IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 1, 2020, accepted December 12, 2020, date of publication December 15, 2020,
date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3045048

Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records

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

[1]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi 127788, United Arab Emirates
[2]Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi 127788, United Arab Emirates
[3]Heart and Vascular Institute, Cleveland Clinic Abu Dhabi, Abu Dhabi, United Arab Emirates
[4]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA

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 Patients are becoming aware of the importance of taking secure control and managing access over their medical data, thereby leading to the rise in the adoption of personal health record (PHR) systems. However, today's PHR systems fall short in providing secure and trustable data sharing and access facilities to patients when they are in emergency situations or temporarily incapacitated. Also, the existing PHR systems are centralized and vulnerable to the single point of failure problem. Integrating PHR systems with blockchain technology can help to overcome such limitations. In this paper, we propose a blockchain-based PHR architecture that employs smart contracts to implement multi-party authorization (MPA) and threshold cryptographic schemes to automate secure and trustable medical data sharing and access in PHR systems. Moreover, we mitigate the limited storage and computation capabilities of blockchain by using InterPlanetary File System (IPFS) storage and reputation-governed trusted oracles into the proposed architecture. MPA and threshold cryptographic schemes allow the patient to split and share a secret key with a set of trusted parties, such as the healthcare regulatory agency, guardians, and hospitals, in such a way that they can collectively decide on sharing medical data on behalf of patients. We present algorithms along with their full smart contract function implementation details. We evaluate the robustness and performance of our solution by performing correctness verification and cost analysis. Furthermore, we evaluate the proposed approach in terms of security, generalization, and limitation aspects to find out its feasibility and practicality. We make our smart contract code publicly available on GitHub.

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

I. INTRODUCTION

A personal health record (PHR) is the set of a patient's medical data, collected from multiple medical institutions (MIs), consumer health devices, and patient-gathered medical data (PGHD). Considering how the engagement of patients with their medical data leads to more positive healthcare experience, more patients are becoming interested in taking control over their medical data [1], [2]. For a PHR system to be viable, it needs to be patient-centered, which requires that the contents of the health records are never accessed, viewed, or modified by any entity other than the patient, unless given explicit access rights that are secure, traceable, and auditable. Several enterprise solutions, such as Microsoft HealthVault, Google Health, and Apple Health, began to surface and gain popularity as PHR platforms [3]. However, due to their centralized nature, such solutions cannot be trusted to safely store patient data or to never use it without user consent. Furthermore, in the light of increasing security attacks, centralized medical data services are becoming more susceptible to security attacks and data hacks [4].

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

Blockchain is a decentralized technology that can be used as the foundation of a trustful and immutable system. The potential of blockchain and smart contracts have been explored and tested in various domains in terms of delivery assurance of digital and physical assets, incentive management for self-organizing networks, healthcare data management, and soybean traceability in the agricultural supply chain [5]–[9]. Utilizing a blockchain-based architecture in a PHR system, provides transparency, security, and provenance, and auditable features. Choosing the optimal blockchain architecture type, such as public, private, or consortium, depends on the requirements and goals to be met. In our previous work [10], we proposed a blockchain-based patient-centered PHR management system. Our approach integrated several other protocols and systems, such as Inter-Planetary File System (IPFS), reputation-governed trusted oracles (RGTO), and proxy re-encryption (PRE) into the blockchain. IPFS is a peer-to-peer decentralized storage platform that provides the means to indirectly store large data off-chain and mitigating the storage drawbacks of blockchain. RGTO is a set of public computation nodes that compete to process a task, with a reputation system that punishes malicious actions and rewards virtue. The trusted oracles are utilized to execute resource-intensive tasks, which blockchain is too slow to perform. For example, fetching files from IPFS and cryptography operations, such as PRE cannot be done efficiently on-chain. PRE is a mechanism that atomically - with no middle steps - transforms the encryption state of a certain file from one key to another, provided a re-encryption key created by the original file owner.

A practical PHR management system must give the patient the choice to give partial or full access rights to other entities, such as doctors, nurses, relatives, or institutions. Moreover, the system should be able to decide whether or not to automatically authorize future access rights to entities chosen by the patient. For example, the patient can set their PHR such that in a case of emergency, if two relatives and a doctor request access to the medical documents, their requests will be automatically granted. In addition to that, the health records of an incapacitated patient, who could be in that medical state before or after creating their PHR profile, should be managed with utmost transparency and privacy, in such a way that ultimately keeps the full control rights with the patient. In this paper, we develop a PHR architecture that can efficiently deal with urgent emergencies. Note that our proposed solution works under the same rules for all types of patients and all kinds of use case scenarios.

A. RELATED WORK

A patient-centered dynamic access control scheme for PHRs has been proposed in [11]. The proposed scheme, based on individual authorizations to entities and public-key cryptography management, help to preserve the privacy of the patient PHR while allowing multiple users to get full access to the data. However, the solution cannot provide partial access to some of the patient records, in addition to being dependent on cloud services for storing the PHR data. The cloud dependency makes the solution architecture centralized, and therefore it is vulnerable to security attacks, such as the denial of service (DoS), thereby making the entire system unavailable.

In [12], the authors have proposed a blockchain-based framework to enable secure health data sharing between different healthcare providers. Another study conducted in [13] proposed a blockchain-based system named ''MeDShare'' to address the problem of medical data sharing in terms of data provenance and auditing. The MeDShare keeps the track record of all the entities along with their actions. All the actions performed on the MeDshare are recorded in a tamper-proof manner. To deal with the privacy issues that exist in today's healthcare data storage and sharing systems, a decentralized and permissioned blockchain-based solution has been proposed in [14]. The proposed solution ensures privacy by enabling a separate channel communication scheme. It also enhances identity management using the membership service offered by the permissioned blockchain.

The study conducted in [15] proposed a more advanced attribute-based access control (ABAC) scheme that uses identity-related policies set by the users to securely share the electronic health records (EHR). The approach uses attribute authorities to validate user attributes without requiring the users to have prior registration in the network. The attribute authorities are trusted public entities, such as hospitals, which process the attribute verification on a centralized server, exposing the system to network security risks. Moreover, access control decisions based on user attributes are limited and do not allow time-based control of patient data sharing. On the other hand, another attribute-based approach presented by Li *et al.* [16] uses encryption to enforce the access control rules, in addition to providing multiple one-way privilege levels. The leveled privileges allow cascading the access rights, such that a higher-privileged entity can pass down partial access rights to a lower-privileged entity. As such, the patient can give their primary doctor full access rights to certain medical data, and the doctor can give partial rights to the nurse or another doctor. This approach takes away the patient-centered aspect of the healthcare system because the patient data access rights can be cascaded without patient consent. Moreover, the paper does not discuss a solution to emergency cases and incapacitated patients, where the patient data must be secured and shared with the smallest number of entities.

Another cloud-based access control scheme for EHRs was developed in [17]. The scheme uses ABAC in conjunction with extensible access control markup language (XACML) to offer a fine-grained and privacy-preserving EHR sharing. However, the design is centered around the requester (such as the doctor), and not the patient, which in patient-centered healthcare systems is supposed to own the data and have full control over it.

In [18], the authors introduced a blockchain-based solution for managing EHR data sharing. The study proposes a granular access authorization with support for flexible queries.

Unlike previous cloud-oriented solutions, this approach does not mention the types of possible attributes required for decision making. In the same vein, the authors in [19] proposed a framework for secure interoperable and efficient access to health records while preserving patient privacy. However, the solution proposed is not entirely patient-centered, as there are dependencies on the hospital.

More recent research looked into building PHR management systems that have blockchain as their foundation rather than using it as an auxiliary technology. The authors in [20] designed a tamper-resistant and secure PHR system with support for granting and revoking access rights. In [21], the authors have leveraged ABAC to provide support for dynamic attributes in a blockchain-based EHR solution. Even though both solutions primarily use blockchain, they are not fully decentralized because of their partial use of cloud services or hospitals for storing public keys of entities and patient data. Furthermore, these solutions are not patient-centered, since the doctors have the privilege to override patient decisions.

