its own unique Ethereum address Γ_E, which consists of 40 hexadecimal characters. Moreover, each entity has its unique ID (PID, DID and VID), making it impossible for an attacker to impersonate any registered entities. In addition, the EHRs are encrypted using the *Encryption(MPK,EHR,AP)* function, resulting in a ciphertext *ct* such that users with a set of attributes satisfying policy can decrypt the EHR. In addition, the hashes of the EHRs are placed in blockchain ledger so that an attacker cannot access them.
- **Resistant to Man-in-the-Middle attack:** In context of healthcare, a "man-in-the-middle attack" occurs when an attacker intercepts data being transferred between a patient and a doctor, including medical reports, personal information, and treatment plans, etc. With the goal of financial or medical identity fraud or other illegal conduct, the attackers can further alter this data. As the data is transferred directly between the owner, the patient TP, and the data requester, i.e., the physician Γ_D, this type of attack is not feasible in our system. An attacker, TA, somehow gains knowledge of the data hash, i.e.,$H_{(sha256)}(EHRs)$, and attempts to access the data stored in the IPFS but cannot decrypt it.

This is because the attacker must comply with the policy AP to decrypt the data. The decryption key of an authorized user can only decrypt data whose attributes match the policy AP. Consequently, the sensitive data is not disclosed during the sharing process, preventing the system from a man-in-the-middle attack.

- **Collusion Resistance:** A collusion attack occurs when multiple users unite to decrypt the ciphertext. Consider the case where user A of organization A having attributes attr1, attr2, collaborates with another user B of organization B of different location having attributes attr3, attr4, to decrypt the confidential data by combining their attributes. The proposed system is resistant to such attacks as private keys of each user are randomly generated. The *KeyGen()* uses the randomly generated value r along with other attributes of each user to generate private keys. As it is impossible for the users who unites to determine the same random number r, for decryption of data using private keys, thus fail to decrypt the data.
- **Data Integrity:** The proposed scheme CP-ABE with blockchain ensures data integrity. The data encrypted with an encryption function, *Encryption()* is stored in IPFS, and only the hashes are placed in the blockchain. The data in the chain ensures data integrity because of the cryptographic structure of the blockchain. Any slight change in the data will cause an avalanche effect, and these changes will be reflected in the entire chain unless there is a risk of a 51% attack. Additionally, users can check the changes in the data at any time. Also, the data stored at IPFS is decrypted by authorized users only satisfying the policy.
- **Non-Repudiation:** All the operations in the proposed system are recorded in the blockchain ledger as transactions. The transactions contain the unique Ethereum address and the unique ID of the user performing the action, such as creating and uploading records, accessing records, etc. Thus, the proposed approach exhibits the property of non-repudiation, where the user cannot repudiate a message or request they have made at any point in the future.

4.3 Performance measure of various cryptographic algorithms

- **Execution Time w.r.t. to varying number of attributes:** *KeyGen(), Encryption(), and Decryption()* depend upon the count of attributes used. Figure 11 shows the time taken (in milliseconds) by various algorithms w.r.t varying the attributes count.

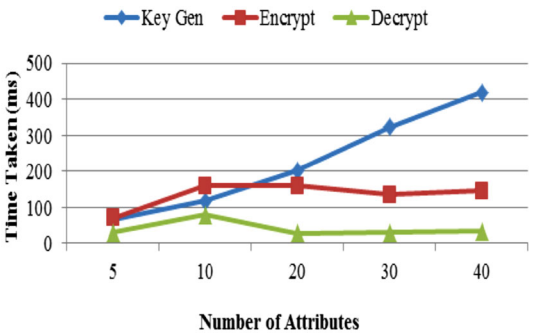

Fig. 11 Time taken by KeyGen, Encrypt and Decrypt algorithm w.r.t. number of attributes

It is noticed that Key Generation time grows linearly with an increase in the attributes count, which is due to the random values used in *KeyGen()* for generating private keys in exponential. In addition, encryption time increases from 5 to 10 ms by increasing the attributes count used to specify the access policy. On the other hand, decryption time does not follow the same pattern. This is because decryption is also influenced by the access policy and the quantity of characteristics. The more comparisons there are, the longer it takes to decode the data.

