Blockchain-Driven Decentralization of Electronic Health Records in Saudi Arabia: An Ethereum-Based Framework for Enhanced Security and Patient Control

Atef Masmoudi[1], Maha Saeed[2]
Laboratory of Electronics and Technology of Information National Engineering School of Sfax,
University of Sfax, Sfax, Tunisia[1]
College of Computer Science, King Khalid University, Abha, Saudi Arabia[1,2]

***Abstract*—In the rapidly evolving landscape of e-HealthCare in Saudi Arabia, enhancing the security and integrity of Electronic Health Records (EHRs) is imperative. Existing systems encounter challenges stemming from centralized storage, vulnerable data integrity, susceptibility to power failures, and issues of ownership by entities other than the patients themselves. Moreover, the sharing of sensitive patient information among anonymous bodies exacerbates the vulnerability of these records. In response to these challenges, this paper advocates for the transformative potential of blockchain technology. Blockchain, with its decentralized and distributed architecture, offers a revolutionary approach to communication among network nodes, eliminating the need for a central authority. This paper proposes a solution that places the patient at the forefront, empowering them as the primary controller of their medical data. The research delves into the current state of e-HealthCare in Saudi Arabia, examines the challenges faced by existing EHR systems, and introduces blockchain technology, particularly Ethereum, as a viable and transformative solution. The paper details the use of Ethereum blockchain to secure and manage medical records, with a Public Key Infrastructure (PKI) applied to safeguard the confidentiality of patient information. The decentralized InterPlanetary File System (IPFS) is employed for the secure and resilient storage of encrypted medical records. Additionally, Smart contracts, integral to the Ethereum blockchain, play a central role in automating and enforcing the rules governing access to medical records. Moreover, a Web 3.0 decentralized application (DApp) is developed to provide a user-friendly interface, empowering patients to seamlessly interact with and control access to their health data. At the end, this paper presents a guiding framework for clinicians, policymakers, and academics, illustrating the transformative potential of blockchain and associated technologies in revolutionizing EHR management in Saudi Arabia's healthcare systems.**

***Keywords*—Blockchain; Ethereum; smart contract; Web 3.0; decentralized application; electronic health records**

I. Introduction

The progression of technology in the modern era has significantly altered human perspectives and lifestyles. Technological advancements extend their influence to various facets of life, including healthcare, with the primary goal of transforming and enhancing the sector. In the healthcare realm, technology offers substantial benefits, revolutionizing systems for improved ease of use, security, and overall efficiency. One such advancement that has played a pivotal role is the implementation of Electronic Medical Record (EMR) systems.

Before the advent of EMR systems, hospitals relied on handwritten documentation on paper, leading to data loss and patient challenges in updating records across different healthcare facilities. Recognizing the limitations of this approach, the healthcare sector embraced EMR systems, providing a transformative solution by storing data electronically and reducing reliance on paper. These electronic records encompass clinical notes and comprehensive laboratory results with multiple components [1]. This shift addressed prevailing issues in medical record management, enhancing information accessibility [2], error prevention, and the establishment of an adaptable system that could evolve over time [3]. Consequently, the adoption of EMR systems emphasizing enhanced usability, streamlined data retrieval, efficient updates, and overall improved data management processes, making them pivotal in modernizing healthcare practices.

However, despite the intended improvements in patient care, EMR systems encountered critical challenges, falling short of expectations [3]. Issues such as data security, integrity, and user ownership emerged as significant concerns. A study conducted in four academic hospitals in Saudi Arabia revealed that EMR systems faced reliability issues, particularly insecure access to records [3]. Furthermore, vulnerability to data breaches became evident, with a study reporting 173 million data breaches since October 2009 [4]. Compounding the challenges were issues of data duplication and the need for repetitive medical examinations when patients visited different hospitals [5]. The cumulative impact of these challenges emphasized the necessity for a secure, interoperable, and patient-centric healthcare data management system.

In response to these challenges, this paper advocates for the transformation of the healthcare sector into a patient-centered model using blockchain technology. Blockchain has the potential to revolutionize data control and management within existing EMR systems. The paper focuses on the Saudi Arabian context, proposing the utilization of the blockchain platform, IPFS (InterPlanetary File System), and smart contracts to address the prevalent issues in medical records. The proposed solution aims to empower patients by enabling them to control, protect, and share their medical data securely.

Smart contracts, defining the roles of participating entities, have been developed and tested to govern transactions and monitor procedures carried out on the EMRs. This research seeks to demonstrate the efficacy of blockchain in mitigating challenges associated with medical records in Saudi Arabia, ultimately fostering a more secure, patient-centric healthcare data management paradigm.

The subsequent sections of the paper are structured as follows: Section II provides a background review, offering a concise explanation of blockchain technology. In Section III, previous studies relevant to our research project are discussed. Section IV outlines the proposed solution, presenting the methodology employed. Section V delves into the smart contract functions, detailing the specific functionalities implemented in the Ethereum-based framework. Following this, Section VI discusses the findings and implications in the context of the proposed solution. Finally, Section VII encompasses the conclusion.

II. Background

This section serves as a foundational guide, introducing key elements of the proposed research. We delve into blockchain technology, its decentralized architecture, and the pivotal role played by Peer-to-Peer (P2P) networks. Types of blockchains, including a focus on Ethereum blockchain, are explored, shedding light on their unique characteristics. The Ethereum blockchain, with its support for smart contracts and decentralized applications (DApps), emerges as a prominent player in this space. The significance of smart contracts in executing predefined rules on the blockchain is emphasized, highlighting their transformative potential. Additionally, we delve into the essential role of consensus protocols in governing the validation of transactions and maintaining the integrity of the blockchain. The section also explores the transformative IPFS, acknowledging its importance in reducing storage costs and providing an extra layer of privacy through data encryption. This comprehensive background lays the groundwork for understanding the technological choices and components integral to our research.

A. Blockchain Technology

Nakamoto is the name of the person who invented the blockchain technology [6]. His idea was based on creating a decentralized and encrypted currency to be used in financial transactions. Eventually, this technology was used in many different areas of life [7], such as the healthcare sector. A lot of research was conducted to study of applying blockchain technology to the current healthcare systems, and this research has reached to identify the pros and cons related to the use of this technology.

The benefit of using a blockchain is to facilitate the exchange of transactions between two parties without the need for an intermediary [6]. The process begins with creating a block whose hash is linked to the hash of previous block, and this method is repeated until the chain is complete. Each block in the blockchain includes records of transactions which will be verified by so-called miners. The contents of the block include previous block hash, timestamp, data, and nonce, collectively forming a secure and transparent ledger [8], [9]. The transaction could be data of any data type and the use of nonce is to control of the hash that is generated from the block. All these components will be hashed together to represent a block.

1) P2P network: A peer, which is a node in the blockchain network, contributes resources including storage, processing power, and bandwidth. These nodes are either for specific people or is available to everyone on blockchain because there are multiple types of blockchain network (as will be explained later). The most important characteristic of the nodes in the blockchain network is that the identity of each node remains safe, due to the public key belonging to each user is visible only to other peers in the network. The nodes also act as miners to verify the transaction before adding it to the chain.

2) Role of miners: As mentioned earlier, the miner's job is to verify the block before adding it to the blockchain network [10]. This does not guarantee the eligibility of the transaction, but rather to perform other additional work after that, which is called Proof-of-Work (PoW). Nonce will create a hash value that is less than the target difficulty level, and this is done through the PoW process. Then it is solved in a short period of time when the miner approaches the value below the target to get the rewards.

B. Types of Blockchain

Currently, there are three types of blockchain, depending on features such as network size, application, and type of algorithms, as follows:

1) Public blockchain: The public blockchain network is available for access by anyone on the network, and after joining, he can access the blocks' data and become authorized to mine. The use of pseudo-anonymous hash value is to create unique address to identify the users even in the public type of blockchain networks. Thus, people in the network are known by their addresses, not by their actual identity. And when someone wants to interact, such as adding or download a document such as EMRs, he must pay the costs of this interaction (transaction fees).

2) Private blockchain: Private and public blockchains are similar in many aspects such as process and algorithms, but the difference lies in the purpose. A private blockchain network is defined as a restrictive network, as it is designed in proportion to closed networks and based on the element of access control. Private blockchain networks are usually used for small businesses where the nodes are controlled to perform transactions and execute smart contracts. This type is used in many applications that require privacy and security, such as digital asset management, online voting, and supply chain management. In addition, the private blockchain networks require the approval from network administrators to allow people to join.

3) Consortium blockchain: Consortium blockchain networks combine the features of centralized and decentralized systems. Instead of being used to serve a single organization, the consortium blockchain is used on a large scale and in many organizations. This type is similar to private blockchain in that no one can access the network directly without registering, as that the approval of other organizations is required to perform any operation.

C. The Consensus Protocols

Blockchain, at its core, relies on consensus protocols [11] to ensure agreement among distributed participants on the state of the ledger. These protocols dictate how nodes or participants in a network reach a unanimous decision on the validity of transactions and the order in which they are added to the blockchain. The consensus mechanisms underpin the fundamental principles of trust, security, and decentralization in blockchain technology.

The choice of the consensus protocol is essential for designing blockchain systems tailored to specific use cases, whether in public, private, or consortium settings. It influences the network's characteristics, including security, efficiency, and governance, shaping the overall success and adoption of blockchain technology.