EACMS is an emergency access control management system for blockchain-based PHRs, proposed by Rajput et al. [22]. This system is developed for emergency cases. However, the study assumes that PHRs are stored in a cloud server that hinders its use in other architectures that store PHRs in a decentralized manner. On top of that, the solution does not have a mechanism for preventing abuse of emergency access.

An approach proposed by Battah *et al.* [23] showcases a blockchain-based architecture for multi-party authorization (MPA) and access control for encrypted data that is stored over distributed storage systems, such as IPFS. The design proposed in the paper is generic, which can be adapted into a variety of use cases; however, more refinement of the architecture is required to make it most suitable for the PHR management systems.

In summary, the existing healthcare solutions do not meet the necessary requirements for the PHR systems in terms of traceability, resiliency against security attacks, and delegating decision making to their trusted guardians in case of emergencies. In this paper, we propose a fully decentralized multi-party consent management system for accessing encrypted PHR medical documents and delegating access permission rights to trusted entities. Our solution incorporates various technologies to ease off the weaknesses in current PHR systems. Note that our study is significantly different from our previous work [10] as it aims to solve completely different challenges and proposes a novel solution that is particularly designed using new techniques and flow of interactions among the entities. First, this paper defines a simple system structure that merges the patient and doctor into a single entity type, tightly integrates the reputation system with the oracles, and deploys a single one-time smart contract that keeps track of all interactions. On the other hand, in [10], the stakeholders are broken down without considering situations where the doctor entity could also be a patient in a different scenario, the oracles operate and interact with functions different than the ones that evaluate them, and the smart contracts are neither unified into a universal version nor broken down further such that each entity deploys its own. Second, all entities in this solution, except RGTOs, have their identities verified by the governing body that deploys the smart contract, therefore preventing cases of identity theft or repeated registrations, whereas [10] does not deal with such actions. Third, patients must have a secure and private mechanism to delegate the decision making regarding the sharing of medical documents to other trusted entities; however, the architecture proposed in [10] assumes the patients to be always available and legally able to grant or deny access permissions of the medical documents to doctors. Our key contributions are as follows:

1) We propose a fully decentralized blockchain-based multi-party consent management for patient-centered PHR systems to provide provenance of access log events in an immutable, auditable, trustful, and secure manner.

2) We develop smart contracts to implement MPA and threshold cryptographic schemes to automate secure and trustable medical data sharing and access in PHR systems even in cases of emergencies.

3) We integrate decentralized IPFS storage and trusted oracles to perform re-encryption into the proposed PHR architecture.

4) We introduce a reputation-system layer to govern the entities that perform their processing off-chain, including the re-encryption oracles, doctors, and the regulatory agency that deploys the system.

5) We develop algorithms that translate our proposed architecture and write smart contract functions and trigger events to implement the algorithm. The implementation code is made publicly available.[1]

6) We analyze the cost and security aspects of our solution and verify the correctness of our implementation, to know the limitations in a realistic deployment environment.

The remainder of the paper is organized as follows. In section II, we present the proposed approach by explaining the different types of entities and smart contracts 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

This section presents our proposed blockchain-based solution along with its full system component details, including Ethereum smart contracts, threshold cryptography, MPA, IPFS, RGTO, and PRE.

A. ETHEREUM AND SMART CONTRACTS

Ethereum is a public and open-source blockchain platform that provides a developer-friendly infrastructure for solutions that require full decentralization. Ethereum smart contracts are written in Solidity language. The smart contract code is

[1] https://github.com/anon092020/PHR-MPA

executed by Ethereum virtual machines (EVMs), which are an execution environment that exists inside mining peer-to-peer nodes. The mining nodes store mined Ethereum transactions in a chained sequence of blocks, in addition to mining new transactions by executing their smart contracts and reaching consensus using the Ethash proof-of-work (PoW) algorithm. In PoW, distributed nodes compete to solve a computationally-intensive problem, in exchange for Ether, which is the Ethereum cryptocurrency. The amount of Ether the winning nodes receive is correlated with the complexity of the smart contract code, which is measured in gas units [24], [25]. The average gas price is measured typically in Gwei units, where 1 wei is 10^{-18} Ether.

Although our solution keeps all patient data secure using public-key infrastructure (PKI) cryptography, certain sensitive files cannot be shared on the public Ethereum network due to privacy issues. In such cases, there is a need to use forks of Ethereum that are permissioned and private, such as Quorum and Hyperledger Besu [26], [27].

B. THRESHOLD CRYPTOGRAPHY AND MULTI-PARTY AUTHORIZATION

Threshold cryptographic schemes are generally encryption and decryption protocols where more than one party can contribute to the operation [28], hence named as MPA. Threshold cryptography can be highly advantageous in decentralized systems that require the approval of m out of n entities for the cryptographic operation to succeed, where $m \leq n$, n is the total number of entities involved in the encryption operation and m is the required number of entities for the decryption operation. The MPA comprises all the parties that generate, protect, or share the secret data. In our proposed architecture, these parties are the patient, the guardians (who could be the patient's relatives or trusted people), the hospital, and the regulatory agency.

C. InterPlanetary FILE SYSTEM

IPFS is a distributed storage solution that relies on public peer-to-peer nodes to maintain and share files. IPFS possesses unique features, such as content-based addressing which makes the SHA-256 cryptographic hash of a certain file its identifier, InterPlanetary Name System (IPNS) for maintaining the same path for updated files, and version control system (VCS) as a protocol for file editing and updating. IPFS can help to overcome the storage limitations posed by blockchain systems by enabling them to store the hash of the file on-chain and using it as a pointer to the file [29].

D. REPUTATION-GOVERNED TRUSTED ORACLES

Oracles are used to make computation requests in exchange for monetary incentives. They are used to make up for the slow, limited, and expensive computation of EVMs; however, passing on all computations to a single oracle no longer makes the overall system architecture decentralized and trustful. In the proposed solution, we use proxy re-encryption servers/oracles to perform compute-intensive tasks because implementing them using the Ethereum-based smart contract can be very expensive [23].

Reputation-governed trusted oracles (RGTO) are constantly incentivized to act truthfully and quickly in two forms, the first is fueled by competition among the oracles to return the first and most accurate response to get a larger monetary payment, and the second relies on a reputation system that may potentially block untruthful oracles from receiving any request. In order to use oracles without forgoing the desired characteristics of blockchain, our smart contracts send the computation request to a large number of oracles and collect a sufficient number of results. Depending on the request, if it is possible to verify the result on-chain, the smart contract will accept the response of the quickest oracle, otherwise, the accepted responses are judged based on the majority or average of the results. A reputation system is a technique that helps to distinguish the truthful and untruthful oracles. Smart contracts determine what oracles to communicate with based on the reputation score of those oracles. Therefore, oracles are encouraged to maintain a high reputation score by returning accurate results as quickly as possible.

E. PROXY RE-ENCRYPTION

PRE is a cryptosystem scheme that enables the direct transformation of an encrypted file from one key to another without decrypting the file or sharing the private keys [30]. A simple example of using PRE between Alice and Bob is described in the following steps:

1) Encryption: Alice encrypts a secret message with her public key.

2) Re-encryption key generation: Alice uses her private key and Bob's public key to generate a re-encryption key and sends it to the PRE server.

3) Re-encryption: The PRE server transforms the secret message from Alice's public key to Bob's public key and sends the message to Bob.

4) Decryption: Bob decrypts the message using his private key, revealing the plaintext message.

In our solution PRE is used primarily to share patient medical documents with doctors, although not in a direct way as in the example above, because the medical document may have to be shared in emergency cases where the patient is not available to generate the re-encryption key.

F. OVERALL SYSTEM ARCHITECTURE

Our proposed system architecture is summarized in Figure 1 that shows all the entities that interact with each other in order to share a patient's medical document with a certain requesting doctor. Our solution requires all entities involved to be registered on the Ethereum blockchain network. Therefore, all of them have a unique Ethereum address (EA) and a private-public Ethereum key pair.