- **Comparison of encryption and decryption time:** The performance comparison of existing [21, 27] and proposed work is represented in terms of encryption and decryption time, respectively. The encryption and decryption time is estimated w.r.t varying file sizes to evaluate the performance of the proposed approach. The file size varies from 10 to 50 MB. From the Fig. 12, it is seen that for smaller file sizes (10–20 MB), the existing related works [21, 27], and the proposed work performs similarly, with a slight difference in time taken to encrypt EHRs. However, as the file size increases, the proposed method becomes faster and takes less time to encrypt than the existing related works. For example, for a file of 40 MB, the proposed

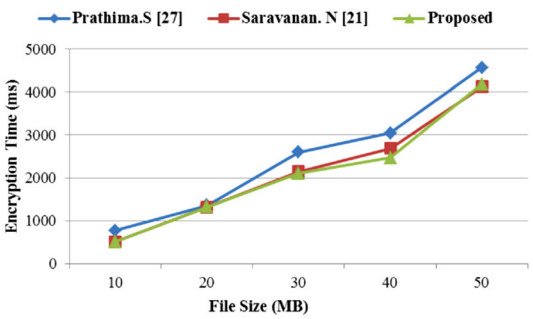

Fig. 12 File size vs encryption time

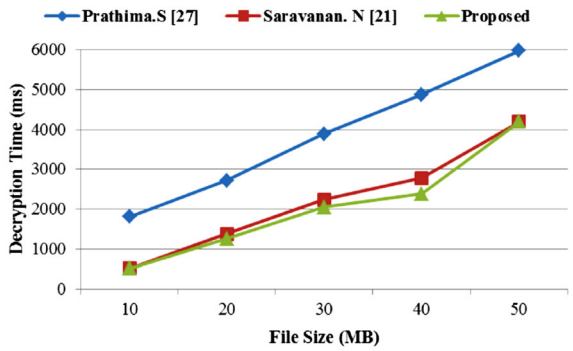

Fig. 13 File size vs decryption size

approach consumes 2470.23 ms for encryption compared to 3047 ms and 2689 ms for related works. Figure 13, shows the decryption time for the proposed and existing work. It is clearly observed that for smaller file sizes (10–20 MB) [21], takes lesser time and faster in comparison of proposed and related work [27]. However, as the file size increases, the proposed method becomes faster as it takes less time for decryption than the related work [21, 27].

5 Conclusion

This paper proposes an Ethereum blockchain-based EHR storage and sharing approach for the healthcare sector. An IPFS is utilized as an off-chain repository for encrypting records associated with patients, preventing them from malicious attacks or unauthorized access and resolving on-chain data storage and scalability issues by storing only hash values on the blockchain. The CP-ABE algorithm is utilized for encryption and decryption such that only intended user that satisfies the defined policy can decrypt the requested records. The combination of CP-ABE with blockchain technology provides a tamper-proof and verifiable audit trail of all data access and updations made to EHRs. This enhances accountability, non-repudiation and ensures that the patients or owners can track and verify all actions taken on the data. The proposed system is implemented using Remix IDE online open-source tool that enables the development and deployment of smart contracts for the Ethereum blockchain.

In this work, Smart contracts are designed such that patients have full authority over their records and have the privilege of deciding with whom they want to share their records. All the transactions are recorded on the blockchain. Different test cases are presented to show the proposed algorithm's functioning and cost analysis. Finally, the security analysis of the proposed system is carried out by evaluating its resistance to various attacks. Additionally, potential security flaws in the proposed SCs are investigated using the Oyente tool. The outcomes demonstrate that the proposed decentralized system is efficient for securely storing also sharing EHRs with authorized users without exposing private keys. Thus, the proposed system addresses data integrity, security, scalability, and immutability challenges. The proposed approach can also be utilized in other applications, such as incentive-based sharing of medical images to researchers instead of the traditional image management system.

In future, the research goal is to handle queries instantly by improving response duration and reducing latency and restricting cost consumption so that they can be used with devices with limited resource capacity.

Author contributions The idea of blockchain implementation for healthcare: JK, RR and NK. Design of proposed architecture: JK, RR and NK. Implementation of proposed architecture: JK. performance analysis: RR, NK and JK, writing of manuscript: JK. All the authors read, edited and approved the final manuscript.

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Competing interests The authors have not disclosed any competing interests.

Informed consent Not applicable.

