IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 December 2022, accepted 6 February 2023, date of publication 14 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3245046

RESEARCH ARTICLE

Sec-Health: A Blockchain-Based Protocol for Securing Health Records

LEONARDO DA COSTA[1], BILLY PINHEIRO[2], WEVERTON CORDEIRO[3], ROBERTO ARAÚJO[1], AND ANTÔNIO ABELÉM[1]

[1]Department of Computer Science, Federal University of Pará (UFPA), Belém 66075-110, Brazil
[2]Amazônia Blockchain Solutions, Belém 66075-750, Brazil
[3]Institute of Informatics (INF), Federal University of Rio Grande do Sul (UFRGS), Porto Alegre 91509-900, Brazil

Corresponding author: Leonardo da Costa (lbc@ufpa.br)

This work was supported in part by the São Paulo Research Foundation (FAPESP) under Grant 2020/04031-1 and Grant 2020/05183-0. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)-Finance Code 001, and in part by the Pró-Reitoria de Pesquisa e Pós-Graduação (PROPESP)/Universidade Federal do Pará (UFPA) [Programa de Apoio à Publicação Qualificada (PAPQ)].

ABSTRACT Storing and sharing health records through electronic systems pose security risks. To address them, several countries' regulations have established that healthcare information systems must fulfill security properties (confidentiality, access control, integrity, revocation and anonymity) and complementary ones (emergency access and interoperability). Upon tackling these issues, several proposals present security limitations and/or address specific properties only. We propose Sec-Health, a blockchain-based protocol that secures health records, addressing all of the main security and complementary properties defined in current regulations. We show that Sec-Health is a suitable solution by analyzing it under several attack scenarios and describing how it overcomes the problems of existing solutions. Furthermore, we evaluate a Sec-Health Proof of Concept, showing that it can reduce from 26% up to 90% the time to access health records, and reduce up to 50% client-side memory overhead, compared to related work.

INDEX TERMS Blockchain, computer security, data security, electronic healthcare, health information management, regulation.

I. INTRODUCTION

Information technologies introduce a number of resources and benefits to the healthcare field. Electronic Health Records (EHRs), such as patient's medical history, are one of the most widely employed resources [1], providing a wide view of a patient's medical status. EHRs are commonly originated and shared with collaborators (e.g., physicians, nurses) through cloud computing systems, which results in a more convenient approach to managing such records. Cloud-based systems, however, introduce security challenges in healthcare [2]. A recent report shows that healthcare data breaches are highly common [3], wherein several of them are classed as unauthorized access, which may lead to inappropriate use of health records (e.g., unwanted advertisements or lower chances of conquering a job opportunity).

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

Due to security vulnerabilities, various countries (e.g., USA, Brazil, and those from European Union) have established regulations defining health records as sensitive data that should be shared only under patient consent [4]. Such regulations define several requirements, which we call health record properties. For instance, only authorized collaborators should access health records (confidentiality and access control properties). Records must also be protected from unauthorized modifications (integrity property). Mechanisms must also exist to legitimately grant access to records in emergency situations (emergency access property), and to anonymized records for research purposes (anonymity property). Besides, the properties of access revocation and interoperability must also be addressed.

Those properties motivate the design of solutions to secure healthcare information systems. A number of literature proposals provide schemes based on centralized servers to store and share health records (e.g., [5], [6]). The security

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

of such solutions rely on the fact that the server is trusted not to disclose sensitive data, such as information related to user credentials and patient records. This results in a single point that, when compromised, can make the entire system fail. Moreover, these solutions address only a subset of health record properties, while not fulfilling fundamental ones.

Other several works propose the use of decentralized approaches to secure health records (e.g., [7], [8]). For instance, blockchain [9], a technology that improves data security by allowing online data transactions in a decentralized fashion, has been adopted in different protocols. Although these schemes do not present a single point of failure, they still lack an integrated approach that covers all of the aforementioned health records properties, then presenting security limitations. Therefore, there is a lack of proposals in the literature that address all of the properties and afford satisfactory security to health records.