- **Regulatory agency**: A trusted governing body that could be the government or the department of health in the country. This entity deploys the main smart contract and is

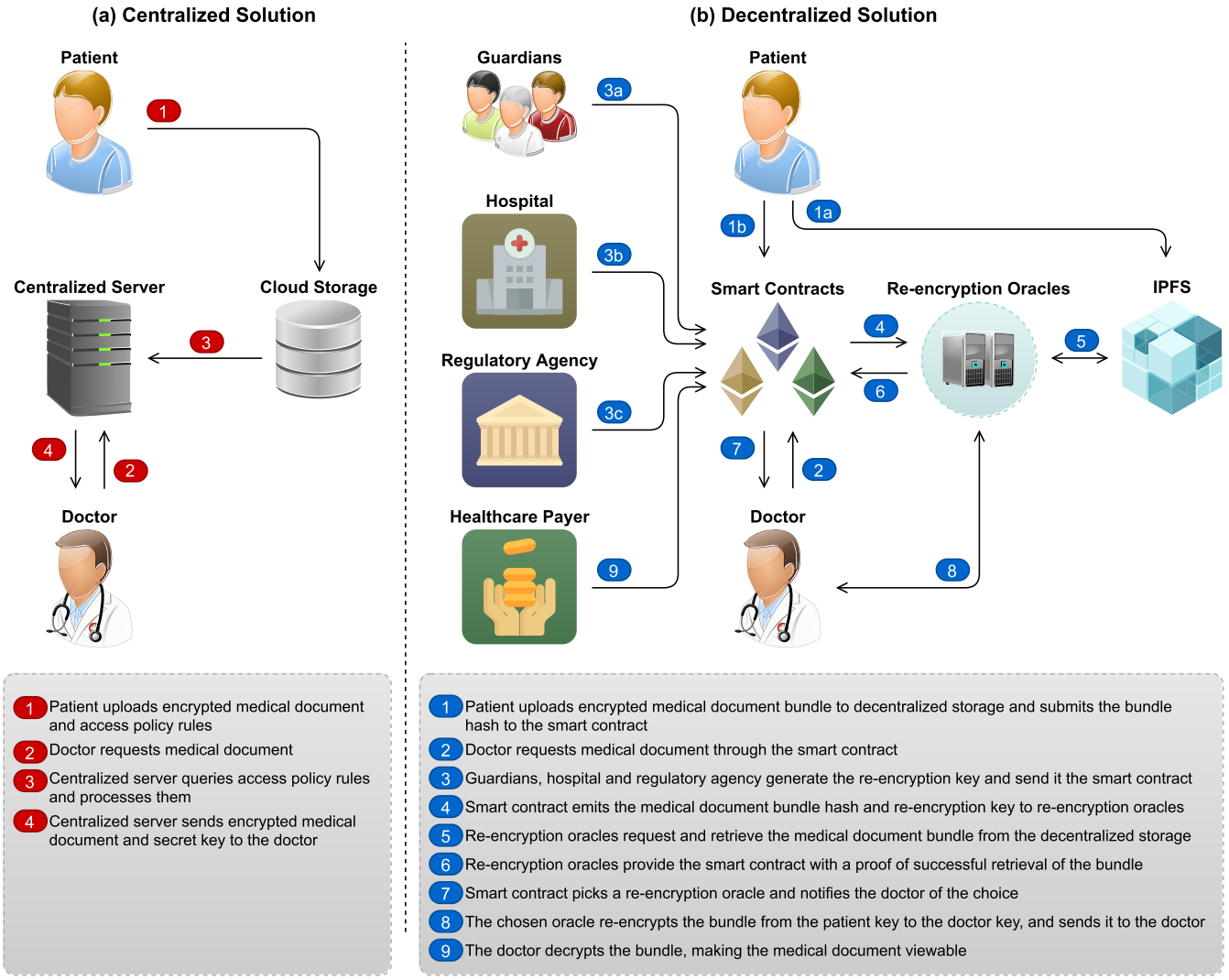

FIGURE 1. An overview of (a) current centralized solutions, and (b) our proposed decentralized solution.

part of the MPA that approves the sharing of absent patient medical documents. The regulatory agency is also responsible for verifying the identities of any person or entity that registers into the network, whether that is the patient, guardian, doctor, hospital, oracle, or healthcare payer. The regulatory agency is ultimately managed by its employees through the decentralized application (DApp).

- **Person**: A general type of entity that by default is registered either as a patient or a guardian, depending on whether the entity is managed by the patient personally or by a trusted guardian or guardians. A patient can issue claims to become a guardian of other patients. Becoming a guardian enables the patient to be part of the MPA to approve the sharing of medical documents in cases where the patient is absent. Furthermore, patients can issue claims to registered as doctors, granting them permission to request patient medical documents. All claims issued by the patient must be verified by the MPA, more specifically, guardian claim requests are verified by receiving patient and regulatory agency approvals, whereas doctor claim requests are verified by receiving hospital and regulatory agency approvals. In addition to the Ethereum private-public key pair each person possesses, patients must have an IPFS key pair, with the private key is split and shared 50% (3 shares) with the regulatory agency and the remaining 50% (3 shares) are kept secret with the patient. Moreover, the patient will produce a unique key pair for each medical document uploaded to IPFS. The person handling this entity manages the PHR DApp either directly on a personal device or through a trusted third-party (TTP) service.
- **Hospital**: In addition to being the source of medical documents for patients, the primary responsibility of the hospital in this MPA scheme is to validate a person's claim of being a doctor, and to confirm the doctor's claim that a patient has an emergency admission to the hospital.
- **Re-encryption RGTO**: An RGTO node that fetches IPFS files, performs PRE process, and acts as an Ethereum Alarm Clock (EAC) for timeout functionality in Solidity [31]. The nodes race to perform PRE to transform the medical

document encryption from the patient to the doctor, after which, the winning node communicates with the doctor directly to transfer the re-encrypted medical document to the latter's local device.

- **Insurance Company**: Responsible for paying the decentralized storage and oracle nodes.

The regulatory agency deploys a universal **regulatory agency smart contract (RASC)** that manages all entities, and provides patients, guardians, and doctors with the ability to send transactions. Moreover, RASC performs reputation evaluation and maintenance of oracle nodes.

G. INTERACTIONS AND MESSAGE SEQUENCE

Figure 2 depicts a typical sequence of interactions among the entities involved in uploading and sharing a medical document. The following actions assume all entities involved are registered and their registration is verified by the regulatory agency.

1) The patient generates a symmetric key, and encrypts the medical document using the key.

2) The patient generates medical document private-public key pair and encrypts the symmetric key using the medical document public key.

3) The patient generates an IPFS private-public key pair (only for the first medical document, will be used for all future medical documents), and encrypts the medical document private key using the IPFS public key.

4) The patient uses a threshold signature to split the IPFS private key into 6 shares (3 kept on local devices and 3 securely shared with the regulatory agency).

5) The patient uploads a bundle that consists of the encrypted medical document, the encrypted symmetric key, the encrypted medical document key, and a pseudo-random number to IPFS. Then the SHA-256 hash of the bundle is submitted to RASC.

6) The doctor requests the medical document, and optionally specifies whether the request is for an incapacitated patient or an emergency case.

7) The RASC informs the patient of a new request. If the request was for an absent patient the appropriate MPA will be informed as well. The MPA for an incapacitated patient requires the patient to be registered as such, validation of doctor credentials, and the approval of $\min(\lceil 0.7g \rceil, 5)$ guardians, where g is the total number of verified guardians. The MPA for an emergency case requires the confirmation of emergency admission from the hospital in concern and the validation of doctor credentials.

8) In case 1, the patient denies the request or the MPA requirements are not met within 1 hour, after which the RASC informs the doctor of denied access. The sequence terminates.

9) In case 2, the patient grants access then generates a re-encryption key and sends it to RASC. The sequence continues from step 11.

10) In case 3, the MPA requirements are met within 1 hour. The RASC sends a reputation token to the patient to allow rating the doctor, then the regulatory agency nodes independently use their 3 shares and decrypt the medical document private key, which is then used by one of the nodes to generate a re-encryption key and sends it to RASC. The sequence continues from step 11.

11) The RASC informs the doctor of a granted access.

12) The RASC sends the medical document bundle hash and re-encryption key to a set of RGTOs.

13) The RGTOs request and receive the medical document bundle from IPFS, then get the random number from the IPFS bundle and send it privately to RASC.