References

1. Henry, J., Pylypchuk, Y., Searcy, T., Patel, V.: Adoption of electronic health record systems among U.S. non-federal acute care hospitals: 2008–2015. ONC Data Brief **35**, 1–9 (2016)
2. National trends in hospital and physician adoption of electronic health records: (2021). https://www.healthit.gov/data/quickstats/national-trends-hospital-and-physician-adoption-electronic-health-records
3. Saha, A., Amin, R., Kunal, S., Vollala, S., Dwivedi, S.K.: Review on blockchain technology based medical healthcare system with privacy issues. Secur. Priv. **2**(5), e83 (2019). https://doi.org/10.1002/spy2.83
4. Healthcare-data-breach-report. HIPAA Journal. (2020). https://www.hipaajournal.com/july-2020-healthcare-data-breach-report/
5. Kaur, J., Rani, R., Kalra, N.: A blockchain-based framework for privacy preservation of electronic health records (EHRS). Trans. Emerg. Telecommun. Technol. (2022). https://doi.org/10.1002/ett.4507
6. Xhafa, F., Feng, J., Zhang, Y., Chen, X., Li, J.: Privacy-aware attribute-based PHR sharing with user accountability in cloud computing. J. Supercomput. **71**(5), 1607–1619 (2015). https://doi.org/10.1007/s11227-014-1253-3
7. Rodrigues, J.J., de la Torre, I., Fernández, G., López-Coronado, M., et al.: Analysis of the security and privacy requirements of

cloud-based electronic health records systems. J. Med. Internet Res. **15**(8), e2494 (2013). https://doi.org/10.2196/jmir.2494
8. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) Advances in Cryptology-EUROCRYPT 2005. EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 457–473. Springer, Berlin (2005). https://doi.org/10.1007/11426639_27
9. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security, Alexandria Virginia, pp. 89–98. (2006). https://doi.org/10.1145/1180405.1180418
10. Mubarakali, A.: Healthcare services monitoring in cloud using secure and robust healthcare-based blockchain (SRHB) approach. Mobile Netw. Appl. **25**(4), 1330–1337 (2020). https://doi.org/10.1007/s11036-020-01551-1
11. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **24**(1), 131–143 (2013). https://doi.org/10.1109/TPDS.2012.97
12. Hong, H., Liu, X., Sun, Z.: A fine-grained attribute based data retrieval with proxy re-encryption scheme for data outsourcing systems. Mob. Netw. Appl. (2018). https://doi.org/10.1007/s11036-018-1102-3
13. Makani, S., Pittala, R., Alsayed, E., Aloqaily, M., Jararweh, Y.: A survey of blockchain applications in sustainable and smart cities. Clust. Comput. (2022). https://doi.org/10.1007/s10586-022-03625-z
14. Buterin, V.: A next-generation smart contract and decentralized application platform. https://ethereum.org/en/whitepaper/ (2014). Accessed 2 Aug 2021
15. She, W., et al.: New blockchain technology for medical big data security sharing. J. Chin. Comput. Syst **40**(7), 1449–1454 (2019). (http://xwxt.sict.ac.cn/EN/abstract/abstract5022.shtml)
16. Simplyvital health: https://www.f6s.com/simply vitalhealth (2020). Accessed 15 Sept 2021
17. Koepsell, D.: The future of genomic data encryption. https://encrypgen.com/ (2020). Accessed 15 Sept 2021
18. Vora, J. et al.: Bheem: A blockchain-based framework for securing electronic health records. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps) 1–6. (2018). https://doi.org/10.1109/GLOCOMW.2018.8644088
19. Shahnaz, A., Qamar, U., Khalid, A.: Using blockchain for electronic health records. IEEE Access **7**, 147782–147795 (2019). https://doi.org/10.1109/ACCESS.2019.2946373
20. Thwin, T.T., Vasupongayya, S.: Blockchain-based access control model to preserve privacy for personal health record systems. Secur. Commun. Netw. (2019). https://doi.org/10.1155/2019/8315614
21. Saravanan, N., Umamakeswari, A.: HAP-CP-ABE based encryption technique with hashed access policy based authentication scheme for privacy preserving of phr. Microprocess. Microsyst. **80**, 103540 (2021). https://doi.org/10.1016/j.micpro.2020.103540
22. Ali, A., et al.: A novel secure blockchain framework for accessing electronic health records using multiple certificate authority. Appl. Sci. **11**(21), 9999 (2021). https://doi.org/10.3390/app11219999
23. Li, F., Liu, K., Zhang, L., Huang, S., Wu, Q.: Ehrchain: a blockchain-based EHR system using attribute-based and homomorphic cryptosystem. IEEE Trans. Serv. Comput. **15**(5), 2755–2765 (2022). https://doi.org/10.1109/TSC.2021.3078119
24. Sharma, P., Jindal, R., Borah, M.D.: Blockchain-based cloud storage system with CP-ABE-based access control and revocation process. J. Supercomput. (2022). https://doi.org/10.1007/s11227-021-04179-4
25. Ali, A., et al.: An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. Sensors **22**(2), 572 (2022). https://doi.org/10.3390/s22020572
26. Ali, A., et al.: Security, privacy, and reliability in digital healthcare systems using blockchain. Electronics **10**(16), 20–34 (2021). https://doi.org/10.3390/electronics10162034
27. Prathima, S., Priya, C.: Improved CP-ABE based crypto technique to secure EHRS with access policy-based authentication schemes. J. Pharm. Negat. Results **13**, 2365–2379 (2022)
28. Almaiah, M.A., Hajjej, F., Ali, A., Pasha, M.F., Almomani, O.: A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS. Sensors **22**(4), 1448 (2022). https://doi.org/10.3390/s22041448
29. Almaiah, M.A., Ali, A., Hajjej, F., Pasha, M.F., Alohali, M.A.: A lightweight hybrid deep learning privacy-preserving model for FC-based industrial internet of medical things. Sensors **22**(6), 2112 (2022). https://doi.org/10.3390/s22062112
30. Buterin, V.: What is ethereum? Ethereum official webpage. http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html (2020). Accessed 2 Aug 2021
31. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. https://git.dhimmel.com/bitcoin-whitepaper/ (2020). Accessed 15 Sept 2020
32. Zheng, Q., Li, Y., Chen, P., Dong, X.: An innovative IPFS-based storage model for blockchain. In: Proceedings of 2018 IEEE/WIC/ACM international conference on web intelligence (WI) pp. 704–708 (2018). Santiago, Chile. https://doi.org/10.1109/WI.2018.000-8
33. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of 2007 IEEE symposium on security and privacy (SP '07), pp. 321–334, (2007). Berkeley, CA, USA. https://doi.org/10.1109/SP.2007.11
34. Akinyele, J.A., et al.: Charm: a framework for rapidly prototyping cryptosystems. J. Cryptogr. Eng. **3**(2), 111–128 (2013). https://doi.org/10.1007/s13389-013-0057-3
35. Remix ide: https://remix-project.org/ Accessed 2 Aug 2021
36. Solidity. https://docs.soliditylang.org/en/v0.7.4/. Accessed 15 Oct 2020
37. Dika, A., Nowostawski, M.: Security vulnerabilities in ethereum smart contracts. 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. 955–962 (2018). Halifax, NS, Canada. https://doi.org/10.1109/Cybermatics_2018.2018.00182
38. Luu, L., Chu, D.-H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp. 254–269 (2016). Vienna, Austria. https://doi.org/10.1145/2976749.2978309

