IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 27, 2020, accepted October 15, 2020, date of publication October 20, 2020, date of current version November 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032553

Blockchain for Giving Patients Control Over Their Medical Records

MOHAMMAD MOUSSA MADINE[1], (Member, IEEE), AMMAR AYMAN BATTAH[1], IBRAR YAQOOB[1], (Senior Member, IEEE), KHALED SALAH[1], (Senior Member, IEEE), RAJA JAYARAMAN[2], YOUSOF AL-HAMMADI[1], SASA PESIC[3], AND SAMER ELLAHHAM[4]

[1]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[3]ASU's Blockchain Research Laboratory, Arizona State University, Tempe, AZ 85281, USA
[4]Heart & Vascular Institute, Cleveland Clinic Abu Dhabi, Abu Dhabi, United Arab Emirates

Corresponding author: Ibrar Yaqoob (ibrar.yaqoob@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Award CIRA-2019-001.

ABSTRACT Personal health records (PHRs) are valuable assets to individuals because they enable them to integrate and manage their medical data. A PHR is an electronic application through which patients can manage their health information. Giving patients control over their medical data offers an advantageous realignment of the doctor-patient dynamic. However, today's PHR management systems fall short of giving reliable, traceable, trustful, and secure patients control over their medical data, which poses serious threats to their authenticity and accuracy. Moreover, most of the current approaches and systems leveraged for managing PHR are centralized that not only make medical data sharing difficult but also pose a risk of single point of failure problem. In this paper, we propose Ethereum blockchain-based smart contracts to give patients control over their data in a manner that is decentralized, immutable, transparent, traceable, trustful, and secure. The proposed system employs decentralized storage of interplanetary file systems (IPFS) and trusted reputation-based re-encryption oracles to securely fetch, store, and share patients' medical data. We present algorithms along with their full implementation details. We evaluate the proposed smart contracts using two important performance metrics, such as cost and correctness. Furthermore, we provide security analysis and discuss the generalization aspects of our solution. We outline the limitations of the proposed approach. We make the smart contract source code publicly available on Github.

INDEX TERMS Blockchain, Ethereum, smart contracts, personal health records, healthcare, access control.

I. INTRODUCTION

Personal health records (PHRs) have played a key role in enabling safer, more efficient, and consumer-driven healthcare systems. A PHR is a collection of data gathered from various sources, such as patients' devices, clinics, care centers, pharmacies, and care delivery organizations (CDOs). Patients have shown a large interest in controlling their medical records and decoupling them from health providers in recent years [1]. Electronic medical records (EMRs) and electronic health records (EHRs) are different than PHRs. An EHR is a digital collection of a patient's medical history in terms of diagnoses, medications, treatment plans, allergies, laboratory and test results, among others [2]. On the other hand, an electronic medical record (EMR) is a digital version of the paper chart in the clinician's office [3]. An EMR contains the medical and treatment history of the patients in one practice [3]. In a simpler term, an EMR is a narrower view of a patient's medical history [4]. One of the major differences between PHRs and EHRs is that a PHR is controlled by patients; whereas, an EHR is controlled by doctors. In PHRs, individuals own and manage their data collected from healthcare providers or medical institutions (MIs).

To capitalize on the need for PHR, several well-known companies are offering PHRs management services. Most notable examples include Google Health, Apple Health [5], and Practice Fusion. Such PHR systems aim to provide a user-friendly interface, support multiple MIs, and enable integration between existing solutions via application programming interfaces (APIs). Despite such advantages; however,

The associate editor coordinating the review of this manuscript and approving it for publication was Yan Huo.

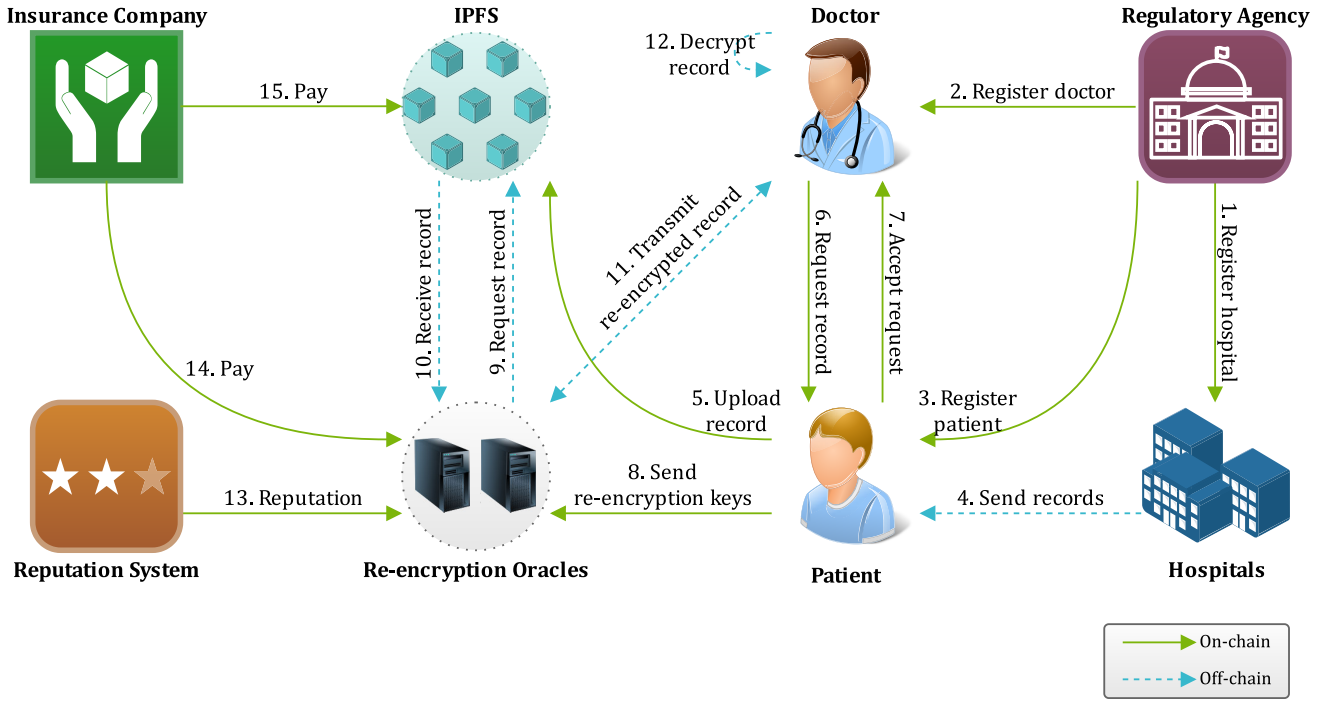

FIGURE 1. An overview of the main components of the system.

collecting data from MIs is a time consuming and tedious process. Although automated PHR solutions can enable individuals to manage their data efficiently, they take away the ownership of data from the patient due to the involvement of third parties. Besides, mostly such solutions are centralized and lack transparency, privacy, traceability, immutability, trust, and security features. An ideal patient-centered solution for a PHR system can only be designed by combining multiple features, such as immutability and provenance of records and patient-doctor interactions, resiliency against security attacks, and audit and accountability. These features and requirements nominate the blockchain technology as an ideal option for laying the groundwork for a decentralized, trustful, and secure PHR system.

Blockchain is a promising technology that has the potential to reshape the way data is being controlled or managed in existing PHRs management systems. It employs smart contracts to ensure that transaction processes are secure and traceable [6]. The decentralized architecture of blockchain can guarantee that the PHR is stored in a manner that is immutable, traceable, transparent, auditable, and secure. Also, blockchain can enable individuals to manage their health records information in such a way through which they can authorize certain entities (e.g., patients and health institutions the authority) to securely access and update their PHRs [7], [8]. Blockchain architectures are mainly of three types, such as public, private, and consortium. Based on the specific needs and requirements of individuals, these blockchain architectures can be employed as they can help to meet the objectives of different use case scenarios. Undoubtedly, blockchain can bring major improvements in the existing PHR systems. In certain use case scenarios, blockchain technology requires pairing it with some complementary technologies, such as IntraPlanetary File System (IPFS), trusted oracles, reputation systems, and proxy re-encryption. IPFS is a decentralized peer-to-peer storage system. Integrating blockchain with IPFS can help to overcome the issue of large-size file storage in existing blockchain systems. Trusted oracles can be used to retrieve medical records in a trustful manner. On the other hand, a reputation system can assist to stop/lessen or prevent oracle misbehaviors. The proxy re-encryption scheme helps to preserve the privacy of medical records and ensure they can only be shared with intended doctors. Based on the high merits and favorable features of blockchain technology, we propose leveraging blockchain technology for PHR management systems. Specifically, the main focus of our proposal is to enable decentralized access control for medical records between a patient and a doctor, along with interacting with various other entities, such as IPFS and trusted oracles, as shown in Figure 1. Note that our proposed system design does not focus on other aspects, such as standardization of medical record file formats, digital rights management, the inheritance of PHR data upon the death of a patient, and monetization of patient data.