14) The RASC evaluates the RGTOs, then updates the RGTO ratings and chooses the most reputable one. Then RASC sends a token to the winner RGTO and the doctor.

15) The doctor requests the re-encrypted medical document from the RGTO, which in computes and sends it to the doctor.

16) The doctor decrypts the symmetric key using the public key, then decrypts the medical document using the symmetric key.

17) Judging from the interaction with the RGTO, the doctor submits a rating to RASC, which in return updates the RGTO reputation.

Each of the steps of generating the symmetric key, IPFS key pair, and medical document key pair play an important role in the solution and provide an improvement to the design. Using the symmetric key rather than the patient's IPFS public key to encrypt the medical documents allow encrypting and uploading the medical document files, which are commonly large in size, only once. The file that is going to be re-encrypted by the PRE-running RGTOs is the encrypted symmetric key. The secret message that is split and shared with the trusted MPA entities is the medical document private key and not the symmetric key, this is because the PRE nodes eventually need the private key used in the symmetric key encryption for a successful re-encryption, however, revealing the Ethereum private key is prohibited, and therefore we use the health record private key.

III. IMPLEMENTATION

Considering the scope of our implementation, we defined the structure of 5 types of entities, 1) regulatory agency owner (RAO), 2) regulatory agency member (RAM), 3) hospital, 4) person, 5) RGTO. Figure 3 depicts the entities along with their attributes, methods, and relationships. All these entities interact with RASC, and therefore must have an Ethereum account. On top of that, the identities of these entities must be validated through the regulatory agency, with the exception of the RGTOs. RAM identities are verified at the time of registration since they are registered by the trusted RAO with the function registerRAM. The rest of the entities must issue claims and provide appropriate proof of identity documents to RAM employees off-chain as part of the regular identification system in the region. The registration claims are set by the smart contract functions corresponding to the different entities, which are registerPatient, registerDoctor, registerHospital, and registerRGTO.

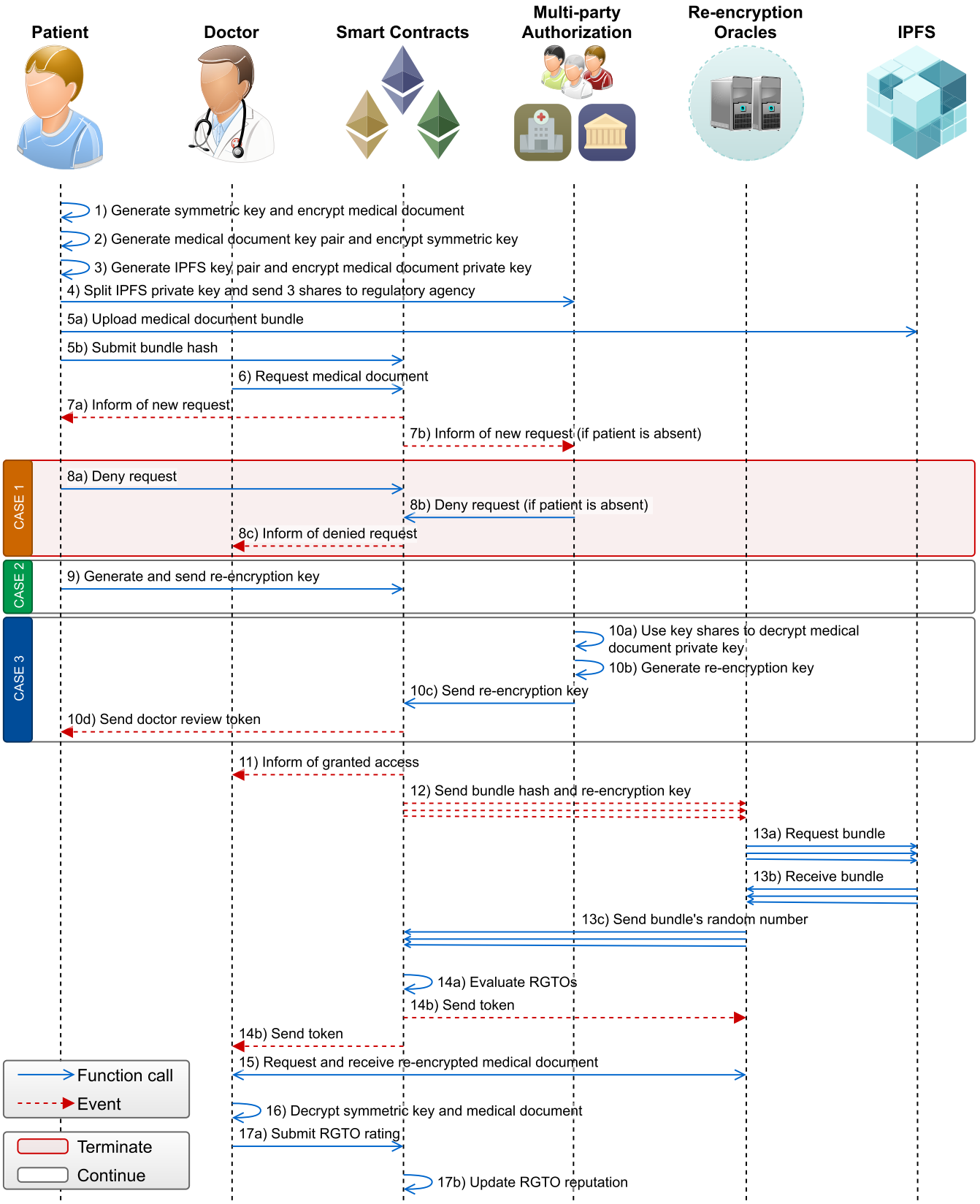

FIGURE 2. Sequence diagram of accessing health records of active or absent patients.