The most pivotal consensus mechanisms that have shaped the development of blockchain are:

1) PoW: PoW is a consensus algorithm widely employed in public blockchain networks like Ethereum. In this model, miners engage in solving complex mathematical puzzles, and the first to solve it gains the privilege to add a new block to the blockchain. While PoW ensures decentralization and robust security, its drawback lies in its energy-intensive nature, a concern that has prompted the exploration of alternative consensus mechanisms.

2) Proof of Stake (PoS): PoS is an alternative consensus algorithm where validators are chosen to create new blocks based on the amount of cryptocurrency they hold and are willing to "stake" as collateral. This approach, compared to PoW, offers a more energy-efficient solution. Participants are incentivized to act in the best interest of the network by risking their held cryptocurrency as collateral.

3) Proof of Authority (PoA): In the context of private blockchain networks, PoA emerges as a compelling consensus algorithm. Unlike PoW and PoS, PoA does not rely on competitive mining or staking. Instead, network participants, often identified and reputable entities, are designated as authorities. These authorities validate transactions, offering a more efficient and scalable solution for private consortium networks.

D. IPFS

The IPFS [12], [13], [14] stands as a revolutionary solution in healthcare data management, offering a distributed file system shared through a P2P network. In the context of storing medical records on the blockchain, IPFS plays a pivotal role in significantly reducing storage costs. Instead of storing voluminous medical data directly on the blockchain, IPFS allows for the storage of a unique hash that represents the content's identity. This hash serves as a pointer to the actual data stored in the IPFS network, optimizing the efficiency of the overall system.

Furthermore, the IPFS introduces an additional layer of privacy by enabling the encryption of data. This ensures that sensitive medical information remains confidential and secure throughout its lifecycle within the decentralized network. The ability to encrypt data in IPFS not only aligns with privacy regulations but also addresses the paramount importance of safeguarding patient information in healthcare settings.

In essence, by leveraging IPFS in conjunction with blockchain technology, the proposed framework not only enhances data security but also demonstrates a cost-effective approach to managing medical records. This integration allows for the creation of a robust and efficient healthcare data management system, aligning with the evolving needs of the healthcare sector in Saudi Arabia.

E. Ethereum Blockchain

Ethereum [15] stands as a pioneering blockchain platform renowned for its versatility in deploying DApps and smart contracts. Unlike Bitcoin, Ethereum extends beyond a mere cryptocurrency framework, providing a comprehensive environment for executing complex decentralized applications.

1) Smart Contract: The concept of smart contracts was introduced in 1997 by Nick Szabo [16], to conduct digital transactions securely over a network. Smart contracts [17], [18] are applications that are executed by people joining the blockchain network. Smart contracts are written by computer codes to implement pre-defined rules and conditions to keep transactions controlled. Current examples of projects based on the use of smart contracts [19] are the Ethereum platform and Hyperledger. These platforms allow transactions to be conducted reliably between anonymous parties without the need for a central authority. The Ethereum platform enables the creation of smart contracts tailored to the specific needs of any system within the network. The Ethereum uses Ether as a cryptocurrency. As for the term gas, it is used to measure the cost of any function that takes place in the smart contract. With regard to medical records, smart contracts allow the secure and reliable exchange of records.

2) Geth (Go Ethereum): Geth serves as the official Go implementation of the Ethereum protocol. As a command-line interface tool, Geth facilitates interaction with the Ethereum network, enabling users to run a local Ethereum node, mine Ether, and interact with smart contracts. Geth plays a pivotal role in supporting the infrastructure of the Ethereum blockchain.

3) DApps: Decentralized applications [20], [21], or DApps, represent a new paradigm in application development. Unlike traditional applications that rely on centralized servers, DApps leverage blockchain technology, ensuring decentralized, transparent, and secure operations. These applications run on a P2P network of computers, eliminating the need for intermediaries and fostering a trustless environment.

III. Literature Review

In Estonia, the entire public health infrastructure is being operated using blockchain [22]. Through the application of blockchain technology, the coasts of medical records will be reduced for the patient and other interested parties. Estonia is one of the first countries that uses blockchain technology in most government and commercial sector. The Estonian government has started making GovTech partnerships to apply blockchain to all industries in the region, so that the country become advanced in the field of technology.

Xia et al. [23], proposed a MeDShare system that addresses sharing a big data of medical records among a trust-less

environments. The system uses cloud services along with blockchain technology to store big data. The procedures start on MeDShare system by transferring data side by side from one entity to another and then recorded it in a tamper-proof manner. One of the advantages of the system is that when the permissions are violated, data access is canceled using a control mechanism designed by smart contracts. The system consists of four layers: First, user layer, which can access the data for research purposes. Second layer has a several of functions such as query, process, and respond to other queries on the system. Third layer, it process data requests in the infrastructure layer and performs computational operations on the data that was requested with the ability to monitor it, and this layer called data Structuring and Provenance layer. Fourth layer, existing databases that individual parties work on to accomplish certain tasks. This system guarantees data security by applying smart contracts. MeDShare system can be compared with the current systems that use cloud services. However, this system still depends on a third party.

Daraghmi et al. [24], suggested a MedChain system for managing EMRs. In this approach, the healthcare provider is responsible for creating, verifying, and adding new blocks to the blockchain, as well as allowing the patient to control their own data and granting or denying access to it. Smart contracts are written to control transaction times and monitor operations performed on medical records. Experiments conducted on MedChain system showed the efficiency of the proposal in response time, connection times and dealing with large data set. However, this system may encounter problems related to the fact that the database is central and pre-existing.

Dubovitskaya et al. [25], proposed a system to help cancer patients manage their medical records using the blockchain network. The role of the doctor and the patient is determined through membership service, through which the system verifies any doctor who joins the system, whether he is registered in the National Practitioner Data Bank or not. The patient's medical record and encryption keys are also signed with the membership service to ensure confidentiality. Th data is then stored in the hospital database and in the cloud to grant access to other individuals in the network. A patient key is used to encrypt their data before it is stored in the cloud. The user is provided with an application programming interface (API) that transmits actions from the user to the nodes that are organized by the leader node to initiate the consensus protocol. However, as a result of storing data locally in hospitals and using cloud services, the patients cannot fully control their data, and this is one of the limitations that must be considered in this system.

Abid et al. [26], proposed NovidChain system to issue and verify the COVID-19 test/vaccine certificate using a private Blockchain network. They used a number of emerging technologies such as self-sovereign Identity platform called uPort. uPort is a mobile application that serves as an authentication mechanism for decentralized applications. They used IPFS to store data off-chain after it's encryption. The main process of NovidChain begins with Healthcare provider registration and service provider by creating an account on a self-managed Blockchain wallet then send the public key to the account belong to the healthcare authority. The healthcare authority then check if the Healthcare provider authorized or not, then adds his account to the blockchain. Next the user must create a Blockchain account and install a self-managed Blockchain wallet for the healthcare provider to verify the user's official ID. After that, the user becomes eligible to get the COVID-19 vaccine from a Healthcare provider. The data is then encrypted, and its hash is stored in the blockchain. Finally, official physical ID is presented with a QR code to verify the health status of the individual in the registration step. They also explained that the NovidChain system ensures self-sovereign identity (SSI), as they adopted decentralized IDs (DIDs) for countersigning, signing and verifying COVID-19 credentials.

Sun et al. [27], proposed a system for sharing and storing medical records using smart contract technology. The system begins with encrypting the medical record by the doctor, adjusting the appropriate access settings, and then storing the encrypted record on IPFS. They demonstrate the benefit of using IPFS with blockchain technology in enabling healthcare providers to process and store a greater amount of medical data on IPFS rather than on the blockchain itself for savings in network bandwidth. After the medical records are encrypted and stored in the IPFS, they create an index of keywords to use in searching the encrypted records. Index words are stored in the Ethereum blockchain, where they can be accessed after the smart contract is published which in turn defines the way to access it in the distributed system.

Azaria et al. [28] proposed MedRec that enables patients to check a log of their health records. They used Ethereum's smart contracts to represent the data stored into individual nodes on the network. They used metadata about the medical record ownership, data integrity, and permissions by writing contracts. To deal with these properties, blockchain transactions carry signed and encrypted instructions for dealing with big data, and then the system inserts them into the blockchain network using three main types of contracts: First, Registrar Contract (RC), to link the Ethereum address identity with the corresponding identification strings, then change existing identities and organize the registration of new ones and append identity strings to the blockchain. Second, Patient-Provider Relationship Contract (PPR): which symbolizes any pairwise data stewardship interaction. Third, Summary Contract (SC): to determine medical record history by identifying the previous and current links of the participant with other nodes in the system.

Marcela et al. [29] proposed a hybrid system for protecting patient data privacy that involves combining blockchain technology with public key infrastructure. They store medical data on the blockchain using secret session keys. Initially, the digital certificate defines the main roles in the system. After the patient data and the session key are stored in the blockchain, these data are sent to the doctor, to ensure that the patient is the primary controller of access to his data. The second role is the role of the doctor, who can send and receive the patient's session key, but doctors are not authorized to share the session key among themselves. There is a pair of public and private keys that must be present for every user on the system so that every transaction made on the blockchain is assigned to the actor by the public and private keys of both the doctor and the patient shown in their certifications. To keep the security kernel of their proposal, they design their approach as a private permissioned network. In order to ensure the consistency and availability of transactions, the system converts the consensus

protocol to a three-phase commit protocol. The result showed that the transaction overload is appropriate compared to the number of transactions on the blockchain.