A. RELATED WORKS AND CONTRIBUTIONS

Blockchain has the potential to bring major improvements and key innovations in the existing healthcare data management systems as discussed in [9]–[11]. To explore the potential of blockchain technology in the PHR systems, there have been numerous research efforts. For example, the authors

of [12], [13] have discussed how blockchain technology can be leveraged to facilitate patients to devise efficient access control policies and store health data in a secure and decentralized manner. On the other hand, several solutions have been proposed to extend the capabilities of typical blockchain technology by adding additional features, such as bottom-up design, robust data provenance, accountability, and decentralization. On top of that, the studies conducted in [14]–[16] investigate certain methods used to grant access to medical data by incorporating multi-signature technology into the blockchain architecture. The major limitations of these solutions are that they are partially decentralized and neither tested nor verified in a real-world blockchain ecosystem.

HealthBank [17] has proposed a trusted ecosystem that enables end-users to manage and control their healthcare data. The solution is not only general data protection regulation (GDPR) compliant but it also offers a wide range of features including user-friendly interface, complex data encryption, immutability, and accountability. Factom [18] has employed blockchain technology to ensure the integrity of patients' medical records while providing complete transparency and maintaining their privacy.

In [19], the authors have presented an approach that requires all entities of a typical PHR system to be on the chain; whereas, the encrypted medical data is stored on a separate centralized storage server to enable faster and low-cost access. One of the key limitations of the approach is routing the medical records through a transaction call, which ends up storing the entire medical record files on the chain. Centralization is another limitation of this approach because the process of retrieving medical records can be compromised internally by the datastore owner or externally through attacks, such as the denial of service (DDoS) attack. MedRec [20] is another approach that aims to resolve the issue of storing large chunks of data in the ledger by offloading them to the centralized database. Only pointers' information is stored in the ledger. However, the proposal does not discuss how medical records can be encrypted before uploading them to centralized servers. Another disadvantage of the approach is that it involves third parties that make it vulnerable to security attacks and pose the risk of a single point of failure problem.

The study conducted in [21] proposes the fast healthcare interoperability resources (FHIR) prototype that enables patients to securely and scalably share their clinical data using blockchain technology. Another approach proposed in [22] considers the hospital as the medical record creator and the patient as the owner. In this approach, the process of sharing medical records is not fully decentralized because all the data related to retrieval, querying, and doctor requests are executed off-chain. In [23], the concept of using hospitals for data storage and managing access permissions has been introduced. However, one of the major limitations is that through this solution patients do not have full control over their data because it is stored in hospitals.

Iryo is a healthcare ecosystem that employs blockchain technology to decentralize access to medical records [24]. It uses NuCypher key management system (KMS) to address the limitations of using consensus networks for securely storing and manipulating encrypted data [25]. Nucypher offers encryption and cryptographic access control through proxy re-encryption. Despite many advantages of the solution; however, utilizing NuCypher is costly because nodes in the NuCypher network need to be incentivized to prevent misbehavior.

The authors of [26] have employed blockchain technology to efficiently maintain patient records in terms of privacy, scalability, and availability. The solution encrypts patients' data with their public keys. It uses a proxy re-encryption mechanism on the centralized server for transferring the encrypted data from the patient to the doctor. In this solution, the patient-centered aspect is still missing because medical records are under the control of hospitals. Another limitation is that the process of re-encryption is conducted on a single server.

In summary, most of the existing healthcare data management systems are centralized and fall short to give patients control over their health records in a traceable, trustful, and secure manner. They are unable to trace and track PHRs in a tamper-proof and transparent manner. Also, the existing literature lacks patient-centric solutions. Such limitations can be overcome by integrating blockchain technology with PHR management systems. In this paper, we propose a blockchain-based architecture to manage access control of PHR systems. Our proposed approach decentralizes all the patient-doctor interactions. Our solution integrates multiple technologies to alleviate the typical limitations of blockchain technology in terms of large-size data storage and program execution. Our key contributions are summarized below:

1) We showcase a blockchain-based approach for patient-centered PHRs that constitutes a fully secure and decentralized architecture with complete medical record provenance and immutability.
2) We develop smart contracts and propose algorithms to implement the functions, modifiers, and trigger events. The implementation code is made publicly available.[1]
3) We integrate our blockchain-based system with the IPFS and trusted reputations-based oracles to securely fetch, store, and retrieve PHRs. We incorporate a proxy re-encryption scheme to preserve the privacy of medical records and ensure they can only be shared with intended doctors.
4) We present cost and security analysis, and perform correctness verification to evaluate the limitations, reliability, and practicality of the proposed solution.
5) We propose a generic solution that can be customized and implemented on public or private blockchains based on the needs and preferences of healthcare industries.

[1] https://github.com/madmoh/patient-phr

The remainder of the paper is organized as follows. In Section II, we present the proposed approach by explaining the different entities and technologies involved in the solution. Section III presents the design, implementation, and evaluation details. In Section IV, we provide a detailed discussion on how the proposed solution meets the crucial requirements along with the security analysis and limitations of the study. We present conclusion in Section V.

II. PROPOSED BLOCKCHAIN-BASED SOLUTION

In this section, we present the details of our proposed Ethereum blockchain-based solution along with its system components, such as proxy re-encryption, trusted oracles, reputation systems, and IPFS. We also explain the system architecture and sequential interactions between the entities and the smart contracts.

A. ETHEREUM

Ethereum is a public blockchain platform that enables developers to deploy decentralized applications through smart contracts. Ethereum smart contracts are executed using Ethereum virtual machines (EVMs). Ether is the native cryptocurrency used on the Ethereum blockchain. Gas is the unit used to measure the cost of executing a function in the smart contract. The average price of gas is about 20Gwei, where 1wei $=$ 10^{-18}Ether. To ensure that all the distributed EVMs follow their agreements in terms of execution, Ethereum uses Ethash, which is a proof-of-work (PoW) function [27], [28].

B. PROXY RE-ENCRYPTION

Proxy re-encryption schemes are cryptosystems that enable third parties to re-encrypt the ciphertext that has already been encrypted by one party. The notable examples of re-encryption schemes include Ateniese, Fu, Green and Hohenberger (AFGH) [29]. To further improve the security and efficiency of the classic approaches, several solutions have already been proposed [30], [31]. In general, proxy re-encryption schemes consist of the following functions:

1) **Key generation**: Generates the public and private key pairs (k^p, k^s) of the patient (k_P^p, k_P^s) and doctor (k_D^p, k_D^s).
2) **Encryption**: Encrypts a message m with a certain key k to get encrypted message $m_k = E(m, k)$.
3) **Decryption**: Decrypts a message m_k with the counterpart key k^{-1} to get the original message m.
4) **Re-encryption key generation**: Patients employ their private keys, and the public keys of the doctors to generate the re-encryption key $k_{P\rightarrow D}$.
5) **Re-encryption**: Changes the encrypted message from m_{k_P} to m_{k_D} using $k_{P\rightarrow D}$.

C. TRUSTED ORACLES AND REPUTATION SYSTEM

Oracles are trusted computation nodes that execute their software off-chain and report back to a certain smart contract operating on the blockchain. In our solution, there are two types of trusted oracles. Oracles of the first type act as proxy re-encryption nodes. They are capable of fetching data from the IPFS network and send it to the doctor after re-encrypting it. The second type of oracles is used for timeouts and time-based event triggers, which are crucial since such functionality cannot be natively supported in the Solidity language. Ethereum Alarm Clock (EAC) is one example of the second type of oracles [32].

A reputation system that keeps track of the oracles' behavior is important to avoid misbehaving oracles. Our reputation system design evaluates the oracles on two measures: based on its interactions with the smart contracts, and based on its interactions with the doctor. The main advantage of the reputation system is that it helps to identify the misbehaving oracle nodes by giving them low rating scores which can cause their removal.