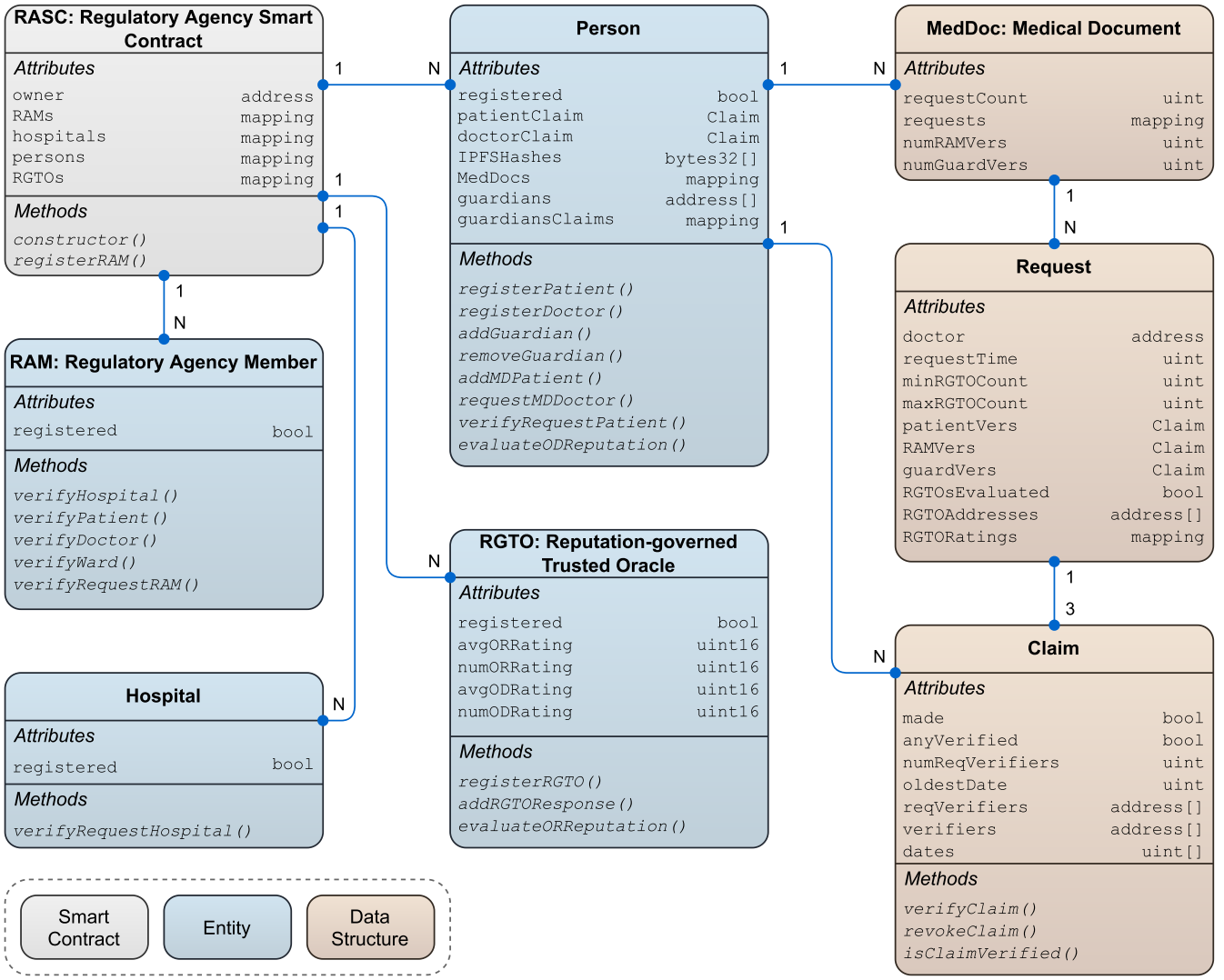

FIGURE 3. Entity relationship diagram of smart contracts, stakeholders, and data structures in our implementation.

A claim is a data structure that keeps track of the status of the claim, by storing a set of verifications and comparing them to a set of required verifications. The set of required verifications can be empty, in which case the verifications received will only be ensured to be recent. algorithm 1 shows the implementation details of verifying, revoking, and checking the status of a claim. Several functions implement the claim verification process, including `verifyPatient`, `verifyDoctor`, `verifyHospital`, `verifyGuardianship`, `verifyRequestByRAM`, and `verifyRequestByGuardian`.

To verify a claim that is made, the algorithm learns whether the verification is new or repeated, and in case the claim requires specific verifiers it makes sure the new verification is one of those. New verifications are added to the set of verifiers, whereas repeated ones only update the date of verification. To revoke a claim verification, the algorithm simply iterates through the set of verifiers, then finds and deletes the verification if it was found. To check the overall verification state of a claim, the algorithm mainly tracks the number of recent verifications, both optional and required, then compares them with the minimum number of verifications needed.

Verified patients can make a claim that a certain verified person is their guardian, using the `addGuardian` function. The function takes the address of the guardian and adds it to a set of claims of guardianships, in which each claim can be only verified by the guardian, using the `verifyGuardianship` function. Patients can remove any person from being their guardian using the `removeGuardian` function. Our implementation of these functions is detailed in algorithm 2.

Verified patients can also submit their medical documents to `RASC` while specifying the wanted number of RAMs or guardians. However, as seen in the implementation details in algorithm 3, the actual needed number of RAMs and guardians is bounded between a range that is seen appropriate by the RAO, preventing the patient from mistakenly choosing zero or an impossible number of verifiers.

Algorithm 1 Algorithm for Verifying, Revoking, and Checking the Verification Status of a Claim

`verifyClaim`: Verifying a claim, performed by an identified verifier
Function `verifyClaim` (Claim `c`) **:**
  Require that `c.made` is *true*
  `didVerify` ← `msg.sender` ∈ `c.verifiers`
  `i` ← index of `msg.sender` ∈ `c.verifiers`
  if ¬ `didVerify` **then**
    `c.verifiers` ← `msg.sender` ∪ `c.verifiers`
    `c.dates` ← `now` ∪ `c.dates`
  else
    $c.dates_i$ ← `now`

`revokeClaim`: Revoking a claim verification, performed by a previous verifier
Function `revokeClaim` (Claim `c`) **:**
  `didVerify` ← `msg.sender` ∈ `c.verifiers`
  `i` ← index of `msg.sender` ∈ `c.verifiers`
  if `didVerify` **then**
    `c.verifiers` ← `c.verifiers` − $c.verifiers_i$
    `c.dates` ← `c.dates` − $c.dates_i$

`isClaimVerified`: Checking the overall verification status of a claim, can be performed by various types of entities
Function `isClaimVerified` (Claim `c`) **:**
  if ¬ `c.made` ∨ (`c.verifiers.length` < `c.numRequiredVerifiers`) **then**
    Return *false*
  `vers` ← 0, `reqVers` ← 0
  for `i` ← 0 ... `claim.verifiers.length` **do**
    if `now` − $claim.dates_i$ ≤ `claim.oldestDate` **then**
      `vers` ← `vers` +1
      `reqFound` ← `msg.sender` ∈ `c.reqVerifiers`
      if ¬ `c.anyVerifier` ∧ `reqFound` **then**
        `reqVers` ← `reqVers` +1
  if `c.anyVerifier` **then**
    Return `vers` ≥ `c.numRequiredVerifiers`
  else
    Return (`vers` ≥ `c.numRequiredVerifiers`) ∧ (`reqVers` ≥ `c.reqVerifiers.length`)

A verified doctor can request a specific medical document from the patient's record by submitting the patient address and the medical document's IPFS hash to the `requestMDDoctor` function shown in algorithm 3. The doctor must also specify whether the request must be approved by the patient directly, or by the MPA, which consists of either RAMs and guardians for incapacitated patients, or RAMs and hospital in case of emergency.

Algorithm 2 Algorithm for Adding, Removing, and Verifying Guardianships

`addGuardian`: Adding a new guardian, performed by the patient
Function `addGuardian` (Guardian address `ga`) **:**
  `p` ← `persons[msg.sender]`
  Require that `isClaimVerified` (`p.patientClaim`) is *ture*
  Require that `p.guardians.length` < 5
  Require that `ga` ∉ `p.guardians`
  `p.guardians` ← `p.guardians` ∪ `ga`
  Issue a claim of guardianship, require verification of `ga`

`removeGuardian`: Removing an existing guardian, performed by the patient
Function `removeGuardian` (Guardian address `ga`) **:**
  `p` ← `persons[msg.sender]`
  Require that `ga` ∈ `p.guardians`
  `p.guardians` ← `p.guardians` - {`ga`}
  Retract the claim of guardianship

`verifyGuardianship`: Verifying the claim of guradianship made by the patient, performed by the relevant guardian
Function `verifyGuardianship` (Person address `pa`) **:**
  `p` ← `persons[pa]`
  Require that `msg.sender` ∈ `p.guardians`
  Call `verifyClaim` on the guardianship claim

As a result of the claim verifications having a validity period, a doctor's request can be used to receive a certain medical document for a specific period of time chosen by the patient. This allows patients to grant partial or full access to their health records for a limited period of time, even if they are no longer available to directly grant or deny access to their health records.

To proceed with the typical sequence of actions for a successful medical document sharing, the details of a verified request are emitted by the `RASC` to be seen by the RGTOs. RGTOs will fetch the medical document from IPFS, get the pseudo-random number, and submit the number to the `RASC` using the `addRGTOResponse` function. Based on the correctness and latency of the responses, and based on the current reputation of the RGTOs, `RASC` picks a winning RGTO that will re-encrypt the medical document and send it to the doctor. All RGTOs have an RGTO-`RASC` (OR) reputation and an RGTO-doctor (OD) reputation, the first of which is done by `evluateORReputations` function to keep track of the correctness and latency of RGTO responses, while the second is done by `evaluateODReputation` function to keep track of the experiences of doctors with the RGTO.

TABLE 1. Testing accounts and their Ethereum addresses.