Sammeta et al. [30] developed a new model for transferring patient's data as well as making a diagnosis, called HBESDM-DLD model. The idea of this model is to take advantage of the Hyperledger blockchain-based in managing many orations such as encryption, data management, diagnosis, and optimal key generation. Firstly, the GTOA-based optimal key generation technique plus SIMON technology are used to encrypt the patient's medical data. Whether or not the patient is allowed to access to medical institutions is determined by the specific policy imposed by the global and local blockchain in partnership with the Hyperledger blockchain.

Azbeg et al. [31] presented a system named BlockMedCare which uses a set of technologies in addition to Blockchain, such as IoT and IPFS to manage health systems related to chronic diseases. The approach consists of three main sections patient side, medical team side and IPFS side. Initially, the patient's IoT device is registered using the owner's identity and MAC address on the blockchain network to identify the patient, and then the data collected from the patient's device is shared with the specialized medical team through the patient's smartphone. Hospitals also store a copy of the blockchain in addition to their ability to participate in the consensus process. The system enables the medical team to use patient data for procedures other than monitoring, including analysis and research. IPFS is used to store data as IoT devices collect a huge amount of data that cannot be stored only using blockchain. The system is used Clique PoA as a consensus algorithm to reach agreement and using the proxy re-encryption mechanism in addition to blockchain features to maintain the security of the system.

Alternative methods for leveraging blockchain in the management of medical records are detailed in [32], [33], [34].

IV. Proposed Solution

In response to the evolving landscape of healthcare digitization in Saudi Arabia, this research presents a holistic solution aimed at revolutionizing the management of EHRs. The proposed system integrates a private Ethereum blockchain, a private IPFS, and a DApp developed using React with web3.js integration. The system architecture adopts a PoA consensus mechanism, utilizing the Geth client for Ethereum, and JSON-RPC for communication. This multifaceted approach aims to address the existing challenges in terms of security, scalability, and user accessibility within the Saudi Arabian healthcare ecosystem.

Fig. 1 illustrates the blockchain-based framework designed to enhance security and empowering patients with control over their medical records.

A. Blockchain Infrastructure

This section introduces the foundational elements that shape the proposed blockchain-based healthcare management system. Central to this infrastructure is the adoption of a private Ethereum blockchain, leveraging a PoA consensus mechanism for enhanced efficiency and security. This choice aligns with

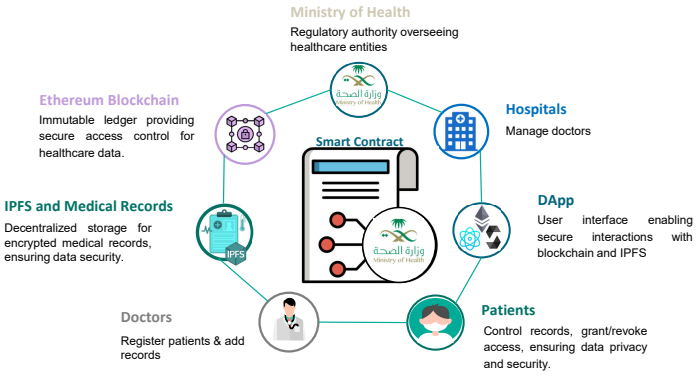

Fig. 1. The proposed blockchain-based framework to enhance security and afford patients control over their medical records.

the sensitive nature of healthcare data, emphasizing privacy and trust. The node structure mirrors the Saudi Arabian healthcare system, strategically designating healthcare authorities, hospitals, and clinics as nodes, ensuring compatibility with the existing infrastructure. Smart contracts form the core of the system, defining hierarchical rules for entity registration, with the Ministry of Health holding registration authority over Local Health Authorities, which, in turn, register entities such as hospitals and clinics. This controlled registration mechanism, cascading from higher to lower authorities, mitigates security risks. The DApp, built on the React framework, acts as the user interface, seamlessly integrating with the Ethereum blockchain through the web3.js library. This DApp serves as a user-friendly gateway for healthcare authorities, doctors, and patients to manage registrations, access medical records, and interact with smart contracts. The subsequent discussions in the section delve into the encryption and IPFS integration processes, ensuring the secure issuance and storage of patients' medical records.

The different elements that shape the proposed blockchain-based healthcare management system are:

- Private Ethereum network: The proposed system adopts a private Ethereum blockchain to ensure data security and privacy. Leveraging a PoA consensus mechanism enhances the overall efficiency of the network. Unlike public blockchains, PoA networks consist of a known set of nodes, each with a proven identity, reducing the risk of malicious activities. This choice aligns with the sensitivity of healthcare data, emphasizing confidentiality and trust among participating entities.
- Node structure: Nodes within the private Ethereum network are strategically structured to mirror the Saudi Arabian healthcare system. Healthcare authorities, hospitals and clinics are designated as nodes. This design not only aligns with the existing healthcare infrastructure but also facilitates scalability and streamlined interactions.
- Smart contracts: At the core of the proposed system are smart contracts that define the rules and interactions within the healthcare network. These contracts account for the hierarchical structure of healthcare

authorities, wherein only higher authorities possess the privilege to register the different entities. The Ministry of Health holds the authority to register Local Health Authorities, which, in turn, have the capability to register various healthcare entities such as hospitals and clinics. Within this framework, hospitals and clinics are empowered to register doctors, and once registered, doctors assume the responsibility of registering patients. This registration mechanism ensures a controlled and secure onboarding process, mitigating risks associated with unauthorized access and denial-of-service (DOS) attacks.

- DApp: The user interface of the proposed system is developed as a decentralized application using the React framework. This DApp integrates with the Ethereum blockchain through the web3.js library, providing a user-friendly experience for healthcare authorities, doctors, and patients. The DApp serves as the gateway for managing registrations, accessing medical records, and interacting with smart contracts seamlessly.
- Encryption and IPFS integration: The is the pivotal aspects of ensuring the confidentiality and integrity of medical records within the proposed system. To achieve this, a robust encryption mechanism is employed, safeguarding the sensitive patient information contained in medical records. The encryption process involves the generation of a unique symmetric key for each medical record, ensuring that access is restricted to authorized entities only. These encrypted medical records are then securely stored on the IPFS. The integration with IPFS not only enhances data availability and reliability but also significantly reduces storage costs in the blockchain. This combination of encryption and IPFS integration establishes a secure and efficient framework for handling and storing medical records, ensuring the utmost privacy and integrity of patient information.

B. The Patient Registration Flow

To enable a doctor to register a patient through the DApp, including the creation and communication of account details to the patient, and the addition of the patient to the list of patients in the blockchain using the corresponding smart contract, a multi-step process is implemented. Below are the detailed steps:

The detailed patient registration flow, as presented in Fig. 2, is as follows:

1) Doctor initiates registration: The doctor, using the DApp, initiates the registration process for a specific patient.
2) Patient details entry by doctor: The doctor enters the necessary patient details into the DApp. This could include the patient's national ID, name, date of birth, and any other required information.
3) Generate Ethereum account: The DApp generates a new Ethereum account for the patient, including the Ethereum address and private key.
4) Register the patient: The DApp interacts with the smart contract on the private Ethereum blockchain.

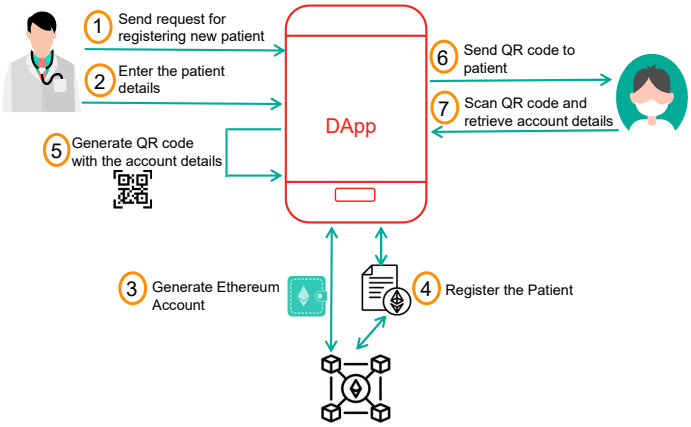

Fig. 2. The patient registration flow.

The smart contract includes a function to add a new patient to the list of patients, recording their their relevant details.
5) QR code generation: The DApp generates a QR code containing the Ethereum account details along with the patient's information.
6) DApp provides QR code to patient: The DApp communicates the generated QR code to the patient by sending it through a secure channel.
7) Patient scans QR code: The patient uses a mobile device to scan the QR code and extract the Ethereum account details (address and private key). Then, the patient securely stores the Ethereum account details.

Following these steps, patient registration is completed, ensuring secure access to their Ethereum account for interactions with the private blockchain. It's essential to highlight that the patient's access is confined to their individual medical records, with interactions strictly governed by the smart contract.

C. The Patient's Medical Record Issuing Flow

Ensuring the confidentiality and integrity of medical records is a pivotal aspect of the proposed system. The process of issuing a patient's medical record and storing it on IPFS involves series of secure steps. These steps are described in Fig. 3 and detailed as follows:

1) Medical record upload: The process begins with the doctor initiating the upload of the patient's medical record to the blockchain.
2) Symmetric key generation: The DApp generates a unique symmetric key for each patient and for each new diagnosis. This approach ensures that access to medical records is controlled and traceable.
3) Record encryption: The patient's medical record is encrypted using the generated symmetric key. This encryption guarantees that only authorized entities can decipher and access the sensitive information.
4) IPFS upload: The encrypted medical record documents are securely uploaded to the IPFS network and the corresponding Content Identifier (CID) is then generated. This step enhances confidentiality, data integrity and accessibility.