D. OVERALL SYSTEM ARCHITECTURE

In our approach, all entities, except for the decentralized storage must be registered on the blockchain network. Figure 1 depicts the main components of the system, which are discussed below:

- **Regulatory Agency**: The government or a trusted public authority that is responsible for registering hospitals, patients, and doctors, in addition to overseeing the general process.
- **Hospital**: An entity that communicates with the patient to generate a symmetric key unique to each medical record file. Hospitals are responsible for transferring the medical records and their associated keys to the patient directly. Ideally, this process is automated and requires no human interaction, which is possible with the Internet of things (IoT)-enabled hardware and software components [33], [34].
- **Patient**: A PHR software that could be deployed on a device belonging to a patient (personal smartphone or computer) or on a trusted third-party (TTP) automated PHR service. Patients are responsible for registering themselves into the system, deploying their smart contract, uploading and submitting the medical records, and responding to data queries from doctors (requests to share medical records). In this paper, we assume all patients can decide on accepting or rejecting the request from their doctors. Note that patients may choose some third party to deploy the smart contract.
- **Doctor**: An entity that requests encrypted medical record files and decrypts the files locally.
- **Trusted Re-encryption Oracles**: The trusted general-purpose nodes that work as a re-encryption proxy to re-encrypt the symmetric keys from the patient to the doctor. Oracle nodes execute their programs off-chain, and therefore must be incentivized using a reputation system to avoid misbehavior.
- **Decentralized Database Storage**: The off-chain nodes that are used to store the encrypted medical record files along with the encrypted symmetric keys. One of the

possible decentralized database storage services is IPFS. These nodes store the files voluntarily, and therefore must be incentivized using proof-of-stake or a reputation system technique.
- **Insurance Company**: Responsible for paying the decentralized storage and oracle nodes.

Two smart contracts that are responsible for managing the above-mentioned entities are listed below:

- **Controller Smart Contract (CSC)**: It is deployed once and responsible for registering the above entities, as well as keeping track of the reputations of involved oracles. The smart contract allows the doctor to submit an evaluation of their interaction with the oracle.
- **Patient Records Smart Contract (PRSC)**: It is deployed once per patient and responsible for storing metadata about patient records and requests made by doctors. It allows the patients to respond to data access requests, and accepts oracles participation to send the records to the doctor. It is also responsible for evaluating the oracles and selecting the most reputable one.

E. INTERACTIONS AND MESSAGE SEQUENCE

A typical successful sequence of actions for receiving a medical record file from the hospital, and sharing it with a doctor is shown in Figure 2. The activities depicted in the sequence diagram start after all the entities have been registered and the medical records have been sent to the patient. The sequence of actions is as follows:

1) The patient generates a symmetric key against the medical record file to perform encryption. The public key k_P^p of the patient is used to encrypt the symmetric key. Both the encrypted medical record and the encrypted symmetric key files are uploaded on the decentralized storage, and the hash of the encrypted medical record file is stored on-chain.
2) The doctor queries the available medical record files. This communication takes place off-chain. Once the doctor decides what medical records need (based on the recorded metadata), the patient is notified with a data request. The patient decides whether to accept the request or deny it, which is done by sending the response as a transaction to their personal PRSC.
3) In case a patient accepts the request, the patient generates a re-encryption key (through the PHR software) and sends it to the PRSC. At this point, this smart contract informs the doctor and the oracles that a request for data has been granted.
4) The oracles will fetch the requested file from the IPFS. The file gets downloaded as a bundle, which contains both the medical record data and the encrypted symmetric key. The task for the oracles is to compute the hash of the encrypted symmetric key and send it to the patient smart contract.
5) Based on multiple responses from oracles, the PRSC determines which oracle had the correct response. This is decided by comparing the key hashes and deducing which oracle provided the fastest response. Based on these two factors, along with the previous reputation of the oracles, the system picks the most reputable oracle. At this point, a token is sent both to the doctor and to the selected oracle.
6) The doctor requests the medical record from the selected oracle by submitting the token. After acknowledging the correctness of the token, the oracle will re-encrypt the symmetric key using the re-encryption key generated by the patient, so it becomes encrypted by the public key of the doctor that initiated the request. Once the re-encryption process is done, the entire medical record bundle is sent to the doctor.
7) The doctor decrypts the re-encrypted symmetric key using their private key, revealing the plaintext symmetric key that was used to encrypt the medical record. Next, the doctor decrypts the medical record using the plaintext symmetric key, thereby getting the original readable medical record data.
8) Based on the doctor-oracle interaction, the doctor submits an honest rating to the controller smart contract, which will update the reputation of the oracle based on Equation 1, where $\overline{X}$ is the average reputation, N is the number of oracle interactions with doctors, and x_{N+1} is the new rating.

$$\overline{X}_{\text{new}} = \frac{N\overline{X}_{\text{current}} + x_{\text{N+1}}}{N+1} \tag{1}$$

We upload the medical record files on the decentralized storage (i.e., IPFS) to lessen the burden on the network and increase the efficiency of accessing the files. To ensure the traceability of the medical records, they are only considered valid once the hash of the file is registered in the smart contract. This makes the file available to the patient. It is crucial in this step to have the patient as the owner of the smart contract that governs the medical record. The PRSC stores metadata, such as title, creation date, and description of the file, in addition to the hash of the file. Furthermore, the PRSC keeps a log of all the access requests and responses.

An alternative to querying patient data off-chain is to search the distributed encrypted data directly in a decentralized manner. The authors in [35] have proposed an encrypted decentralized storage architecture that supports keyword searching to prevent returning the malicious results. A downside of implementing such a technique is the requirement of establishing a new storage architecture that can add complexity to the solution.

When a patient authorizes the doctor to access a certain medical record, he/she generates a re-encryption key that can, through the proxy re-encryption oracle nodes, atomically re-encrypt the medical record without disclosing the symmetric key to the proxy re-encryption nodes. At every interaction of the patient or the doctor with the trusted oracles, there is a possibility that some oracle nodes will misbehave. Therefore, on top of ensuring that a majority of the oracle nodes have a

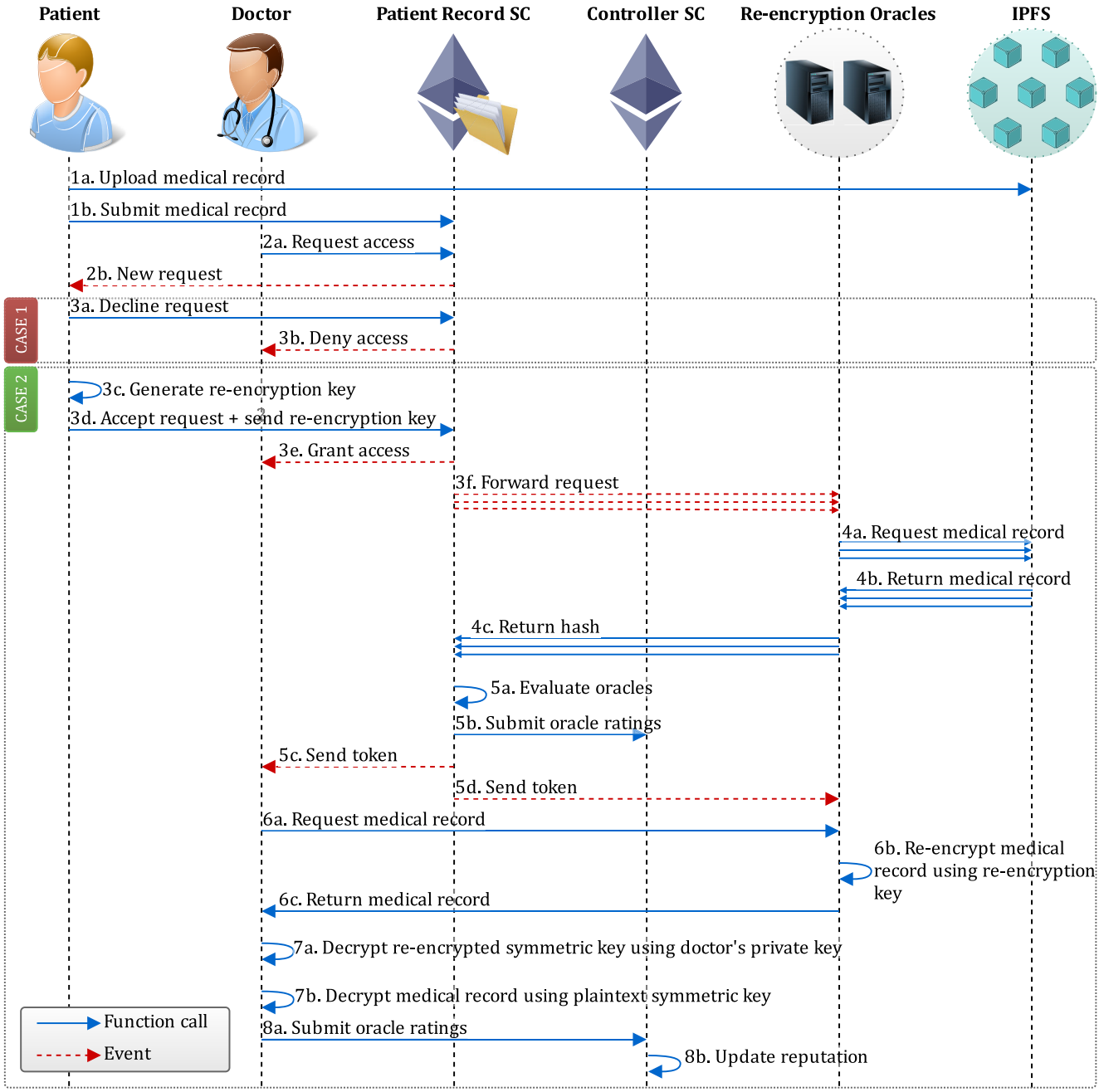

FIGURE 2. Sequence diagram of accessing medical records.

consensus on the result, we chose to implement a reputation system that reactively incentivizes the nodes to act properly.