Account Name	Ethereum Address	Account Name	Ethereum Address
RAO	0x4e6fc547fBEE09955208692df39CB6ea8b272304	Guardian 1	0xa9ACb790f0511cA12224466B8E402FA1ABB5599F
RAM 1	0xEf90126d509A488CfcD685Ff77E0500B6BE3e5A7	Guardian 2	0x411df4fee441cEF1d8cc9f064e3649B322670c7f
RAM 2	0xAc12d706143893cE74F98C3Ca3D3D54299F27fA3	Guardian 3	0xe20641E497910b8295f44FF2a31D6b81bf9DA49A
RAM 3	0x816AF185f84cCE1eA8246A8177f42E9619c40b74	Doctor	0x74eA3e3Bfd7DE112e98B12bFc244059Eb1367841
RAM 4	0x9bBEB269C3D17dCcc1C69A7e210e07Cd005521b5	RGTO 1	0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c
Patient	0xEb4Bc6470aBE95b6404B890157234dd2ca8E20E0	RGTO 2	0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C
		RGTO 3	0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB

Algorithm 3 Algorithm for Submitting and Requesting Medical Documents

`submitMDPatient`: Submitting a new medical document, performed by the patient
Function `submitMDPatient` (IPFS hash `h`, Number of RAM verifications `nram`, Number of guardian verifications `ng`) **:**
- `p` ← `persons[msg.sender]`
- Require that `isClaimVerified(p.patientClaim)` is *ture*
- `p.IPFSHashes` ← `h` ∪ `p.IPFSHashes`
- Create a `MedDoc` file with `nram` and `ng`
- Add the `MedDoc` file to `p.meddocs`

`requestMDDoctor`: Requesting a certain patient's medical document, performed by the doctor
Function `requestMDDoctor` (Patient address `pa`, IPFS hash `h`, Minimum number of RGTOs `minr`, Maximum number of RGTOs `maxr`, Request type `t`) **:**
- `p` ← `persons[pa]`
- Require that `p.meddocs[h]` exists
- Require that `minr` ≤ `maxr`
- Create a `Request` with `minr` and `maxr`
- Add the `Request` to `p.meddocs[h].requests`
- **if** `t` is *PATIENT* **then**
  - Issue a claim requiring the patient approval
- **else if** `t` is *INCAPACITATED* **then**
  - Issue a claim requiring the RAM approvals
  - Issue a claim requiring the guardian approvals
- **else if** `t` is *EMERGENCY* **then**
  - Issue a claim requiring the RAM approvals
  - Issue a claim requiring the patient's admission hospital approval

IV. DISCUSSIONS

In this section, we present our testing methodology, results, and discussions. Our testing of the implementation includes a correctness verification that ensures the relevant functions execute correctly according to the typical sequence of interactions. Additionally, we perform a cost analysis of all the developed algorithms and showcase our results in gas units and physical currency. Furthermore, we discuss various security aspects of our architecture design, highlight generalization possibilities and potentials, and describe open challenges and limitations.

registerRGTO

from	0xEf90126d509A488CfcD685Ff77E0500B6BE3e5A7
input	-
output	Error: Only unregistered entities can call this function
events	-

FIGURE 4. Transaction details of a repeated registration attempt by RAM 1.

A. CORRECTNESS VERIFICATION

To perform correctness verification of the smart contract, we compiled, deployed, and executed the code according to the typical sequence of actions as described in the previous section. Our testing involved 13 unique Ethereum accounts, as shown with their corresponding Ethereum addresses in Table 1. The compilation was made with Solidity compiler version 0.6.12 with code optimization enabled, and the deployment was on a JavaScript-based EVM running on a virtualized local Ethereum testnet. The major six phases involved in the life span of sharing medical documents are discussed below.

1) Deploying smart contract and registering entities: The RAO deploys the `RASC` and registers all RAM entities. After that, the patient, guardians, and doctor register their accounts as a person. The RGTOs register their accounts as RGTOs. Once an entity is registered, it cannot register as a different type of entity, this design decision is made to separate the roles and privileges of each type. Figure 4 shows the transaction result of RAM 1 trying to register as an RGTO.

2) Verifying identities: Accounts registered as a patient must have their identities verified before being allowed to perform any action using their accounts. RAMs verify the claims of each person. A person requires any two verifications by RAMs to be considered as a verified patient, these verifications expire after 1 year, requiring to be resubmitted. A doctor requires 4 RAM verifications, which expire every 3 years.

3) Adding and verifying guardians: The verified patient can add up to 5 guardians to his account. Guardianship claims must be verified by the relevant person. A guardianship claim must be renewed every 2 years.

FIGURE 5. Transaction details of RAM 2 verifying a medical document request claim made by the doctor.

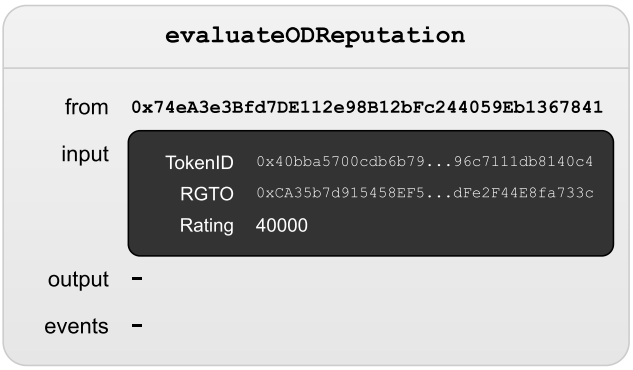

FIGURE 6. Transaction details of the doctor giving a score of 40,000 out of 65,535 to RGTO 1.

4) Submitting and requesting medical documents: The patient uploads the encrypted medical documents to IPFS and submits the hash to RASC along with the desired number of RAMs and guardians needed to approve permitting a doctor request in cases of patient absence. The doctor can request the medical document of patients, and based on the request type, it can be forwarded to the MPAs.

5) Verifying requests: The required number of RAMs and guardians approve the medical document request made by the patient, upon which details about the request, such as the medical document IPFS hash and request identifier are shared with the RGTOs. Any verification made by the MPA members results in sending an event to the patient to inform them about the status of the request, as Figure 5 depicts.

6) Evaluating RGTOs and choosing the winner: The RGTOs respond to the RASC with a proof that they successfully fetched the file from IPFS. For each RGTO response, the RASC will evaluate whether the received responses are sufficient or not based on the minimum and maximum number of RGTOs specified by the doctor, and by a timeout of 1 hour. The minimum and maximum number of RGTOs the doctor can specify must fall within the range of 2 to 20. Setting a fixed limit on the number of responses caps the cost of the RGTO evaluation function, which is called when a sufficient number of RGTOs respond to the request. Further, after the doctor contacts the selected RGTO, they can submit their rating of the off-chain performance, as seen in Figure 6.

B. COST ANALYSIS

To perform cost analysis of our developed functions, we executed all of them as part of our testing and verification, giving us the output of the transactions that reveal the transaction and execution costs of the function in units of gas. The execution cost includes the computational operations done in the EVMs, whereas the transaction cost includes the execution cost plus the cost of sending the transaction data and smart contract deployment. Generally, the cost gets higher as a function accesses more smart contract state variables.

Table 2 details all the transaction and execution costs of our functions, in addition to showing how these gas prices translate to the real world by estimating their value in USD. The average gas price as of September 8, 2020, is 82 Gwei, and the average price of Ether is $345.52. The gas price plays a role in determining how fast the transactions are mined since higher gas prices mean higher incentives for miners.

As evident in the table, the deployment of the RASC makes up the majority of the overall cost of the smart contract, however, it is worth considering that RASC is a one-time deployment smart contract. The remaining functions have much lower costs, ranging from just above $1.00 to around $10.00.

C. SECURITY ANALYSIS

Our solution enables patients to securely share their medical documents with doctors without compromising their privacy. Herein, we evaluate our solution against different security parameters.

- **Authenticity**: Our architecture is comprised two layers that protect against authentication attacks, which rely on the Ethereum address verification and physical identity authentication. All entities involved in interacting with the smart contracts are required to have an Ethereum account with a valid Ethereum address that is managed by an Ethereum wallet, therefore, unless an individual entity loses its exclusive possession of the private Ethereum key due to undefendable security attacks, such as social engineering, it can be trusted to be operated by its authentic owner, with zero chance of having the address of an entity be tampered with. On top of that, the presence of a regulatory agency that authenticates the identity of each component in the network protects against identity theft attacks, which is an important aspect because otherwise, entities can claim to be of a certain high-privilege category without consequences. However, even in the case of mistaken categorization of identity, our medical document sharing architecture protects against that by being a patient-centered solution that does not share any information unless the patient explicitly grants such access. On top of that, our implementation design prohibits the owner of one Ethereum

TABLE 2. Function caller, and gas and currency costs of smart contract functions.