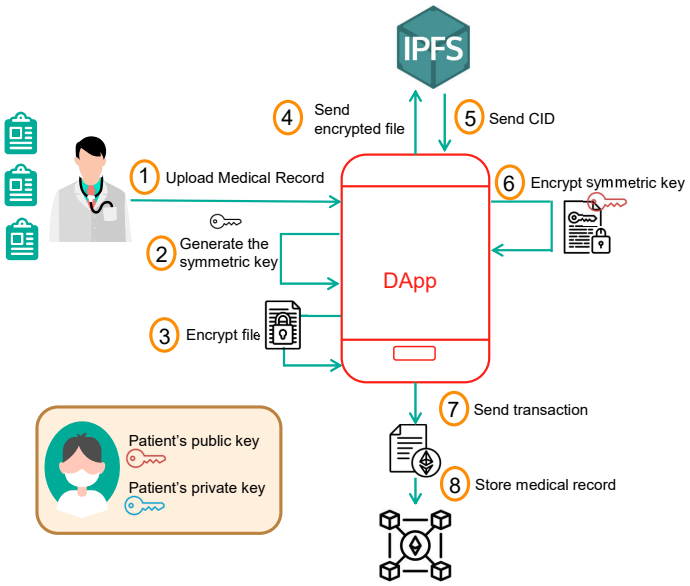

Fig. 3. Medical records issuing.

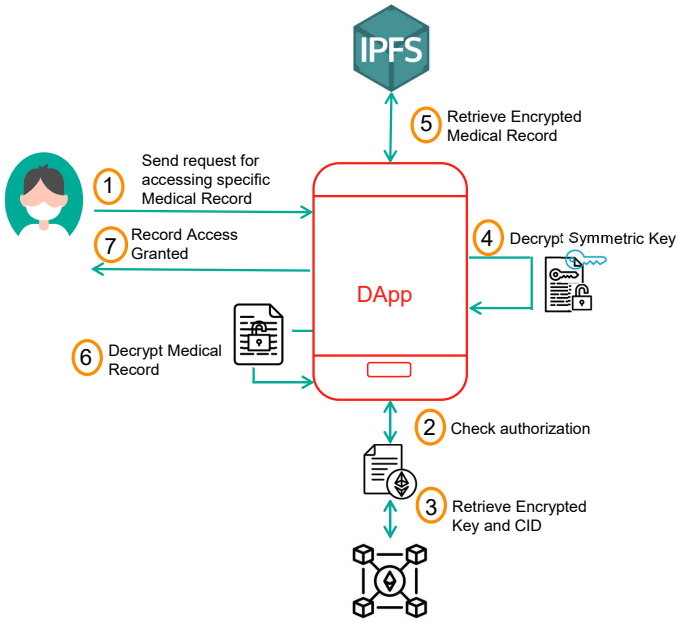

Fig. 4. The retrieval flow for the patient's medical record.

5) Send CID to DApp: The CID generated by IPFS is sent to the DApp, creating a reference point for accessing the complete medical record.
6) Key encryption: The symmetric key undergoes encryption with the patient's public key, sourced from their securely stored data on the blockchain. This step reinforces data security, ensuring that only the patient possesses the means to decrypt their medical records.
7) Initiate transaction: To securely store the encrypted symmetric key and the IPFS CID for every new set of patient medical records, it is imperative to initiate a new transaction on the blockchain through the execution of a smart contract.
8) Blockchain storage: This method ensures the immutability of both the key and CID associated with each medical record, thereby streamlining the secure retrieval of data from the IPFS.

D. The Medical Record Retrieval Flow

This section provides a detailed exploration of the steps involved in patients accessing their medical records within the proposed blockchain-based healthcare management system. Illustrated through Fig. 4, this section explains how patients can securely retrieve their medical information using the DApp, emphasizing the integration of security measures and a user-friendly interface for a controlled and confidential experience.

The steps involved in the medical records retrieving process are:

1) Initiate request: The patient or an authorized entity initiates a request for accessing specific medical records stored on the blockchain.
2) Authenticate identity: The system authenticates the identity of the requester, ensuring that only authorized individuals or entities can proceed with the retrieval process.
3) Retrieve encrypted key and CID: The DApp retrieves both the encrypted symmetric key and the CID associated with the requested medical record, which were initially stored in the blockchain during the record creation process.
4) Decrypt symmetric key: The requester, possessing the necessary decryption capabilities, decrypts the symmetric key using their private key. This step ensures secure access to the encrypted medical record.
5) Retrieve encrypted medical record from IPFS: The authorized party communicates with IPFS using the CID to retrieve the corresponding encrypted medical record from storage.
6) Decrypt medical record: The decrypted symmetric key is applied to decrypt the medical record, revealing the patient's health information in its original form.
7) Record access granted: The authorized entity now has access to the patient's medical record, facilitating the retrieval process securely and efficiently.

This comprehensive process ensures that sensitive health data remains confidential, with access granted only to authenticated and authorized entities while maintaining the integrity of the information stored on the blockchain and IPFS.

E. Granting Authorization to Patient's Medical Records

This section unveils the steps involved when patients grant access to their medical records to authorized entities. Illustrated through Fig. 5, this section elucidates the steps and security measures embedded in the process, ensuring controlled and secure access permissions for healthcare practitioners and authorized entities.

The detailed steps for granting access to patient's medical records are:

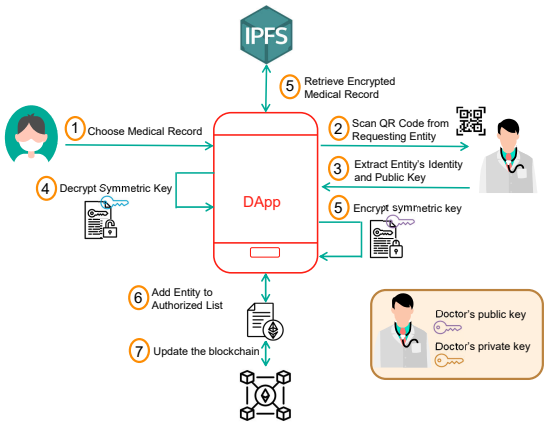

Fig. 5. Flow of authorizing access to patient's medical records.

1) Choose medical records: The DApp displays a list of the patient's health records. The patient initiates the process by selecting specific medical records for which access is to be granted.
2) Scan QR code from requesting entity: The patient scans the QR code provided by the entity requesting access to the medical records.
3) Extract entity's identity and public key: From the scanned QR code, the patient extracts information about the requesting entity's identity. The public key of the entity is retrieved from the QR code.
4) Decrypt symmetric key with patient's private key: The patient decrypts the symmetric key associated with the requested medical record using their own private key.
5) Encrypt symmetric key with entity's public key: The patient takes the decrypted symmetric key and encrypts it using the public key of the requesting entity.
6) Add entity to authorized list: The patient adds the address (identity) of the requesting entity to the list of authorized entities for the specific medical record, along with the encrypted symmetric key, now tied to the entity's public key.
7) Update blockchain: The blockchain is updated with the changes, reflecting the newly added entity to the authorized list and the associated encrypted symmetric key. This ensures a transparent and immutable record of access permissions.

This flow outlines the patient's actions, the interaction with the requesting entity, encryption and decryption processes, and the retrieval of medical records from the IPFS. The use of public and private keys, along with the secure exchange of encrypted symmetric keys, ensures controlled access to patient data while maintaining its confidentiality and integrity.

V. Smart Contract Functions

In this section, we delve into the heart of the proposed blockchain-based healthcare system's architecture-the implementation of our smart contract using the Solidity programming language. This smart contract encapsulate the rules and interactions governing the entire healthcare network. We explore the key functions embedded in these contract, each meticulously designed to manage entity registrations, handle medical records, and enforce access control. The transparency and immutability of blockchain technology, combined with the programmability of smart contract, form the backbone of the secure and efficient healthcare infrastructure we propose.

A. Implementation of the Entities Registration Process

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;

contract HealthRecords {

    address public ministryAddress;

    enum EntityType { MinistryOfHealth,
        LocalAuthority, Hospital, Clinic, Doctor }

    struct Entity {
        string name;
        address entityAddress;
        address registeringEntity;
        EntityType entityType;
    }

    mapping(address => Entity) public entities;
    mapping(address => bool) public
        isEntityRegistered;
    mapping(EntityType => mapping(EntityType => bool
        )) public entityRegistrationPermissions;

    constructor() {
        ministryAddress = msg.sender;
        isEntityRegistered[msg.sender] = true;
        entities[msg.sender].name = "
            MinistryOfHealth" ;
        entities[msg.sender].entityAddress =
            ministryAddress;
        entities[msg.sender].registeringEntity =
            ministryAddress;
        entities[msg.sender].entityType = EntityType
            .MinistryOfHealth;
    }
```

Listing 1: Entities and Constructor

```
    modifier onlyMinistryOfHealth() {
        require(msg.sender == ministryAddress, "Only
             the Ministry of Health can call this
            function");
        _;
    }

    modifier onlyDoctor() {
        require(isDoctorRegistered[msg.sender], "
            Only a Doctor can call this function");
        _;
    }

    modifier onlyPatient() {
        require(isPatientRegistered[msg.sender], "
            Only a Patient can call this function");
        _;
    }


  // Modifier for checking entity registration
      permission
  modifier onlyAllowedEntity(EntityType
      registeringEntityType, EntityType entityType)
      {
    require(entityRegistrationPermissions[
        registeringEntityType][entityType], "Entity
        registration not allowed");
```