Along with receiving the symmetric key, the doctor also attains the encrypted medical record file by requesting an access token from the PRSC. This is crucial because the doctor will not directly download the medical record files from the decentralized storage, but rather through the oracles. In this way, we ensure that attempts made by the doctor to download the medical record files are indisputably registered into the blockchain.

III. IMPLEMENTATION AND EVALUATION

In this section, we present the implementation details of our smart contracts developed in Solidity language. We use the online Remix IDE to write, compile, debug, and deploy the Solidity code. Testing was carried out by sending real Ethereum transactions using multiple accounts.

A. IMPLEMENTATION

Figure 3 shows the entity-relationship diagram to provide the necessary implementation details. It shows that the

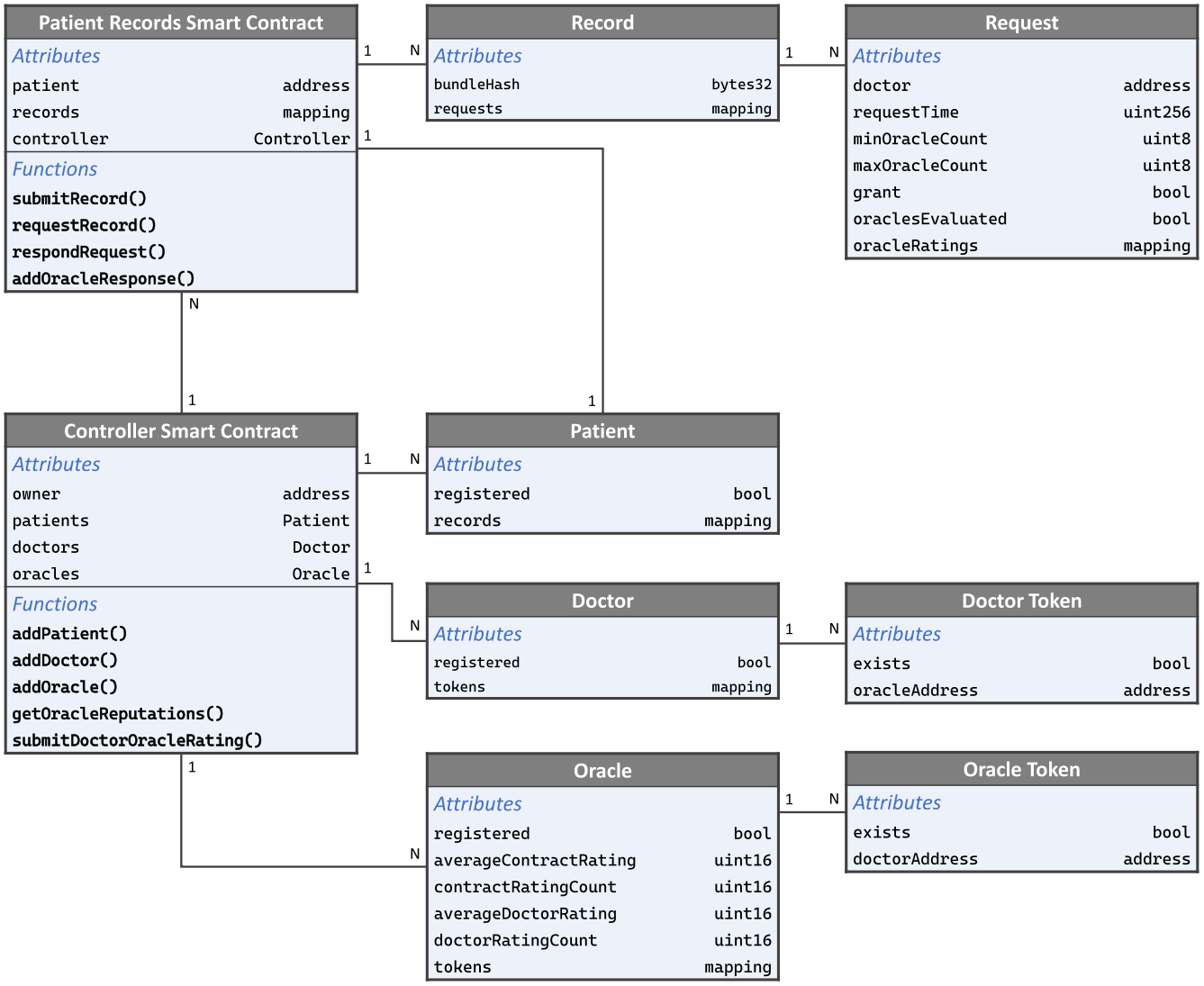

FIGURE 3. Entity relationship diagram.

`Controller` smart contract is deployed once by the regulatory agency, and it determines which entities are part of the network.

All the stakeholders are initially required to be a part of the Ethereum blockchain network. The first interaction is initiated by the patient, and it includes deploying a smart contract called `PatientRecord`. This contract is used for managing the patient's medical records. This smart contract is dedicated to one patient, and it stores an array of hashes of medical record bundles and their mapping to original medical records. Additionally, this smart contract is connected to a universal `Controller` smart contract, which governs the existing patients, doctors, oracles; and updates the reputations of oracles and controls the tokens of oracles. The patient submits a new medical record by calling the `submitRecord` function of the `PatientRecord` smart contract (by issuing a transaction on the Ethereum blockchain), as described in detail in Algorithm 1.

Algorithm 1 `submitRecord`: Submit New Medical Record

1 **Input**: new bundle hash *b*#
2 **Require**: owner patient only
3 Push *b*# to array of uploaded bundle hashes *B*#
4 Create new record *r* with empty requests list
5 Add *r* to array of records *R*
6 **Emit**: inform patient about successful record addition

Once a doctor wants to access a certain record of the patient, the doctor needs to call a `requestRecord` function of the `PatientRecord` smart contract, as discussed in Algorithm 2. The function takes several parameters to identify the desired medical record, the public key of the doctor, and the acceptable range of the number of oracles. The function verifies that the doctor supplied the correct public key by computing its hash, and performing a bit-wise *AND* operation

Algorithm 2 `requestRecord`: Request Medical Record

Input: medical record r, doctor public key k_D^p, oracle range $o_{\min}$, $o_{\max}$
Require: function caller is a doctor
Require: valid k_D^p
Require: $0 < o_{\min} \leq o_{\max}$
Create new request q with the attributes of the doctor address, current time, specified $o_{\min}$ and $o_{\max}$ parameters, false *grant* status, and false *oracles evaluated* status
Add q to the array of requests Q located inside of r
$r.c \leftarrow r.c + 1$, where $r.c$ is the number of requests for the medical record r
Emit: inform patient about k_D^p
Emit: inform doctor about function execution result

with $2^{20\times 8} - 1$, then ensuring the result is equal to the doctor Ethereum address. The exponent in $2^{20\times 8} - 1$ refers to the number of bits in an Ethereum address, which is 20 bytes. The function also ensures that the caller is a doctor, and the range of the number of oracles is valid. A new request structure is then created with the appropriate attributes provided by the doctor, which gets added to the array of requests in the patient's medical record. The function ends with informing the patient about the doctor's public key, such that the patient can generate the re-encryption oracle, and informs the doctor about the status of the execution result.

At this point, the patient is responsible for responding to the request by either granting it or denying it. If the patient grants access, the re-encryption key $k_{P\rightarrow D}$ will be sent with this event. This keeps updating the state variable and broadcasts to the oracles about a new/accepted request. The oracles, using the details in the broadcast, are only able to request the medical record bundle from the IPFS network. This bundle contains the encrypted key $k_{sP} = E(k_s, k_P^p)$ and using `keccak256`-hash $k_{sP}\#$ can be calculated.

The oracles subsequently call the `addOracleResponse` function described in Algorithm 3 and Algorithm 4. At this step, oracles request the medical record bundles from the IPFS by sending the bundle hash, then compute the hash of the encrypted symmetric key (found in the bundle) and send the result to the PRSC.

A malicious set of oracles may want to trick PRSC by quickly responding and submitting the same incorrect hash result; however, since PRSC already has the correct hash result stored privately, it can verify the correctness of oracle results without relying on a vulnerable majority vote mechanism. Therefore, there is no way for the oracles to bypass the process of retrieving the bundle from the IPFS.