Jasleen Kaur is pursuing PhD from Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India. She has done her Master of Engineering (CSE) from TIET (Deemed to be University), Patiala, Punjab, India, in 2018, and B.Tech (CSE) from BBSBEC, PTU, (Pb.) India, in 2016. She has cleared the GATE exam and is also UGC-Net qualified. Her areas of interest are Blockchain Technology, cryptography, Database &information security and Machine Learning.

Dr. Rinkle Rani is working as Associate Professor in Computer Science and Engineering Department, Thapar University, Patiala. She has done her Post graduation from BITS, Pilani and Ph.D. from Punjabi University, Patiala. She has more than 26 years of teaching experience. She has over 172 research papers, out of which 90 are in international journals and 80 are in conferences. She has supervised 11 PhD and 47 ME theses. Her areas of interest are Big Data Analytics and Machine Learning. She is member of professional bodies: ACM, IEEE and CSI. She may be contacted at raggarwal@ thapar.edu

Dr. Nidhi Kalra is working as Assistant Professor in Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala. She has done her PhD in Computer Science and Engineering Department from Thapar Institute of Engineering & Technology, Patiala in the year 2017, Master of Engineering from PEC University of Technology, Chandigarh, India in the year of 2014 and B.Tech from AIET, PTU, (Pb.) in the year 2012 India. She has published numerous papers in SCI indexed Journals and Scopus Indexed Conferences. Her main research interests are Theoretical Computer Science and Blockchain Technologies. She is the member of professional bodies: IEEE and ACM. She can be contacted at nidhi.kalra@thapar.edu, kalranidhi8@gmail.com.