Motivated by the properties defined in regulations and the literature limitations, we propose Sec-Health, a protocol that secures health records by addressing all of their properties. In essence, Sec-Health is composed of a set of schemes, based on decentralized approaches (e.g., blockchain and InterPlanetary File System [10]) and cryptographic primitives (e.g., Ciphertext-Policy Attribute-based Encryption [11] and public key encryption), which allow records to be stored and shared securely. Sec-Health fills the gap of the literature that lacks integrated approaches which fulfill all health records properties. It overcomes the security problems of proposals based on centralized servers and presents advantages over other decentralized solutions by covering not only the most addressed properties of health records (confidentiality, access control, and integrity), but also more challenging ones (e.g., emergency access, access revocation, and anonymity).

In this paper, we provide the following novel contributions: (i) a blockchain-based protocol (Sec-Health), based on our previous work [12], which enhances the schemes employed in the previous protocol to fulfill the security properties of confidentiality, access control, and integrity; (ii) Sec-Health includes novel schemes to address additional properties, i.e., emergency access, access revocation, anonymity, and interoperability; (iii) an analysis of Sec-Health, describing the security strategies adopted under several attack scenarios, explaining how it satisfies health record properties, and comparing it with related work; and (iv) an experimental evaluation of a Sec-Health Proof of Concept (PoC), showing that it can reduce from 26% up to 90% the time to access health records, and reduce up to 50% client-side memory overhead, compared to related work.

The remainder of this paper is structured as follows. Section II introduces health record properties defined in current regulations. Section III presents background on technologies that Sec-Health employs, while Section IV presents our protocol in detail. In Section V, we describe related work. In turn, in Section VI, we evaluate Sec-Health, describing how it secures health records under several attacks, presenting experimental results, and comparing Sec-Health with related work. Finally, Section VII presents conclusions and points out future work.

II. PROPERTIES OF HEALTH RECORDS

Because health records are targeted by cybercriminals, several countries established regulations requiring any entity to employ security measures when handling health data. The Health Insurance Portability and Accountability Act (HIPAA), enacted by the United States Congress in 1996 [13], provides guidelines that must be observed by all national healthcare organizations (e.g., hospitals). In 2016, the European Union has approved the General Data Protection Regulation (GDPR), recognizing that health records need special limitations regarding access and treatment through appropriate security mechanisms [4]. Inspired by the GDPR, Brazil's government enacted the General Law for Personal Data Protection (LGPD) that presents similar principles [14].

In general, the regulations define requirements about how health records must be collected and processed by organizations and establish penalties for deviations from them (e.g., unauthorized data access or alteration). The regulations establish a number of security properties in common, which are well-known in the literature of health records security [2], as follows:

- *Confidentiality:* technical measures must be adopted to keep health records inaccessible and/or unintelligible for parties that have no permission to gain any knowledge about them.
- *Access control:* patients must own the right to control who accesses their health records, providing consent for any collaborator or type of collaborator that will have access to the records.
- *Integrity:* health records must be protected against unauthorized modification and deletion.
- *Revocation:* patients have the right to revoke, at any moment, the consent given for any collaborator to access their records. In the literature for access revocation, well-accepted mechanisms fulfill the following requirements [15]: (i) once the patient revokes access from a given collaborator, the new access rights are immediately updated on the appropriate system such that the collaborator cannot access the record using this system anymore; (ii) revoking access from a given collaborator should not interfere in the access rights given to another collaborator; and (iii) revoking access from a given collaborator should not require generating new material (e.g., credentials, cryptographic keys) to any collaborator.
- *Anonymity:* research organizations may handle health records, under patient consent, during the realization of studies (e.g., public health research), provided that the records are anonymized beforehand.