Depending on the latency and correctness of the oracle response, the oracle is evaluated by the smart contract. This is performed by linearly mapping the oracle's latency to the range between 1 and 65,535. 1 is the minimum positive value in uint16, and in our case, it is used to indicate the highest latency (which we chose to be 1 hour). 65,535 is the maximum value in uint16, and in our case, it is used for minimum latency, which is 1 second. The initial value for the reputation is 32,768, which in this range's midpoint. As a result of processing the average reputations by the smart contract, they are immutably stored on the chain, and this is handled by the `Controller` smart contract.

Algorithm 3 `addOracleResponse`: Submit an Oracle Response

Input: request q, response hash $k_{sP}^O\#$
Require: q is granted by patient
Require: $q_s = false$, where q_s is the evaluation status of the request
$l \leftarrow t_0 - t_q$ where l is the latency, t_0 is the current time, t_q is the request time
Considering q_O is array of participating oracles...
if $len(q_O) < o_min$ ***or*** $(len(q_O) \geq o_min$ ***and*** $len(q_O) < o_max$ ***and*** $l \leq 1$ *hour)* **then**
    $c \leftarrow (k_{sP}^O\# = k_{sP}\#)$, where c is boolean to evaluate correct response, and $k_{sP}\#$ is the correct hash
    $n \leftarrow \frac{2^{16}(l-1)}{1\text{ hour}-1\text{ second}} + 2^{16} - 1$, where n is the oracle rating
    $n \leftarrow n \times c$
    Add oracle o to q_O
    Add n to array of ratings of the participating oracles N
end
if $len(q_O) \geq o_min$ ***and*** $l > 1$ *hour* ***or*** $len(q_O) = o_max$ **then**
    $q_s \leftarrow true$
    Call `evaluateOracles`
end

The doctor uses tokens to request the medical record bundle from the oracle. The oracle uses the re-encryption key to re-encrypt the files inside the medical record bundle (passing from the patient to the doctor), and will respond to the doctor with the re-encrypted bundle. At this point, the doctor evaluates the performance of the oracle and submits a rating from 1 to 65, 535 with an initial reputation of 32, 768. The submission is made by transacting a simple `submitDoctorOracleRating` function of the `Controller` smart contract. The rating given by the doctor ensures, over multiple requests, that even after an oracle quickly and correctly responded to the patient record smart contract, it will not maliciously send a different bundle file to the doctor, or not even respond to the doctor's request.

When choosing an oracle, the patient record smart contract uses the average of the two performance ratings as the new reputation score, and it multiplies the result by the square of the old reputation score. The old and new reputation scores are multiplied rather than added to ensure that an oracle that returns incorrect response hashes will never be selected as a winner.

Algorithm 4 `evaluateOracles`: Evaluate Performance of Oracles

Input: request q, ratings N
Query array of reputations of participating oracles G
$o_s \leftarrow q_O[0]$, where s is the selected oracle
$o_{s_s} \leftarrow N[0] \times (G[0]+1)^2$, where o_{s_n} is the score of the selected oracle
for $i \leftarrow 1$ **to** $len(q_O)$ **do**
  if $N[i] \times (G[i]+1)^2 \geq o_{s_s}$ **then**
    $o_s \leftarrow q_O[i]$
    $o_{s_s} \leftarrow N[i] \times (G[i]+1)^2$
  end
end
Submit oracle ratings using Equation 1
$k_i \leftarrow \text{hash}(d_{EA} \parallel o_{s_{EA}} \parallel t_0)$, where d_{EA} and $o_{s_{EA}}$ are the Ethereum address of the doctor and the selected oracles, respectively
Emit: send doctor token
Emit: send oracle token

The code was tested and verified for its functionality and completeness by going through the expected sequence of actions, starting from deploying the records, until submitting scores for each oracle. We performed the testing on JavaScript-based Ethereum Virtual Machines (EVMs) in an Ethereum test network (testnet).

B. COST ANALYSIS AND CORRECTNESS VERIFICATION

Herein, we present the cost analysis. We verify our implemented solution in terms of efficiency and correctness. The importance of efficiency in Solidity functions is linked to the reward that miners usually get after executing the functions. While executing a function, the miners keep track of the operations performed in the function that leads to measure the cost of execution based on the data types and number of operations.

1) COST ANALYSIS

We have implemented and deployed two smart contracts. Table 1 shows the transaction costs of the functions, execution costs expended by the miners, and how those costs are converted into USD. On April 10, 2020, Ether closed at $159.68, which is the conversion value used in the table. As for the gas price, we set it to 20 Gwei, which is comfortably above its average, currently floating between 10 Gwei and 15 Gwei.

The constructor calls inside the smart contracts make up the major share of the transaction and execution costs. On the other hand, the adder functions have a much lower cost as they only initialize the variables of their corresponding stakeholders. As for the remaining functions, starting with `submitRecord`, the costs are slightly increased, especially for the functions `requestRecord` and `addOracleRespnose` as they perform expensive checks, array creations, and loops.

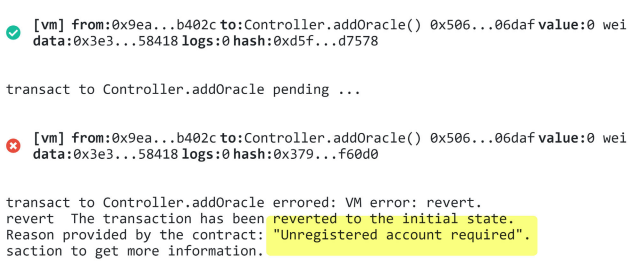

FIGURE 4. Attempting to add two oracles from the same Ethereum address.

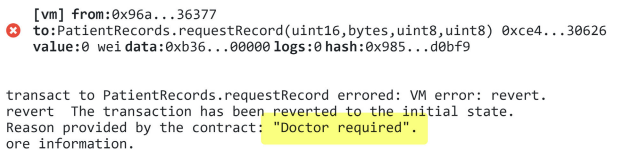

FIGURE 5. Non-doctor account attempting to request a medical record.

2) CORRECTNESS VERIFICATION

We perform testing of the proposed smart contracts under a simplified PHR environment to verify their correctness. Our verification process has comprised six major steps as discussed below.

1) First, `Controller` smart contract gets deployed. Subsequently, a patient, doctor, and three oracles are added using unique Ethereum addresses. In case of any address holder attempts to register again, the request will be rejected. This can be seen in Figure 4, where an Ethereum account attempts to register twice times as an oracle with the same address. As expected, the second time, execution gets failed. This requirement is important since we cannot let accounts reset their state in the network, which is even more crucial for oracles because their accumulated state overtime highlights their reputation. In case if such a requirement does not exist, oracles with low reputations would reset their ratings simply by registering again.
2) The patient adds two medical records. This step is relatively less crucial, and that is because the patient is dealing with his/her own smart contract.
3) The doctor requests both medical records of the patient and sets a minimum oracle count of 2 and a maximum oracle count of 3. This request can only be made by the doctor's account, and therefore, non-doctors cannot successfully execute this function. Figure 5 shows an example of the patient trying to request the medical record, which has eventually failed. After any request, regardless of its failure or success, the patient will be informed through an event message on their DApp.
4) The patient will respond to the request of the doctor, either by denying or granting access to the data. In our

TABLE 1. Gas and currency cost of smart contract functions.

Function Caller	Function Name	Transaction Cost [Gas]	Execution Cost [Gas]	Cost [USD]
Regulatory Agency	`Controller`	872,167	620,867	2.79
Patient	`PatientRecord`	1,193,125	862,033	3.82
Patient	`addPatient`	45,980	24,708	0.15
Doctor	`addDoctor`	46,072	24,800	0.15
Oracle	`addOracle`	89,698	68,426	0.29
Patient	`submitRecord`	80,707	57,099	0.26
Doctor	`requestRecord`	141,049	118,753	0.45
Patient	`respondRequest`	37,877	16,157	0.12
Oracle	`addOracleResponse`	30,316	6,676	0.10
Oracle	`addOracleResponse + evaluateOracles`	101,189	77,621	0.32
Doctor	`submitDoctorOracleRating`	53,818	28,834	0.17

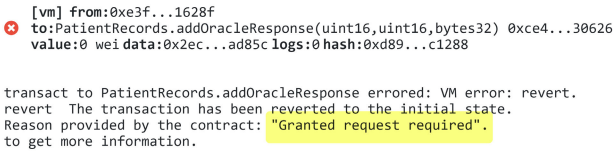

FIGURE 6. Oracle attempting to participate to denied request.

tests, the patient rejects the first request and accepts the second one.

5) Oracles will start to participate when there are accepted requests. However, in a certain case when oracle attempts to participate in a rejected request, the execution will fail, as shown in Figure 6. On the other hand, if enough oracles have participated, and met one of the three cases, the `evaluateOracles` function will be executed and will result in sending tokens to the selected oracle and requesting doctor, which is depicted in Figure 7.
6) The doctor will submit a rating for the interaction with the selected oracle. In the rate submission function, the doctor must specify the token identifier and the correct oracle. In case if the doctor fails to correctly specify either of the two values, the execution will fail. Figure 8 shows a specific case where the doctor inserted an incorrect oracle address.