```
    _;
}
```

Listing 2: List of modifiers

```
  // New function for setting entity registration permissions
    function setEntityRegistrationPermission(EntityType registeringEntityType, EntityType entityType, bool permission)
        external
        onlyMinistryOfHealth
    {
        entityRegistrationPermissions[registeringEntityType][entityType] = permission;
    }

function registerEntity(
    string memory _name,
    EntityType _entityType,
    address _entityAddress
) external onlyAllowedEntity(entities[msg.sender].entityType, _entityType) {
    require(!isEntityRegistered[_entityAddress], "Entity is already registered");

    Entity memory newEntity = Entity({
        name: _name,
        entityAddress: _entityAddress,  // Address of the entity being registered
        registeringEntity: msg.sender,   // Address of the entity initiating the registration
        entityType: _entityType
    });

    entities[_entityAddress] = newEntity;
    isEntityRegistered[_entityAddress] = true;
}
```

Listing 3: Process for registering entities

After deploying the smart contract, the pivotal setEntityRegistrationPermission function comes into play for configuring permissions in the entity registration process. The initial phase entails granting authorization to the Ministry of Health, empowering it to add local authorities, hospitals, or any other entities. This authorization can be established by either hardcoding it within the constructor or invoking the setEntityRegistrationPermission function, given that the Ministry of Health is exclusively empowered to define permissions, as ensured by the onlyMinistryOfHealth modifier. In the illustrated example showcased in Fig. 6, the Ministry of Health (enumerated as zero in the entities enum type within the smart contract) is accorded the privilege to add hospitals, denoted by the number two.

Fig. 6. Set entity registration permission.

Once the Ministry of Health has been granted the capability to add hospitals, the subsequent step involves registering the hospital's information and incorporating it into the list of hospitals. This is achieved by interacting with the smart contract through the invocation of the registerEntity function, as exemplified in Fig. 7. The crux of this process lies in the _entityAddress parameter, serving as the unique identifier for the hospital, corresponding to its blockchain address encapsulating account information. This identical procedure is employed for registering various entities, ensuring the permission to register a specific entity type through the onlyAllowedEntity modifier, inputting the entity details, and subsequently adding it to the blockchain via a smart contract request.

Fig. 7. Hospital registration.

The DApp interfaces play a crucial role in facilitating seamless interactions between different entities within the proposed blockchain-based healthcare management system. The interfaces for the Ministry of Health empower it to oversee and regulate the registration process of various healthcare entities. Through the interface presented in Fig. 8, the Ministry of Health gains the ability to list all registered entities, ensuring transparency in the system. Moreover, it can set permissions for entity registration, as shown in Fig. 9, enabling fine-grained control over the onboarding process. These functionalities empower the Ministry of Health to maintain a structured and secure healthcare ecosystem.

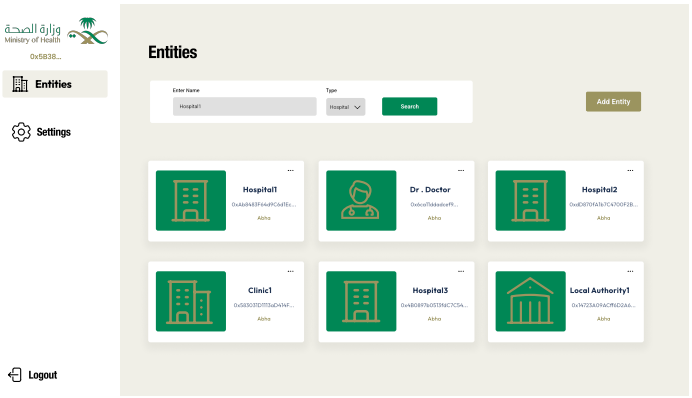

Fig. 8. List of entities

B. Implementation of the Doctors Registration Process

```
struct Doctor {
        string name;
        string phoneNumber;
        address doctorAddress; // Ethereum account address of the doctor
```

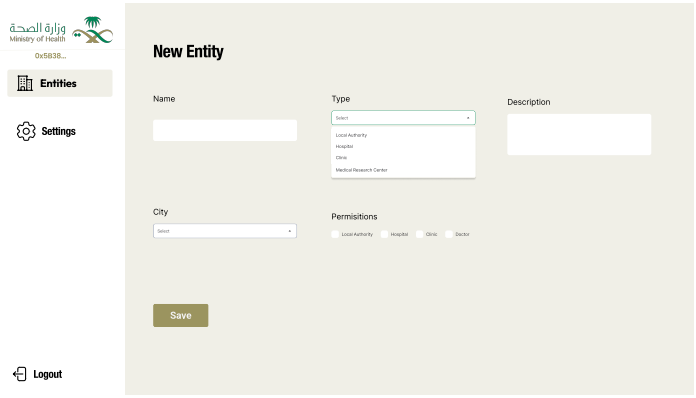

Fig. 9. Entity registration.

```
        bytes encryptionPublicKey; // Public key
            used for encryption
        string dateOfBirth;
        address registeringEntity; // Ethereum
            address of the entity who registered the
             doctor
    }

    mapping(address => Doctor) public doctors;
    mapping(address => bool) public
        isDoctorRegistered;


function registerDoctor(
    string memory _name,
    string memory _phoneNumber,
    address _doctorAddress,
    bytes memory _encryptionPublicKey,
    string memory _dateOfBirth
) external onlyAllowedEntity(entities[msg.sender].
    entityType, EntityType.Doctor) {
    require(!isDoctorRegistered[_doctorAddress], "
        Doctor is already registered");

    Doctor memory newDoctor = Doctor({
            name: _name,
            phoneNumber: _phoneNumber,
            doctorAddress: _doctorAddress,
            encryptionPublicKey:
                _encryptionPublicKey,
            dateOfBirth: _dateOfBirth,
            registeringEntity: msg.sender
        });

        doctors[_doctorAddress] = newDoctor;
        isDoctorRegistered[_doctorAddress] = true;
}
```

Listing 4: Process for registering doctors

Permitted entities have the capability to add a new doctor by providing details such as the doctor's name, email , and other pertinent information. Notably, as illustrated in Fig. 10, the doctor's _encryptionPublicKey is included. This key enables the doctor to access patients' medical records with their consent. The process involves the patient decrypting the encrypted symmetric key of a specific medical record with their private key and subsequently encrypting the symmetric key using the doctor's public key. Consequently, the doctor can decrypt the encrypted symmetric key and access a specific patient's medical record using his private key.

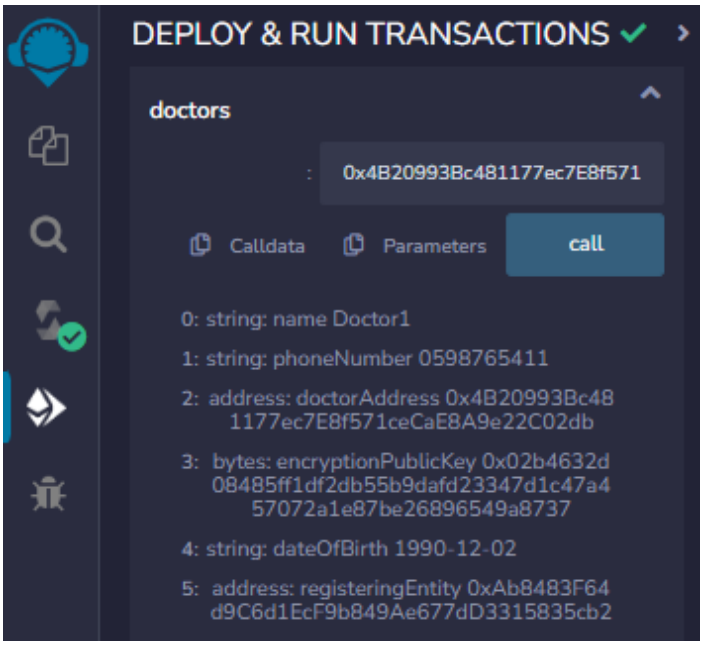

Fig. 10. Doctor registration.

The DApp interfaces for hospitals empower them by providing the capability to list all registered doctors, see Fig. 11. This functionality enhances the efficiency of hospitals in managing their medical staff. The second interface, presented in Fig. 12, enables hospitals to seamlessly register new doctors. Through this interface, hospitals can input and submit all necessary information about a new doctor. These interfaces collectively contribute to the effective coordination and administration of healthcare services within the proposed system, promoting a well-organized and responsive healthcare environment.

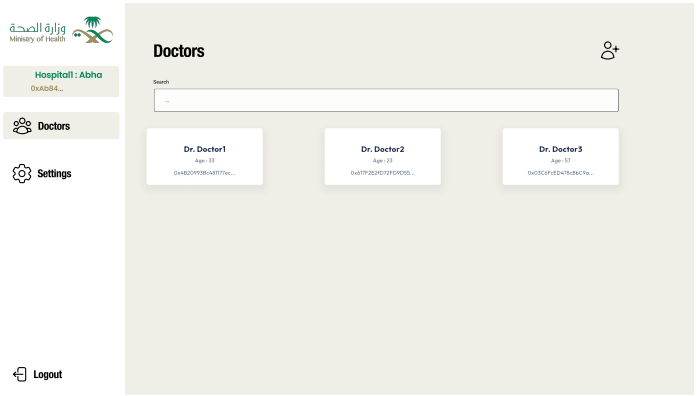

Fig. 11. List of doctors.

C. Implementation of the Patients Registration Process

```
    struct Patient {
        string name;
        string phoneNumber;
        address patientAddress; // Ethereum account
            address of the patient
        bytes encryptionPublicKey; // Public key
            used for encryption
```

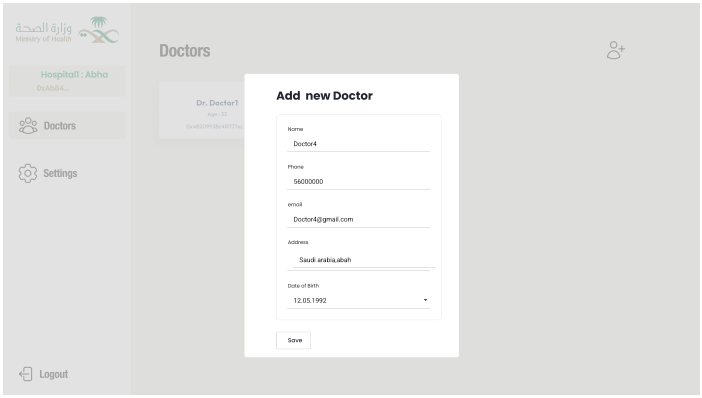

Fig. 12. Register new doctor.