Function Caller	Function Name	Transaction Cost [Gas]	Execution Cost [Gas]	Cost [USD]
RAO	RASC	5,486,067	4,081,475	155.43
RAO	registerRAMember	46,002	23,322	1.30
Hospital	registerPatient	109,778	88,506	3.11
Patient	registerHospital	108,845	87,573	3.08
RGTO	registerRGTO	47,969	26,697	1.36
RAM	verifyPatient	84,923	62,243	2.41
Doctor	registerDoctor	99,211	77,939	2.81
RAM	verifyDoctor	84,967	62,287	2.41
Patient	addGuardian	261,175	238,495	7.40
Patient	removeGuardian	50,773	78,865	1.44
Guardian	verifyGuardian	52,035	29,355	1.47
Patient	addMDPatient	148,061	124,229	4.19
Doctor	requestMDDoctor	394,644	369,212	11.18
RAM	verifyRequestRAMember	90,159	65,111	2.55
Guardian	verifyRequestGuardian	44,260	19,276	1.25
RGTO	addRGTOResponse	93,916	66,756	2.66
RGTO	evaluateORReputation	277,370	250,210	7.86
Doctor	evaluateODReputation	38,829	13,717	1.10

address to operate multiple entities, thus eliminating a possible Sybil attack even further.

- **Confidentiality and privacy**: In our solution, no single information about any entity is made public, as all medical documents and secret metadata are stored privately on IPFS and the Ethereum network using symmetric and public-key encryption. Moreover, even the identities of patients are kept private, reducing the chances of data theft of a targeted PHR. Additionally, sharing the medical documents with doctors is done through PRE, which means no private key used in the encryption of data is disclosed, and no entity other than the patient and granted doctors can access any medical document. Even in the cases where the patient is having an emergency or in an incapacitated state, the sharing permissions goes through a distributed MPA that includes the regulatory agency, trusted guardians, and hospital, and the only condition for the patient medical documents to be shared without patient consent is for these parties to collaborate maliciously at the same time within the 1 hour request window. Our solution ensures the system is kept patient-centered, even with the existence of a trusted regulatory agency, by implementing all processing on decentralized smart contracts that cannot be self-destructed by the agency, and by allowing the patient to rate the MPA parties such that a non-consensual medical document sharing is never approved and repeated by taking away any MPA privileges on the patient data.
- **Integrity and traceability**: 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 health records, token creation, and transmission, and reputation calculations, all along with their changes across time for full provenance ability.
- **Availability**: 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 health records. Furthermore, health records are stored in a distributed and decentralized server, 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, such as distributed denial-of-service (DDoS).
- **Non-repudiation**: At every step in the span of any medical document sharing, starting from the deployment of the smart contract, and ending with evaluating the RGTOs, there is no way for any entity to deny any action it has made, whether that was on-chain or off-chain. Building our system from the ground up based on blockchain provides full traceability of all actions, because of logging all shreds of evidence immutably on the blockchain ledger, which means neither the sender nor the receiver of a certain message can deny sending or receiving it.

D. GENERALIZATION

Our proposed solution is tailored for the healthcare context in the United Arab Emirates; however, it can be generalized and scaled to other smaller or larger regions of different population sizes and healthcare hierarchy. Moreover, the design of our architecture does not put constraints on the mechanism the region verifies the identities of its people, but rather makes the mechanism more transparent and more enforceable. For example, the RAMs that verify a patient's identity can be employees of different government departments, such as the department of health and the department of digital authority.

A more generalized perspective of our solution can find it useful in completely different use cases outside the healthcare system. The patient in our architecture can be thought of

as a user who uploads private data and provides data to the accepted requesters, in addition to allowing trusted parties to give the requester access to the data in case the user is absent. On top of that, thanks to using RGTOs as PRE nodes, regardless of the status of the user, the data is kept confidential all the time, and no third party, other than the accepted requester, can view it.

E. OPEN CHALLENGES

- **Securing Ethereum private keys**: Ethereum and DApps are new technologies that are still not used by any major solution that is deployed nationwide. Part of the reason is that a nationwide deployment would require demanding the general public to store the Ethereum private keys and passphrases securely and not sharing them with any other person. Considering the current state of failing to secure private information, combined with the spread of social engineering attacks, the users easily become the weakest link in the chain.
- **Solution complexity**: Our architecture integrates various techniques with blockchain to implement a decentralized solution for sharing encrypted medical documents even if the patient is absent. Although the result does achieve our goal, the solution with each added technique gets more complex, which makes it harder to deploy in the real world.
- **Optimizing Solidity code**: After writing and verifying the code of smart contracts, a necessary step of optimizing the code for the least amount of computation and data operations is crucial to avoid ending up with expensive smart contract functions. In Solidity, such optimizations can take a large portion of the development cycle, especially due to the scarcity of advanced debugging and optimization tools, unlike other more mature programming languages.
- **Upgrading smart contracts**: Another limitation of Ethereum smart contracts is the inability to upgrade them after deployment. Such a limitation prevents patching security vulnerabilities and software flaws, thereby putting users at security risks and several other unexpected problems.

V. CONCLUSION

In this paper, we have proposed a fully decentralized blockchain-based multi-party consent management solution for sharing and granting access to encrypted medical documents in a manner that is secure and trustworthy. Our approach maintains complete action traceability while providing an architecture for patients to authorize trusted entities to make access permission decisions on their behalf in case of emergencies. We implemented multi-party authorization and threshold cryptographic using blockchain technology to allow the patients to securely share and grant access to their medical documents along with sharing their secret keys. We integrated decentralized IPFS storage with our system architecture and introduced reputation-governed trusted oracles to mitigate the data and computation related limitations posed by the blockchain platform. We presented algorithms along with their full implementation details. Our correctness verification and cost analysis tests of the implementation verify the functionality and affordability of our system. Our code is made publicly available on GitHub, with a detailed description of how to reproduce our testing results. We perform security analysis and discuss how the proposed approach provides authenticity, confidentiality, integrity, availability, and non-repudiation.

REFERENCES

[1] S. S. Woods, E. Schwartz, A. Tuepker, N. A. Press, K. M. Nazi, C. L. Turvey, and W. P. Nichol, "Patient experiences with full electronic access to health records and clinical notes through the my HealtheVet personal health record pilot: Qualitative study," *J. Med. Internet Res.*, vol. 15, no. 3, p. e65, Mar. 2013.

[2] 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.

[3] (2018). *Health Records–Apple*. Accessed: Mar. 16, 2020. [Online]. Available: https://www.apple.com/healthcare/health-records/

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

[5] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73295–73305, 2019.

[6] H. R. Hasan and K. Salah, "Blockchain-based solution for proof of delivery of physical assets," in *Proc. Int. Conf. Blockchain*. Cham, Switzerland: Springer, 2018, pp. 139–152.

[7] A. Yousafzai and C. S. Hong, "SmartSON: A smart contract driven incentive management framework for self-organizing networks," 2020, *arXiv:2008.11803*. [Online]. Available: http://arxiv.org/abs/2008.11803

[8] I. Yaqoob, K. Salah, R. Jayaraman, and Y. Al-Hammadi, "Blockchain for healthcare data management: Opportunities, challenges, and future recommendations," *Neural Comput. Appl.*, to be published.

[9] H. R. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, vol. 6, pp. 65439–65448, 2018.

[10] M. M. Madine, A. A. Battah, I. Yaqoob, K. Salah, R. Jayaraman, Y. Al-Hammadi, S. Pesic, and S. Ellahham, "Blockchain for giving patients control over their medical records," *IEEE Access*, vol. 8, pp. 193102–193115, 2020.

[11] T.-S. Chen, C.-H. Liu, T.-L. Chen, C.-S. Chen, J.-G. Bau, and T.-C. Lin, "Secure dynamic access control scheme of PHR in cloud computing," *J. Med. Syst.*, vol. 36, no. 6, pp. 4005–4020, Dec. 2012.