The three cases mentioned in verification step 5 refer to the possible alternative conditions that are required in order to consider a request is completed. Such cases include:

1) Timeout (set to 1 hour in our tests) has occurred, but `minOracleCount` has not been reached yet. The moment `minOracleCount` is reached (function called by proxy re-encryption oracle), evaluation will begin.
2) `minOracleCount` is reached, but `maxOracle Count` is not. The moment the timeout occurs, an EAC oracle calls the evaluation function.
3) The moment `maxOracleCount` is reached before the timeout, the evaluation begins.

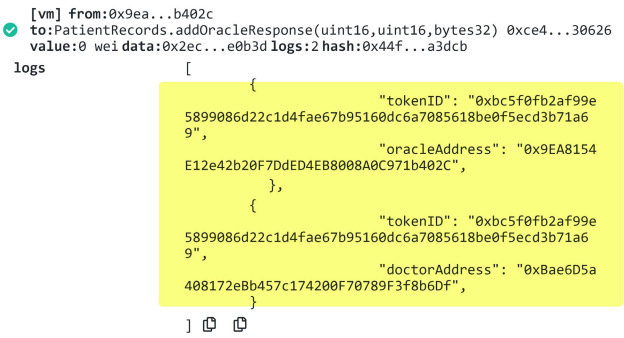

FIGURE 7. Successfully selecting most reputable oracle and sending tokens.

FIGURE 8. Doctor attempting to rate the oracle with invalid token.

IV. DISCUSSIONS

In this section, we analyze the proposed solution in terms of security, generalization, and limitations.

A. SECURITY ANALYSIS

The proposed solution enables patients to securely share their medical record files with their doctors. This is because our approach majorly decentralizes all aspects of the network, adds measures such as the reputation system to keep the parties from acting maliciously, and uses proxy re-encryption to guarantee atomic conversion of files from the patient format to the doctor format.

The proposed solution is based on a strict re-encryption scheme, which ensures confidentiality, as only the patient and the patient-chosen doctors can have access to the medical records. The re-encryption scheme does not expose the private keys of any entity, since it uses the re-encryption key generated by the patient using the patient's private key and

the doctor's public key in a one-way process that cannot be reversed. Furthermore, medical records are stored in a distributed and decentralized storage, such as IPFS, which enables patients to offload storing medical files. Using the proposed approach, the patients do not need to trust any centralized third party entity to store the files. This ensures that the stored data is secure enough against well-knows attacks (e.g., DDoS).

Using Ethereum as a foundation for our approach, allows the fundamental data flow to become fully traceable. Examples of such data include logs of requests to medical records, token creation and transmission, and reputation calculations, all along with their changes across time for full provenance ability. Furthermore, Ethereum gives the ability to authenticate that the addresses of stakeholders are never tampered with and can only refer to their legitimate entity. Our solution design ensures that one Ethereum address is not associated with multiple entities, which eliminates the cases of impersonation or Sybil attacks.

The system network is protected from the internal and external attacks. Security is ensured by multiple levels of protection. First is the decentralization of the network that eliminates the risk of the single point of failure problem, and therefore preventing downtime attacks. Second is limiting the accessibility of the functions to the registered identities and ensuring no function that modifies patient data is accessed by any other entity than the patient specifically. The third is implementing a reputation system that is resilient against the vulnerabilities caused by majority vote mechanisms, thereby preventing oracles from misbehaving when communicating with the smart contract.

The privacy of all entities in general and patients in specific is guaranteed, as a result of the anonymity of all identities and the encryption of all medical record data. Our proposed system does not store or depend on any database that reveals the physical identity of users. Moreover, patients do not disclose any personal information to blockchain, IPFS, or proxy re-encryption nodes even while sharing the medical records with their doctors.

B. GENERALIZATION

Although the proposed solution is targeting a specific use case, it can be generalized for a wide range of other problems. First, we can consider the patient as a general source of information. This is more appropriate than considering the hospital as the source because the source will become the entity that controls who can or cannot access the files. The doctor can be considered as any entity requesting private and sensitive information, which means in some cases the same user may want to act as both, either a source of information or the requester. The structure of oracles in our solution is flexible and can be adapted into other systems. The proposed approach is designed for a generic healthcare system, so it can either be tailored to the healthcare system of a specific country or made even more generalized in the context of universal environments.

TABLE 2. Comparison with existing solutions.

Aspect	Cloud-based PHR Management [36], [37]	Our Solution
Privacy	Yes	Yes
Decentralized Storage	No	Yes
Decentralized Execution	No	Yes
Patient-centered	Partially	Yes
Provenance	Partially	Yes
Immutability	Partially	Yes
Trustful	Partially	Yes

C. LIMITATIONS AND CHALLENGES

Herein, we identify and outline important challenges that pose limitations on the proposed blockchain-based solution for a patient-centered PHR system.

- **Interoperability**: Extending our approach to a global context would require multiple deployments of the smart contracts to interoperate among each other. For example, a registered patient that travels to another country must register again under the new country's `Controller` smart contract. Since the Ethereum blockchain does not offer integration across different deployments, the patient will not have a global view of their medical records. However, a possibility to mitigate this limitation is to rely on a global healthcare DApp that can perform the required integration.
- **Key management**: Even though the key management architecture of blockchain systems is reliable in terms of authenticating the patients; however, they lack user-friendly features and do not have any room of leniency in case the patients forget their wallet credentials.
- **GDPR**: As a result of the immutable nature of blockchain, all data stored on-chain cannot be taken off. Our system design partially mitigates this limitation by storing the medical records on IPFS, and thus the records can be deleted. However, the metadata is stored on-chain, which means it cannot be removed even if it is requested by the patient.
- **Smart contracts upgradability**: In Ethereum blockchain, smart contracts are stored on-chain, making them immutable. However, this poses a major challenge in the development process of smart contracts, as immutability makes them lack upgradability. Once smart contracts are developed and deployed, they can no longer be modified. Therefore, it is not possible to patch security vulnerabilities or software bugs with an update.

D. COMPARISON WITH THE CLOUD-BASED PHR MANAGEMENT SOLUTIONS

We compared our proposed solution with two existing cloud-based PHR management solutions [36], [37] as shown in Table 2. The table shows the superiority of the proposed solution as it employs blockchain technology, Ethereum smart contracts, distributed trusted oracles, distributed decentralized database storage, and proxy

re-encryption technology. The proposed solution achieves all of the requirements set by our initial design.

V. CONCLUSION

In this paper, we have proposed a blockchain-based approach to give patients control over their medical records in a decentralized, traceable, reliable, trustful, and secure manner. We developed two Ethereum-based smart contracts to automate the functionality of the defined events. We integrated our proposed solution with different systems and technologies, such as IPFS, proxy re-encryption, trusted oracles, and reputation systems to securely fetch, store, and share patients' medical records. We presented algorithms along with their implementation and testing details. We evaluated the proposed contracts under a patient health record (PHR) environment to verify their correctness. We presented cost and security analysis to show the practicality, resiliency against attacks, and feasibility of the proposed solution. We discussed how the proposed solution can satisfy defined system requirements. We outlined several limitations of the proposed solution. The proposed solution is generic enough and can be adopted for both permissioned or permissionless blockchain networks. The implemented code of proposed smart contracts has been made publicly available on GitHub.

REFERENCES

[1] J. S. Ancker, M. Silver, and R. Kaushal, "Rapid growth in use of personal health records in new york, 2012–2013," *J. Gen. Internal Med.*, vol. 29, no. 6, pp. 850–854, Jun. 2014.

[2] *EHRs Have Made it Easy for Cardiologists to Treat Their Patients*. Accessed: Jul. 8, 2020. [Online]. Available: http://tbrcinfo.blogspot.com/2018/12/ehrs-have-made-it-easy-for.html

[3] *EMR vs EHR—What is the Difference*. Accessed: Jul. 12, 2020. [Online]. Available: https://www.healthit.gov/buzz-blog/electronic-health-and-medical-record s/emr-vs-ehr-difference

[4] *EHR vs EMR: A Comprehensive Comparison of the Difference Between Them*. Accessed: Jul. 7, 2020. [Online]. Available: https://www.selecthub.com/medical-software/the-difference-between-ehr-v s-emr/

[5] (2018). *Health Records—Apple*. Accessed: Mar. 16, 2020]. [Online]. Available: https://www.apple.com/healthcare/health-records/

[6] A. Bhardwaj, S. B. H. Shah, A. Shankar, M. Alazab, M. Kumar, and T. R. Gadekallu, "Penetration testing framework for smart contract blockchain," *Peer-to-Peer Netw. Appl.*, pp. 1–16, Sep. 2020, doi: 10.1007/s12083-020-00991-6.