```
        string dateOfBirth;
        address registeringDoctor; // Ethereum
            address of the doctor who registered the
             patient
    }

    mapping(address => Patient) public patients;
    mapping(address => bool) public
        isPatientRegistered;



    function registerPatient(
        string memory _name,
        string memory _phoneNumber,
        address _patientAddress,
        bytes memory _encryptionPublicKey,
        string memory _dateOfBirth
    ) external onlyDoctor {
        require(isDoctorRegistered[msg.sender], "
            Doctor is not registered");
        require(!isPatientRegistered[_patientAddress
            ], "Patient is already registered");

        Patient memory newPatient = Patient({
            name: _name,
            phoneNumber: _phoneNumber,
            patientAddress: _patientAddress,
            encryptionPublicKey:
                _encryptionPublicKey,
            dateOfBirth: _dateOfBirth,
            registeringDoctor: msg.sender
        });

        patients[_patientAddress] = newPatient;
        isPatientRegistered[_patientAddress] = true;
    }
```

Listing 5: Process for registering patients

Upon successful registration of the doctor by the hospital, the doctor acquires the capability to register new patients. Subsequently, the patient is included in the patient list by invoking the smart contract through the registerPatient function, as depicted in Fig. 13. The inclusion of the onlyDoctor modifier ensures that only authorized doctors can register patients. The doctor inputs the patient's information, which is then securely stored on the blockchain.

The designated interfaces for doctors in the DApp play a pivotal role in facilitating efficient patients management and registration processes. The first interface, shown in Fig. 14,

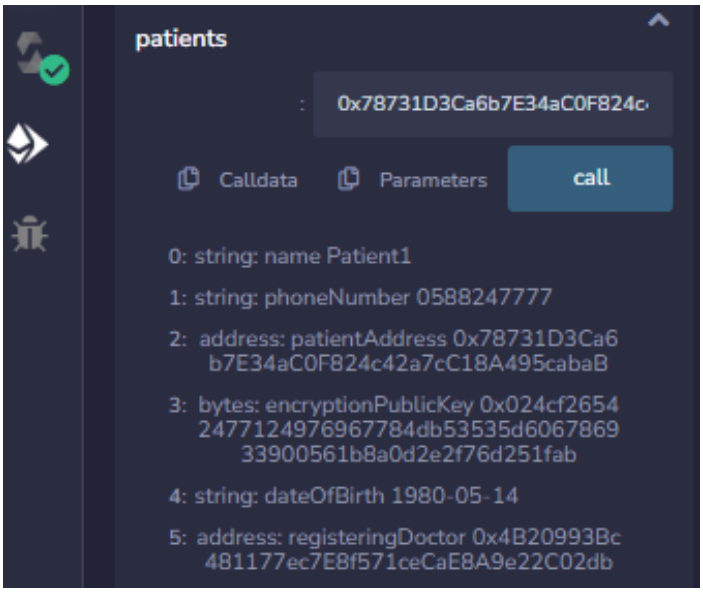

Fig. 13. Patient registration.

empowers doctors by providing the functionality to list all registered patients under their care. This feature enhances the doctor's ability to access and review patient information seamlessly, contributing to informed and personalized healthcare delivery. The second interface, presented in Fig. 15, serves as a key tool for doctors to register new patients. Through this interface, doctors can input and submit essential details about a new patient, ensuring a systematic and secure process for patient onboarding. In addition, the DApp initiates the creation of a new Ethereum account for the patient. Subsequently, the DApp communicates the Ethereum account details to the respective patient.

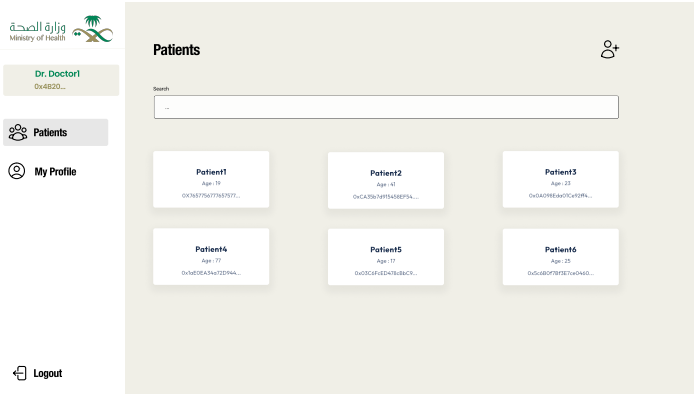

Fig. 14. List of patients.

D. Implementation of the Medical Records Issuing Process

```
struct MedicalRecord {
        uint256 id;
        address doctor;
        uint256 time;
        address patientAddress;
        bytes encryptedSymmetricKey;
        bytes patientPublicKey;
        string ipfsCID;
```

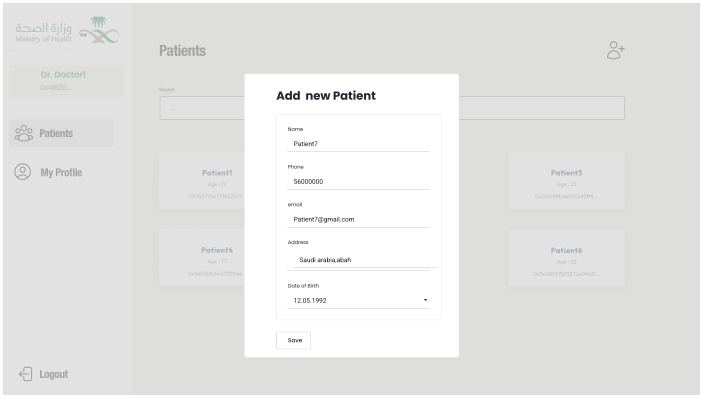

Fig. 15. Register new patient.

```
9       }
10
11      // Mapping to store medical records for each
            patient
12      mapping(address => mapping(uint256 =>
            MedicalRecord)) public patientMedicalRecords
            ;
13
14      // Mapping to store permissions for each medical
            record
15      mapping(address => mapping(uint256 => mapping(
            address => bytes))) public recordPermissions
            ;
16
17       // Function to add a medical record, restricted
            to doctors
18      function addMedicalRecord(
19          uint256 _id,
20          address _patientAddress,
21          bytes memory _encryptedSymmetricKey,
22          bytes memory _patientPublicKey,
23          string memory _ipfsCID
24      ) external onlyDoctor {
25          require(!isEntityRegistered[msg.sender], "
                Caller is not registered");
26          // Check if the patient is registered
27          require(isPatientRegistered[_patientAddress
                ], "Patient is not registered");
28
29          // Add the medical record to the patient's
                records
30          patientMedicalRecords[_patientAddress][_id]
                = MedicalRecord({
31              id: _id,
32              doctor: msg.sender,
33              time: block.timestamp,
34              patientAddress : _patientAddress,
35              encryptedSymmetricKey:
                    _encryptedSymmetricKey,
36              patientPublicKey: _patientPublicKey,
37              ipfsCID: _ipfsCID
38          });
39      }
```

Listing 6: Process for adding medical records

In Fig. 16, it is depicted that the patient's medical record details, including essential components like the _ipfsCID and the _encryptedSymmetricKey, will be entered through the account of the doctor. The _ipfsCID serves as a reference for retrieving the encrypted medical record from the IPFS, while the _encryptedSymmetricKey specifies the used encryption key, ensuring the confidentiality and privacy of the medical record.

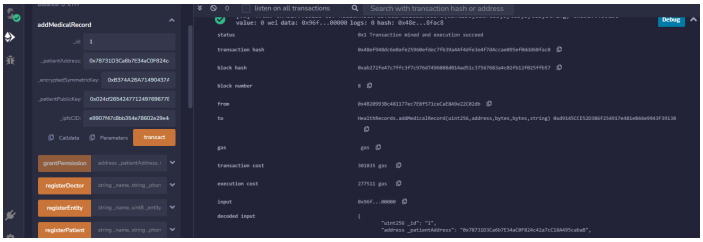

Fig. 16. Add medical record.

The interfaces designated for doctors within the proposed system encapsulate critical functionalities in managing and contributing to the medical records ecosystem. The first interface, shown in Fig. 17, empowers doctors to access and review medical records by listing all records associated with a specific patient. This functionality ensures efficient retrieval and oversight of a patient's medical history. In tandem, the second interface, illustrated in Fig. 18, equips doctors to actively contribute to the patient's medical records database. This involves a comprehensive process initiated by the DApp, orchestrating the generation of a symmetric key, encryption of medical records, and seamless integration with IPFS for secure storage.

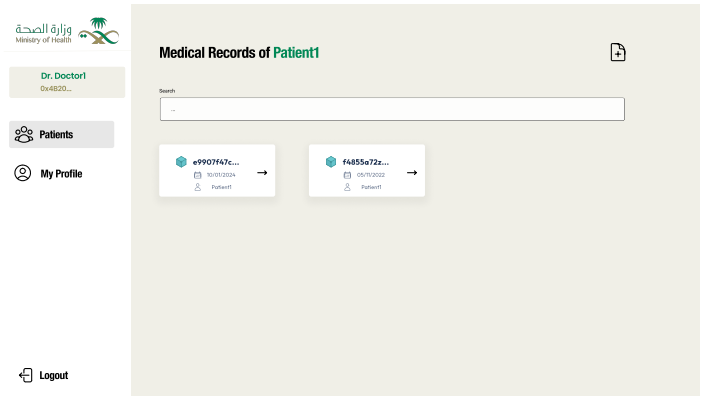

Fig. 17. List of medical records.

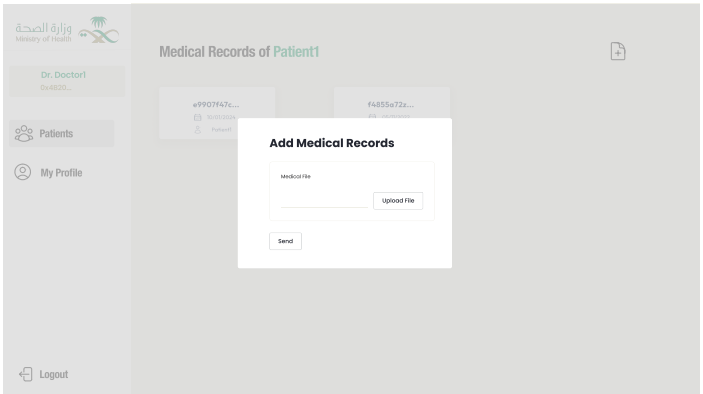

Fig. 18. Add new medical records.

E. Implementation of the Process for Managing Access to Medical Records