[12] A. G. M. Alzahrani, A. Alenezi, A. Mershed, H. Atlam, F. Mousa, and G. Wills, "A framework for data sharing between healthcare providers using blockchain," Tech. Rep., 2020.

[13] 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. 14757–14767, 2017.

[14] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–5.

[15] H. S. G. Pussewalage and V. A. Oleshchuk, "An attribute based access control scheme for secure sharing of electronic health records," in *Proc. IEEE 18th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Munich, Germany, Sep. 2016, pp. 1–6.

[16] W. Li, W. Ni, D. Liu, R. P. Liu, P. Wang, and S. Luo, "Fine-grained access control for personal health records in cloud computing," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Sydney, NSW, Australia, Jun. 2017, pp. 1–5.

[17] K. Seol, Y.-G. Kim, E. Lee, Y.-D. Seo, and D.-K. Baik, "Privacy-preserving attribute-based access control model for XML-based electronic health record system," *IEEE Access*, vol. 6, pp. 9114–9128, 2018.

[18] X. Zhang and S. Poslad, "Blockchain support for flexible queries with granular access control to electronic medical records (EMR)," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[19] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018.

[20] T. T. Thwin and S. Vasupongayya, "Blockchain-based access control model to preserve privacy for personal health record systems," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Jun. 2019.
[21] R. Adlam and B. Haskins, "A permissioned blockchain approach to the authorization process in electronic health records," in *Proc. Int. Multidisciplinary Inf. Technol. Eng. Conf. (IMITEC)*, Vanderbijlpark, South Africa, Nov. 2019, pp. 1–8.
[22] A. R. Rajput, Q. Li, M. T. Ahvanooey, and I. Masood, "EACMS: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84304–84317, 2019.
[23] A. A. Battah, M. M. Madine, H. Alzaabi, I. Yaqoob, K. Salah, and R. Jayaraman, "Blockchain-based multi-party authorization for accessing IPFS encrypted data," *IEEE Access*, vol. 8, pp. 196813–196825, 2020.
[24] 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/
[25] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.
[26] (2018). *Quorum Whitepaper*. Accessed: Mar. 23, 2020. [Online]. Available: https://github.com/jpmorganchase/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf
[27] (2020). *Besu Enterprise Ethereum Client*. Accessed: Mar. 23, 2020. [Online]. Available: https://besu.hyperledger.org/en/stable/
[28] L. T. Brandao, N. W. Mouha, and A. T. Vassilev, "Threshold schemes for cryptographic primitives," Tech. Rep., 2019, doi: 10.6028/NIST.IR.8214.
[29] J. Benet. (2014). *IPFS–Content Addressed, Versioned, P2P File System*. Accessed: Mar. 26, 2020. [Online]. Available: https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf
[30] 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.
[31] *Ethereum Alarm Clock*. Accessed: Jul. 25, 2020. [Online]. Available: https://www.ethereum-alarm-clock.com/

MOHAMMAD MOUSSA MADINE (Member, IEEE) received the B.Sc. degree in computer engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2019. He is currently a Graduate Researcher and a Teaching Assistant for pursuing his master's degree students. His research interests include primarily focused on blockchain solutions in healthcare, personal health records, and edge computing.

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, Chicago, IL, USA, in 1994 and 2000, respectively. He has more than 220 publications and three U.S. patents, and 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 is a member of the IEEE Blockchain Education Commitee. He served as the Chair of the Track Chair of IEEE GLOBECOM 2018 on Cloud Computing. He is an Associate Editor of IEEE Blockchain Tech Briefs. He is currently leading a number of projects on how to leverage blockchain for healthcare, 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. His expertise is in multicriteria optimization techniques applied to diverse applications, including supply chain and logistics, healthcare, energy, environment, and sustainability. He is currently an Associate Professor with the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests are primarily focused on using 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 U.S. 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.

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 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. 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, United Arab Emirates. His research interests include big data, blockchain, edge computing, mobile cloud computing, the Internet of Things, healthcare, and computer networks. 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 is also serving/served as a Guest/Associate Editor for various journals. He has been involved in a number of conferences and workshops in various capacities.

YOUSOF AL-HAMMADI received the bachelor's degree in computer engineering from the Khalifa University of Science and Technology (previously known as the Etisalat College of Engineering), United Arab Emirates, in 2000, the M.Sc. degree in telecommunications engineering from the University of Melbourne, Melbourne, VIC, 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 the Acting Dean of Graduate Studies and an Assistant Professor with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His main 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.

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

He worked in Washington DC at Georgetown University Hospital, the Washington Hospital Center, and in several clinical and leadership positions before moving to United Arab Emirates, in 2008. 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 was recently recertified in 2017 by the American Board of Cardiology (ABIM). He is currently a Cleveland Clinic Caregiver, Cleveland, OH, USA, and seconded as a Senior Cardiovascular Consultant and the Director of Accreditation with the Quality and Safety Institute, Cleveland Clinic, Abu Dhabi. He is also an ISQua Expert, a member of the AHA Hospital Accreditation Science Committee, a member of the European Society of Cardiology Heart Failure Writing, a member, an ex-Middle East Representative of the JCI Standards Subcommittee, and a member of the American College of Cardiology Accreditation Foundation Board. He is also American Board Certified in Internal Medicine, Cardiovascular Disease, and Vascular Medicine. 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), certified as an EFQM Model assessor, and a Lead Trainer with TeamSTEPPS. He is also certified in EFQM, FACMQ, FACP, FACC, FAHA, and FCCP. He finished his internal medicine residency at Georgetown University Hospital, the Washington Hospital Center. He also 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 multidisciplinary 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 USA. He is the Champion of the AHA GWTG in the region. He has served as the 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 the Chief Quality Officer and the 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 U.S.-validated, risk-adjusted, outcomes-based program to measure and improve the quality of surgical care. SKMC is the first multispecialty ACS NSQIP center outside the USA. He led the publication of, first in the region, annual SKMC outcome books since 2011 and he is a strong believer in transparency in health care and external reporting. He is 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 a recognized world-leader in these fields.

Dr. Ellahham is a Fellow of the American College of Cardiology, American Heart Association, American College of Chest Physicians, American College of Physicians, and American College of Medical Quality. He is a Fellow of 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. His fellowship in Cardiology at the Virginia Commonwealth University Health System, Richmond, VA, USA. He is a Distinguished Fellow of the New Westminster College in British Columbia, Canada, and an Advisory Board Member of the University of Wollongong, Dubai. He was 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 a Leader of the use of Lean, Six Sigma, and Change Management to improve healthcare quality and has numerous publications in this area. He is a Lean Six Sigma Master Black Belt Certified. He is an American Society of Quality (ASQ) trainer in Lean and Six Sigma both green and black belt. He is 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 serves on several U.S. and international prestigious committees and advisory bodies. He received several research awards, including the DuPont Pharmaceuticals Research Award, the ACCP 58th Annual Scientific Assembly, the Young Investigator Award, and the Alfred Soffer Research Award. He is a Finalist of the ACCP 58th Annual Scientific Assembly. He also received the First Young Investigator Award of the 12th Annual Meeting of the Mediterranean Association of Cardiology and Cardiac Surgery, the American Heart Association Get With the Guidelines Award, the SKMC Infection Prevention Award in 2011 and 2012, the Sheikh Khalifa Excellence Award in 2014, the Quality Leadership Award from the World Quality Congress and awards, the 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, the Dubai Quality Award in 2015, and the Sheikh Khalifa Excellence Golden Award in 2015. He was also a recipient of the AHA GWTG Award in Wash, DC, USA. He is the Middle East Regional Chair of the Patient Safety Movement Foundation. He is the Eminent Editor of *The Journal of Cardiology & Cardiovascular Therapy* and the Associate Editor of the *American Journal of Medical Quality*. In addition, he 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*, *Focus on Hypertension Journal*, *Journal of Heart Health*, *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 of several peer-reviewed journals, including *Joint Commission Journal on Quality and Patient Safety*, *International Journal of Quality & Reliability Management*, the *Journal of the 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.

PRASAD CALYAM (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering, The Ohio State University, in 2002 and 2007, respectively. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA, and also directs the VIMAN laboratory. His current research interests include distributed and cloud computing, computer networking, and cyber security.