[7] Y. Sharma and B. Balamurugan, "Preserving the privacy of electronic health records using blockchain," *Procedia Comput. Sci.*, vol. 173, pp. 171–180, Jan. 2020.

[8] M. Qazi, D. Kulkarni, and M. Nagori, "Proof of authenticity-based electronic medical records storage on blockchain," in *Smart Trends in Computing and Communications*. Singapore: Springer, 2019, pp. 297–306.

[9] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, Jun. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804519300864

[10] D. V. Dimitrov, "Blockchain applications for healthcare data management," *Health Inf. Res.*, vol. 25, no. 1, pp. 51–56, 2019. [Online]. Available: http://www.e-sciencecentral.org/articles/?scid=1115984

[11] B. Houtan, A. S. Hafid, and D. Makrakis, "A survey on blockchain-based self-sovereign patient identity in healthcare," *IEEE Access*, vol. 8, pp. 90478–90494, 2020.

[12] L. J. Kish and E. J. Topol, "Unpatients—Why patients should own their medical data," *Nature Biotechnol.*, vol. 33, no. 9, p. 921, 2015.

[13] Y. B. Perez. (2015). *Medical Records Project Wins Top Prize at Blockchain Hackathon*. Accessed: Mar. 15, 2020. [Online]. Available: https://www.coindesk.com/medvault-wins-e5000-at-deloitte-sponsored-bloc kchain-hackathon

[14] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 224–230, Jan. 2018.

[15] J. M. Roman-Belmonte, H. De la Corte-Rodriguez, and E. C. Rodriguez-Merchan, "How blockchain technology can change medicine," *Postgraduate Med.*, vol. 130, no. 4, pp. 420–427, May 2018.

[16] X. Huang, "Blockchain in healthcare: A patient-centered model," *Biomed. J. Sci. Tech. Res.*, vol. 20, no. 3, p. 15017, Aug. 2019.

[17] (2018). *Healthbank Creates the First Patient-Centric Healthcare Trust Ecosystem*. Accessed: Mar. 4, 2020. [Online]. Available: https://www.healthbank.coop/2018/10/30/healthbank-creates-the-first-pat ient-centric-healthcare-trust-ecosystem/

[18] (2015). *HealthNautica + Factom Announce Partnership*. Accessed: Mar. 16, 2020. [Online]. Available: https://www.factom.com/company-updates/healthnautica-factom-announce-pa rtnership/

[19] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 180–184.

[20] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, Aug. 2016, pp. 25–30.

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

[22] V. Patel, "A framework for secure and decentralized sharing of medical imaging data via blockchain consensus," *Health Informat. J.*, vol. 25, no. 4, pp. 1398–1411, Dec. 2019.

[23] M. Du, Q. Chen, J. Chen, and X. Ma, "An optimized consortium blockchain for medical information sharing," *IEEE Trans. Eng. Manag.*, early access, Feb. 3, 2020, doi: 10.1109/TEM.2020.2966832.

[24] (2017). *Iryo: Global Participatory Healthcare Ecosystem*. Accessed: Apr. 21, 2020. [Online]. Available: https://iryo.network/iryo_whitepaper.pdf

[25] M. Egorov, M. Wilkison, and D. Nunez, "NuCypher KMS: Decentralized key management system," 2017, *arXiv:1707.06140*. [Online]. Available: http://arxiv.org/abs/1707.06140

[26] D. Tith, J.-S. Lee, H. Suzuki, W. Wijesundara, N. Taira, T. Obi, and N. Ohyama, "Application of blockchain to maintaining patient records in electronic health record for enhanced privacy, scalability, and availability," *Healthcare Informat. Res.*, vol. 26, no. 1, pp. 3–12, 2020.

[27] C. Chinchilla. (2019). *A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Mar. 23, 2020. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper/

[28] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[29] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, Feb. 2006.

[30] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2007, pp. 288–306.

[31] S. S. Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient unidirectional proxy re-encryption," in *Proc. Int. Conf. Cryptol. Afr.*, Stellenbosch, South Africa: Springer, 2010, pp. 316–332.

[32] *Ethereum alarm clock*. Accessed: Jul. 25, 2020. [Online]. Available: https://www.ethereum-alarm-clock.com/

[33] A. Garai, "Empirical and practical implementation methodology for clinical integration of E-Health iot technology," *Int. J. Med. Health Sci. Res.*, vol. 3, no. 12, pp. 117–125, 2016.

[34] R. Xu, S. Chen, L. Yang, Y. Chen, and G. Chen, "Decentralized autonomous imaging data processing using blockchain," *Proc. SPIE*, vol. 10871, Feb. 2019, Art. no. 108710U.

[35] C. Cai, X. Yuan, and C. Wang, "Towards trustworthy and private keyword search in encrypted decentralized storage," in *IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.

[36] S. Pariselvam and M. Swarnamukhi, "Encrypted cloud based personal health record management using Des.scheme," in *Proc. IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Mar. 2019, pp. 1–6.

[37] C.-J. Wang, X.-L. Xu, D.-Y. Shi, and W.-L. Lin, "An efficient cloud-based personal health records system using attribute-based encryption and anonymous multi-receiver identity-based encryption," in *Proc. 9th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, Nov. 2014, pp. 74–81.

MOHAMMAD MOUSSA MADINE (Member, IEEE) received the B.Sc. degree in computer engineering from Khalifa University, Abu Dhabi, UAE, in 2019. He is currently a Graduate Researcher and a Teaching Assistant pursuing his graduate studies with Khalifa University. His research interests include blockchain solutions in healthcare, personal health records, and edge computing.

AMMAR AYMAN BATTAH received the B.Sc. degree in computer engineering from Khalifa University, Abu Dhabi, UAE, in 2019. He is currently a Researcher and a Teaching Assistant pursuing his graduate studies in computer science with Khalifa University. His current research interests include blockchain technologies, the Internet of Things (IoT) security, and information security.

IBRAR YAQOOB (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He worked as a Researcher and a Developer with the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. He is currently working with the Department of Electrical Engineering and Computer Science, Khalifa University, UAE. Previously, he worked as a Research Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, where he completed his postdoctoral fellowship under the prestigious grant of Brain Korea 21st Century Plus. His numerous research articles are very famous and among the most downloaded in top journals. He has been listed among top researchers by Thomson Reuters (Web of Science) based on the number of citations earned in the last three years in six categories of Computer Science. He has been involved in a number of conferences and workshops in various capacities. His research interests include big data, blockchain, edge computing, mobile cloud computing, the Internet of Things, healthcare, and computer networks. He is also serving/has served as a guest/associate editor in various journals.

KHALED SALAH (Senior Member, IEEE) received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, USA, in 1990, and the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, USA, in 1994 and 2000, respectively. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Khalifa University, UAE. He has over 220 publications and three U.S. patents, has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of Blockchain, the IoT, Fog and Cloud Computing, and Cybersecurity. He has served as the Chair of the Track Chair for IEEE Globecom 2018 on Cloud Computing. He is also an Associate Editor of IEEE Blockchain Technical Briefs, and a member of IEEE Blockchain Education Commitee. He is also leading a number of projects on how to leverage blockchain for Healthacare, 5G Networks, Combating Deepfake Videos, Supply Chain Management, and AI.

RAJA JAYARAMAN received the bachelor's and master's degrees in mathematics from India, the M.Sc. degree in industrial engineering from New Mexico State University, and the Ph.D. degree in industrial engineering from Texas Tech University. He is currently an Associate Professor with the Department of Industrial & Systems Engineering, Khalifa University, Abu Dhabi, UAE. His expertise is in multi-criteria optimization techniques applied to diverse applications, including supply chain and logistics, healthcare, energy, environment, and sustainability. His research interests include blockchain technology, systems engineering and process optimization techniques to characterize, model and analyze complex systems with applications to supply chains, maintenance operations planning, and healthcare delivery. His postdoctoral research was centered on technology adoption and implementation of innovative practices in the healthcare supply chains and service delivery. He has led several successful research projects and pilot implementations in the area of supply chain data standards adoption in the US healthcare system. His research has appeared in top-rated journals including: *Annals of Operations Research*, *IISE Transactions*, *Energy Policy*, *Applied Energy*, *Knowledge Based Systems*, IEEE Access, *Journal of Theoretical Biology*, *Engineering Management Journal*, and others.

YOUSOF AL-HAMMADI received the bachelor's degree in computer engineering from the Khalifa University of Science and Technology (previously known as Etisalat College of Engineering), Abu Dhabi, UAE, in 2000, the M.Sc. degree in telecommunications engineering from the University of Melbourne, Australia, in 2003, and the Ph.D. degree in computer science and information technology from the University of Nottingham, U.K., in 2009. He is currently an Acting Dean of Graduate Studies and an Assistant Professor with the Department of Electrical & Computer Engineering, Khalifa University of Science and Technology. His research interests include the area of information security which include intrusion detection, botnet/bots detection, viruses/worms detection, machine learning and artificial intelligence, RFID security, and mobile security.