```
1
2       // Function to grant permission to an entity to
            access a specific medical record
3       function grantPermission(
4           address _patientAddress,
5           uint256 _recordId,
6           address _doctorAddress,
7           bytes memory _encryptedSymmetricKey
8       ) external onlyPatient {
9           // Grant permission to the doctor
10          recordPermissions[_patientAddress][_recordId
                ][_doctorAddress] =
                _encryptedSymmetricKey;
11      }
12
13      // Function to revoke permission from an entity
            for a specific medical record
14      function revokePermission(
15          address _patientAddress,
16          uint256 _recordId,
17          address _doctorAddress
18      ) external onlyPatient {
19          // Revoke permission from the entity
20          delete recordPermissions[_patientAddress][
                _recordId][_doctorAddress];
21      }
```

Listing 7: Process for managing access to the medical records

In the process of granting access to a specific medical record, the patient plays a central role in controlling the confidentiality and accessibility of their health data. Initiated by the patient, this action involves invoking the grantPermission function, a function guarded by the onlyPatient modifier to ensure that only the respective patient can execute this operation.

The key parameters in this operation include:

- _patientAddress: This parameter identifies the Ethereum address of the patient initiating the permission grant.
- _recordId: Serving as a unique identifier, this parameter denotes the specific medical record the patient intends to share.
- _doctorAddress: The Ethereum address of the doctor for whom the patient wishes to grant access to the designated medical record.
- _encryptedSymmetricKey: This parameter holds the encrypted symmetric key, by using the doctor's public key, associated with the medical record, ensuring that only the intended doctor can decrypt and access the sensitive health information.

By specifying these essential details, depicted in Fig. 19, the patient leverages the grantPermission function to securely share access to their medical record with a designated doctor, fostering a patient-centric approach to healthcare data management.

The process of revoking access to a specific medical record is a crucial aspect of patient-centric healthcare data management. Executed exclusively by the patient through the revokePermission function, this operation is safeguarded by

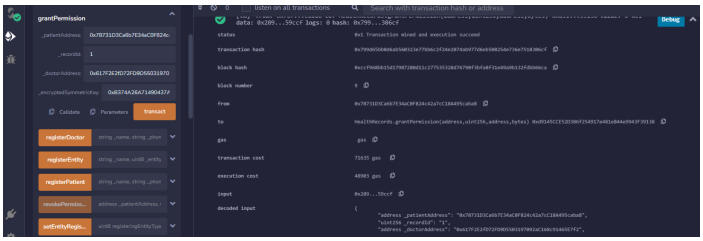

Fig. 19. Enabling the doctor to access the medical record.

the onlyPatient modifier, ensuring that only the authorized patient can control access to their health information. The pertinent parameters in this revocation process include the _patientAddress, the _recordId, and the _doctorAddress. By invoking the revokePermission function, the patient can selectively remove access privileges from a designated doctor, effectively enhancing the patient's control over the confidentiality and security of their medical records. This patient-driven approach empowers individuals to actively manage and regulate access to their health information, aligning with the principles of privacy and data security in the healthcare domain.

In Fig. 20, the illustration demonstrates the patient's capability to revoke a doctor's access to their medical record by deleting the associated medical record data from the doctor's list of accessible records.

Fig. 20. Revoke the access.

The patient interfaces, shown in Fig. 21, within the envisioned healthcare system afford individuals a comprehensive toolset for managing and accessing their medical records. The first interface provides patients with a detailed listing of their medical records. Crucially, this feature enables patients to review the doctors who currently possess access to each record, empowering them to make informed decisions about their healthcare providers.

Moreover, these interfaces enable patients to actively manage access permissions to their medical records. This involves granting access to new doctors, while also offering the ability to selectively revoke access when needed. By seamlessly integrating cryptographic principles and smart contract functionality, this patient-centric approach ensures the confidentiality and privacy of medical records. Patients are empowered with the agency to determine who can contribute to their healthcare information and underlines the commitment of the system to prioritizing user control and data privacy.

In addition to the core functionalities, the smart contract encompasses several functions that contribute to the overall robustness and flexibility of the proposed system, such as

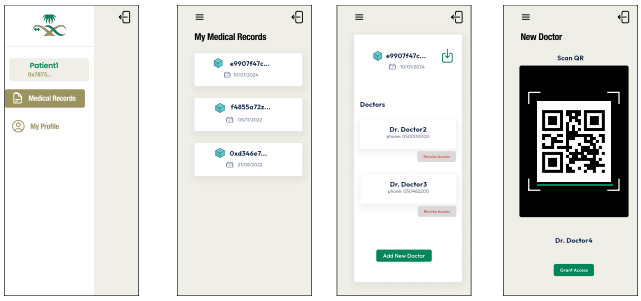

Fig. 21. Managing access to medical records.

functions that enable patients to delete specific medical records from the blockchain, ensuring a patient-centric approach to data management.

VI. Discussion

This section delves into a comprehensive analysis of the proposed system, exploring various facets such as architecture, security considerations, and frameworks employed.

- Architecture: The proposed system adopts a decentralized architecture built on a private Ethereum blockchain. This choice is motivated by the need for transparency, immutability, and distributed control. The utilization of a PoA consensus mechanism, implemented through the Geth client, enhances scalability and efficiency in a closed and permissioned healthcare network. The private Ethereum blockchain is primarily employed for defining rules to manage entities, such as hospitals, clinics, doctors, and patients, facilitating secure and controlled interactions within the healthcare network, while medical records, sensitive in nature, remain off-chain to address privacy concerns and storage limitations on the blockchain.
- Frameworks: The system integrates several key frameworks to facilitate its functionality. The Ethereum blockchain, along with the use of smart contract, forms the backbone for managing entities, doctors, patients, and medical records. The use of IPFS augments data storage capabilities, enabling decentralized and secure storage of medical records. React is employed on the front end to develop a user-friendly DApp, while Web3.js connects the DApp to the Ethereum blockchain, allowing seamless interaction with smart contract.
- Security considerations: Security is paramount in healthcare systems, especially when dealing with sensitive patient data. The use of a private Ethereum blockchain enhances security by restricting access to authorized entities, mitigating the risk of unauthorized access or tampering. Encryption plays a pivotal role in securing medical records, with each record encrypted using a symmetric key. Patient-controlled access through encrypted keys ensures that only authorized individuals, namely patients and authorized doctors, can decrypt and access medical records. Additionally, the private IPFS ensures that data retrieval is restricted to registered patients and doctors, adding an extra layer of security.
- Scalability: The proposed system exhibits scalability through a hierarchical structure. Local health authorities, hospitals, and clinics serve as nodes, ensuring a scalable network where each entity is added by a higher authority. This hierarchical structure allows for the efficient expansion of the healthcare network without compromising security.
- Regulatory compliance: The proposed system considers the hierarchical structure of healthcare authorities in Saudi Arabia, aligning with the regulatory framework. The Ministry of Health, serving as the highest authority, oversees the registration of entities, ensuring compliance with local regulations and preventing security issues such as DoS attacks.
- Patient-centric control: The architecture ensures that patients have granular control over their medical records. The process of granting access, revoking access, and managing encryption keys is in the hands of the patients, enhancing privacy and control over their healthcare data.

The proposed healthcare data management system, leveraging blockchain, IPFS, and encryption, exhibits resilience against several potential attacks, ensuring the integrity, privacy, and security of patient data. Here is a detailed analysis of the system's robustness against various attacks:

- Unauthorized access and tampering: The decentralized architecture and permissioned nature of the private Ethereum blockchain are pivotal in safeguarding against unauthorized access and tampering of patient data. By employing smart contract, the system enforces strict access controls, ensuring that only registered and authorized entities have the right to interact with the blockchain. The immutability of records is guaranteed through the consensus mechanism, making it virtually impossible for any entity without proper authorization to alter or tamper with sensitive medical information.
- DoS attacks: To counter the risk of DoS attacks, the blockchain-based healthcare data management system implements a hierarchical registration process and a permissioned structure. Entities, including hospitals and clinics, are registered by higher authorities, mitigating the potential for a flood of unauthenticated registrations. This hierarchical and permissioned approach enhances the system's resilience against DoS attacks, ensuring stable and secure operation.
- Data interception and eavesdropping: Protection against data interception and eavesdropping is achieved through robust encryption techniques. Medical records are encrypted using a symmetric key, which is then further encrypted with the patient's public key. This double-layered encryption ensures the confidentiality of patient data during both transmission and storage. Even if data is intercepted, it remains