SASA PESIC was born in December, in 1992. He is currently pursuing the Ph.D. degree with the Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Serbia. He is also working as a Blockchain Engineer with VizLore Labs Foundation, Novi Sad, and with VizLore LLC, Scottsdale, AZ, USA. He is also a Visiting Researcher with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA, and with the Blockchain Research Laboratory, Arizona State University. In his research work, he deals with highly distributed Internet of Things and edge computing systems, analyzing their robustness, security, and operating capacity and stability. In addition, as part of his doctoral thesis, he is designing, modeling, and implementing an advanced indoor positioning system called BLEMAT. Finally, his research interests include distributed ledger technologies and their interdisciplinary application in the domains of energy, finance, security of IoT systems, and peer-to-peer insurance. He is the author/coauthor of nine conference articles and one article in an international journal. As a Research and Development Engineer, he is actively working on two Horizon2020 research projects: PhasmaFOOD, Interconnect, and in the past two years he has worked on Vicinity, AgileIoT, and i SymbIote.

SAMER ELLAHHAM received the bachelor's degree in biology and the M.D. degree from the American University of Beirut, Beirut, Lebanon.

He is currently a Cleveland Clinic Caregiver, Cleveland, OH, USA, seconded as a Senior Cardiovascular Consultant and the Director of Accreditation in the Quality and Safety Institute, Cleveland Clinic Abu Dhabi. He is also the Middle East Regional Chair, the Patient Safety Movement Foundation, the ISQua Expert, the AHA Hospital Accreditation Science Committee, a member, the European Society of Cardiology Heart Failure Writing Group, a member, and an ex-Middle East Representative of the JCI Standards Subcommittee and American College of Cardiology Accreditation Foundation Board, Member. He finished his internal medicine residency at Georgetown University Hospital–Washington Hospital Center and his fellowship in Cardiology at the Virginia Commonwealth University Health System in the USA. He worked in Washington, DC, USA at the Georgetown University Hospital–Washington Hospital Center and in several clinical and leadership positions before moving to UAE in 2008. He continues to be an active clinician. He demonstrates great skill and experience in the management of patients with heart failure, ischemic heart disease, and valvular heart disease and led a Multi-Disciplinary Team in the care and delivery of advanced therapies to these patients. He has unique abilities to partner and engages local and regional referring providers. He can work in a highly matrixed environment, possess strong leadership and organizational skills, and have the experience of working effectively in a large health system. He led the First AHA GWTG Heart Failure Initiative outside the US and was a recipient of the AHA GWTG Award in Washington. He is also the Champion of the AHA GWTG in the region. He has served as a Chief Quality Officer for SKMC from 2009 till 2017. In his role, he has led the development of a quality and program that has been successful and visible and has been recognized internationally by several awards. As a Chief Quality Officer and a Global Healthcare Leader, he had a focus on ensuring that the implementation of these best practices leads to breakthrough improvements in clinical quality, patient safety, patient experience, and risk management. He was the Executive SKMC sponsor of the American College of Surgeons National Surgical Quality Improvement Program (ACS NSQIP®) the leading US validated, risk-adjusted, and outcomes-based program to measure and improve the quality of surgical care. SKMC is the first multispecialty ACS NSQIP center outside the U.S. He led the publication of, first in the region, annual SKMC outcome books, since 2011. He is also a strong believer in transparency in health care and external reporting. He was the Leader of the First Pilot International Robust Process Improvement (RPI) project by the Joint Commission Center for Transforming Healthcare and several other similar successful performance improvement projects at SKMC. He is also the American Board Certified in Internal Medicine, Cardiovascular Disease, Vascular Medicine, and American Board of Medical Quality. He was recently recertified in 2017 by the American Board of Cardiology (ABIM). He is also a Certified Professional in Healthcare Quality (CPHQ) by The National Association for Healthcare Quality (NAHQ), Certified in Medical Quality (CMQ) by The American Board of Medical Quality (ABMQ), and Certified as the EFQM Model assessor and the Lead Trainer in TeamSTEPPS. He is also a Fellow of the American College of Cardiology, the American Heart Association, the American College of Chest Physicians, the American College of Physicians, the American College of Medical Quality, and the American College of Cardiology, and a Key Member of Heart Failure and Transplant, Adult Congenital and Pediatric Cardiology, Cardio-oncology, Innovation, Quality, and Peripheral Vascular Disease Sections. He is also a Distinguished Fellow of the New Westminster College in British Columbia, Canada, and an Advisory Board Member, the University of Wollongong in Dubai. He was the Middle East Representative of the JCI Standards Subcommittee and a member on the Editorial Advisory Board of the *Joint Commission Journal on Quality and Patient Safety*. He was a Reviewer of HCAC Cardiac Quality and Safety Standards. He has been a Champion and the Leader of the use of Lean, Six Sigma, and Change Management to improve healthcare quality and has numerous publications in this area. He is also a Lean Six Sigma Master Black Belt Certified. He is also an American Society of Quality (ASQ) trainer in Lean and Six Sigma both green and black belt. He is also the ISQua Expert. He is also a Recognized Innovative Leader in quality, safety, patient experience, artificial intelligence, blockchain, telehealth, clinical cardiology, and the use of robust performance improvement in improving healthcare delivery. He also serves on several U.S. and international prestigious committees and advisory bodies. He is also the Middle East Regional Chair of the Patient Safety Movement Foundation. He also received several research awards, including the DuPont Pharmaceuticals Research Award, ACCP 58th Annual Scientific Assembly, Young Investigator Award; the Alfred Soffer Research Award, ACCP 58th Annual Scientific Assembly, Finalist; the First Young Investigator Award 12th, Annual Meeting of the Mediterranean Association of Cardiology and Cardiac Surgery, American Heart Association Get with the Guidelines Award, SKMC Infection Prevention Award in 2011 and 2012, Sheikh Khalifa Excellence Award in 2014, Quality Leadership Award from the World Quality Congress and Awards, Business Leadership Excellence Award from World Leadership Congress in 2015, one the nominees for Safe Care magazine Person of the Year in the United States, Dubai Quality Award in 2015, and Sheikh Khalifa Excellence Golden Award in 2015. He is also an Avid Researcher; his research interests include heart failure, acute coronary syndromes, frailty, dyslipidemia, accreditation, second victim phenomenon, resilience, innovation, artificial intelligence, telehealth, blockchain, patient flow, patient experience and engagement, lean-six sigma, patient safety, bowtie risk management tool, and KPI management. He is also a Recognized World-Leader in these fields. He is also the Eminent Editor of the *Journal of Cardiology & Cardiovascular Therapy* and an Associate Editor of the *American Journal of Medical Quality*. In addition, he also serves on the editorial board of *Journal of Thoracic Disease and Cardiothoracic Surgery*, *Developments in Clinical & Medical Pathology (DCMP)*, *The Joint Commission Journal on Quality and Patient Safety*, *Telehealth and Medicine Today (TMT)*, *Blockchain* journal (BHTY), *Medical Science*, *Open Journal of Cardiatic Research*, *UPI Journal of Pharmaceutical, Medical and Health Sciences (UPI-JPMHS)*, *Open Access Research in Anatomy*, *Gerontology & Geriatrics studies*, *Open Access Journal of Clinical Trials*, Hypertension Today Journal and Focus on *Hypertension* Journal, *Journal of Heart Health*, *Journal of Cardiovascular Pharmacology*, *Scientific Research and Community*, *Journal of Surgery and Surgical Procedures*, *EC Cardiology*, *Journal of Cardiovascular and Pulmonary Medicine*, and *Canadian Journal of Biomedical Research*. He is also a reviewer for several peer-reviewed journals, including *Joint Commission Journal on Quality and Patient Safety*, *International Journal of Quality & Reliability Management*, the *Journal of American College of Cardiology*, the *American Heart Journal*, *Annals of Internal Medicine*, *Archives of Internal Medicine*, *Chest*, *Circulation*, *Clinical Cardiology*, *Chest*, *Lancet*, *Diabetes Care*, *Archives of Internal Medicine*, *Endocrinology and Metabolism*, *European Journal of Heart Failure*, *Congestive Heart Failure Journal*, *Journal of Nuclear Cardiology*, the *Journal of Transplant Coordination*, the *Journal of Cardiovascular Pharmacology*, the *Southern Medical Journal*, *European Journal of Innovation Management*, *The Anatolian Journal of Cardiology*, and *npj Digital Medicine*. He enjoys volunteering, tennis, healthy lifestyle, innovation, teaching, and future health.