unreadable without the patient's private key, providing a robust defense against unauthorized access.

- Sybil attacks: The proposed system is resilient against Sybil attacks through its hierarchical entity registration and permissioned blockchain architecture. Entities, such as hospitals and clinics, undergo controlled registration by higher authorities, preventing the creation of fake entities within the system. The permissioned blockchain further restricts participation to authorized nodes, effectively mitigating the risk of Sybil attacks by maintaining a trustworthy and controlled network.
- Blockchain consensus attacks: The system's utilization of a PoA consensus mechanism enhances its resistance against blockchain consensus attacks. PoA ensures that only authenticated and authorized nodes participate in the consensus process, eliminating the vulnerabilities associated with traditional PoW or PoS blockchains. This consensus mechanism contributes to the overall security and stability of the blockchain network, making it robust against consensus-related attacks.
- Data leakage prevention from IPFS: To mitigate the risk of data leakage from the IPFS, a combination of patient-controlled access and the use of a private IPFS is employed. Authorized patients and doctors, possessing the required private keys, are the only entities capable of retrieving and decrypting medical records stored on the IPFS network. The implementation of a private IPFS structure ensures that even in the event of an unauthorized intrusion into the network, attackers cannot access or decipher the stored medical records without the necessary cryptographic keys. This approach upholds the confidentiality of patient information, providing an additional layer of security against potential data breaches.

VII. Conclusion

In conclusion, this research introduces a robust and secure framework for managing healthcare data through the integration of blockchain technology, DApps, and a novel access control mechanism. The proposed system leverages a private Ethereum blockchain, enhancing data security and privacy within the healthcare domain. Through strategic smart contract implementations, the hierarchical structure of healthcare entities is mirrored, ensuring a controlled onboarding process and mitigating risks associated with unauthorized access.

The utilization of IPFS for secure and decentralized storage, combined with encryption methodologies, safeguards the confidentiality and integrity of medical records. The seamless interaction between the Ethereum blockchain and the React-based DApp provides a user-friendly experience for healthcare authorities, doctors, and patients. This research also details a comprehensive set of interfaces tailored for each entity type, facilitating smooth interactions and data management.

Furthermore, the proposed access control mechanism places patients at the center of their healthcare journey, allowing them to actively manage and monitor access to their medical records. The cryptographic principles underpinning this mechanism ensure secure key management and data confidentiality. Through the decentralized nature of the system, trust is established among participating entities, fostering a secure and transparent healthcare ecosystem.

As we move forward, this research provides a foundation for the continued exploration and development of blockchain-based healthcare systems. The presented framework not only addresses current challenges in healthcare data management but also aligns with the evolving landscape of digital healthcare. With a commitment to user control, data privacy, and security, the proposed system presents a valuable contribution to the ongoing discourse on innovative solutions for healthcare information management.

References

[1] G. Jetley and H. Zhang, "Electronic health records in is research: Quality issues, essential thresholds and remedial actions," *Decision Support Systems*, vol. 126, p. 113137, 2019.

[2] K. Wisner, A. Lyndon, and C. A. Chesla, "The electronic health record's impact on nurses' cognitive work: An integrative review," *International Journal of Nursing Studies*, vol. 94, pp. 74–84, 2019.

[3] M. Hochman, "Electronic health records: a "quadruple win," a "quadruple failure," or simply time for a reboot?" pp. 397–399, 2018.

[4] W. W. Koczkodaj, M. Mazurek, D. Strzałka, A. Wolny-Dominiak, and M. Woodbury-Smith, "Electronic health record breaches as social indicators," *Social Indicators Research*, vol. 141, pp. 861–871, 2019.

[5] S. Altamimi, T. Storer, and A. Alzahrani, "The role of neutralisation techniques in violating hospitals privacy policies in saudi arabia," in *2018 4th International Conference on Information Management (ICIM)*. IEEE, 2018, pp. 133–140.

[6] S. Dhumwad, M. Sukhadeve, C. Naik, K. Manjunath, and S. Prabhu, "A peer to peer money transfer using sha256 and merkle tree," in *2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM)*. IEEE, 2017, pp. 40–43.

[7] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. Ieee, 2017, pp. 557–564.

[8] B. Sharma, C. N. Sekharan, and F. Zuo, "Merkle-tree based approach for ensuring integrity of electronic medical records," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018, pp. 983–987.

[9] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[10] A. Al Mamun, S. Azam, and C. Gritti, "Blockchain-based electronic health records management: a comprehensive review and future research direction," *IEEE Access*, vol. 10, pp. 5768–5789, 2022.

[11] A. S. Yadav, S. Agrawal, and D. S. Kushwaha, "Distributed ledger technology-based land transaction system with trusted nodes consensus mechanism," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6414–6424, 2022.

[12] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[13] S. Kumar, A. K. Bharti, and R. Amin, "Decentralized secure storage of medical records using blockchain and ipfs: A comparative analysis with future directions," *Security and Privacy*, vol. 4, no. 5, p. e162, 2021.

[14] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. I. Khalaf, "Hyperledger healthchain: patient-centric ipfs-based storage of health records," *Electronics*, vol. 10, no. 23, p. 3003, 2021.

[15] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[16] X. Liu, K. Muhammad, J. Lloret, Y.-W. Chen, and S.-M. Yuan, "Elastic and cost-effective data carrier architecture for smart contract in blockchain," *Future Generation Computer Systems*, vol. 100, pp. 590–599, 2019.

[17] C. D. Clack, V. A. Bakshi, and L. Braine, "Smart contract templates: foundations, design landscape and research directions," *arXiv preprint arXiv:1608.00771*, 2016.

[18] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 1.

[19] W. Zou, D. Lo, P. S. Kochhar, X.-B. D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, "Smart contract development: Challenges and opportunities," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, 2019.

[20] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[21] W.-M. Lee and W.-M. Lee, "Using the web3. js apis," *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pp. 169–198, 2019.

[22] T. Heston, "A case study in blockchain healthcare innovation," 2017.

[23] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE access*, vol. 5, pp. 14 757–14 767, 2017.

[24] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, "Medchain: a design of blockchain-based system for medical records access and permissions management," *IEEE Access*, vol. 7, pp. 164 595–164 613, 2019.

[25] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, and F. Wang, "Secure and trustable electronic medical records sharing using blockchain," in *AMIA annual symposium proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 650.

[26] A. Abid, S. Cheikhrouhou, S. Kallel, and M. Jmaiel, "Novidchain: Blockchain-based privacy-preserving platform for covid-19 test/vaccine certificates," *Software: Practice and Experience*, vol. 52, no. 4, pp. 841–867, 2022.

[27] J. Sun, L. Ren, S. Wang, and X. Yao, "A blockchain-based framework for electronic medical records sharing with fine-grained access control," *Plos one*, vol. 15, no. 10, p. e0239946, 2020.

[28] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd international conference on open and big data (OBD)*. IEEE, 2016, pp. 25–30.

[29] M. T. de Oliveira, L. H. Reis, R. C. Carrano, F. L. Seixas, D. C. Saade, C. V. Albuquerque, N. C. Fernandes, S. D. Olabarriaga, D. S. Medeiros, and D. M. Mattos, "Towards a blockchain-based secure electronic medical record for healthcare applications," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[30] N. Sammeta and L. Parthiban, "Hyperledger blockchain enabled secure medical record management with deep learning-based diagnosis model," *Complex & Intelligent Systems*, vol. 8, no. 1, pp. 625–640, 2022.

[31] K. Azbeg, O. Ouchetto, and S. Jai Andaloussi, "Blockmedcare: A healthcare system based on iot, blockchain and ipfs for data management security," *Egyptian Informatics Journal*, vol. 23, no. 2, pp. 329–343, 2022.

[32] J. W. Kim, S. J. Kim, W. C. Cha, and T. Kim, "A blockchain-applied personal health record application: Development and user experience," *Applied Sciences*, vol. 12, no. 4, 2022.

[33] G. Q. Butt, T. A. Sayed, R. Riaz, S. S. Rizvi, and A. Paul, "Secure healthcare record sharing mechanism with blockchain," *Applied Sciences*, vol. 12, no. 5, 2022.

[34] R. A. Abutaleb, S. S. Alqahtany, and T. A. Syed, "Integrity and privacy-aware, patient-centric health record access control framework using a blockchain," *Applied Sciences*, vol. 13, no. 2, 2023.