Other properties, not specifically related to security, define how health records must be handled in specific cases. In our work, we call them complementary properties, which are introduced in the following:

- *Emergency access:* in order to maintain a patient's health condition stable, the regulations provide that health records from the patient may be accessed by healthcare collaborators without patient consent within emergency scenarios. Those collaborators may have not been given access rights to this patient data before. A secure mechanism is necessary such that only collaborators participating in the emergency situation access patient records [16].
- *Interoperability:* patients own the right to, at any moment, change service provider (e.g., insurance, hospital, physician), and request that their health records be accessed by the new provider under proper given permissions only. A standard on communication and data formats should exist between service providers to facilitate switching from one provider to another.

III. BUILDING BLOCKS

To provide a background to clarify Sec-Health, we now introduce cryptographic primitives and present key concepts of blockchain and InterPlanetary File System, building blocks that we use in Sec-Health. While presenting such background, we define the notation used throughout the paper.

A. CRYPTOGRAPHIC PRIMITIVES

1) PUBLIC KEY ENCRYPTION

In a public key (or asymmetric) encryption scheme (e.g., RSA) [17], a sender u_1 encrypts a message m with a public key pk_{u_2} belonging to a receiver u_2. The resulting ciphertext $E(m, \mathsf{pk}_{u_2})$ can only be decrypted by user u_2, who owns the corresponding private key sk_{u_2}. To this end, u_2 should compute $D(E(m, \mathsf{pk}_{u_2}), \mathsf{sk}_{u_2})$, which returns m.

2) ONE-WAY HASH FUNCTIONS

A hash function (e.g., SHA-256) [18] is a deterministic mathematical algorithm that receives a message m as input and outputs a fixed-size hash value $H(m)$. For a specific message m, this function always outputs the same hash value.

3) THRESHOLD CRYPTOSYSTEMS

In a threshold public key cryptosystem (e.g., threshold ElGamal) [19], a key pair $\langle \mathsf{pk}_P, \mathsf{sk}_P \rangle$ belongs to a set of n parties $P_1, \ldots, P_n$. While public key pk_P is represented by a single element, private key sk_P is shared, where share sk_{P_i} is known only to party P_i. To encrypt a message m, a user employs pk_P. In turn, the decryption process of ciphertext $E(m, \mathsf{pk}_P)$ requires at least $t \leq n$ parties to generate decryption parts $D(E(m, \mathsf{pk}_P), \mathsf{sk}_{P_i})$ using their private key share sk_{P_i}. Message m is obtained by joining at least t decryption parts.

4) CP-ABE

In Ciphertext-Policy Attribute-based Encryption (CP-ABE), data is encrypted with an access policy, a tree-based structure with attributes interrelated through logical operators, such as AND, OR [11]. Decryption occurs with the secret key of a user, composed of attributes. It is only successful if the key's attributes satisfy the access policy used for encryption. As example, let "*(Director OR (Physician AND Surgeon))*" be an access policy. Only a director or surgeon physician can access content encrypted with such a policy. In CP-ABE, an attribute authority is responsible for generating secret keys to users with attributes that describe their profile. To this end, the authority employs a master key known only to itself.

B. BLOCKCHAIN

Blockchain is a disruptive data structure maintained by a decentralized network with several nodes that process and store transactions [9]. A transaction consists of either a resource registration to an address or a resource transfer between addresses. An address refers to a user's public key from a digital signature system [20], which is associated to a private key, forming a key pair. Thus, to be able to issue transactions, a user should generate a signing key pair. A transaction is issued by signing it using a private key, and, in case of a transfer, indicating the destination address. The transaction validity is checked by verifying the transaction signature with the respective public key.

Once a transaction is submitted to a blockchain network, designated nodes insert it into part of the data structure, namely a block, which contains a number of gathered transactions. After constructing a block, a node attempts to link it to the last formed block, a task that gradually constructs a chain of blocks. To link blocks, nodes run a consensus protocol that allows nodes to agree on a valid up-to-date chain, which usually consists of the longest chain constructed. After linking a new block to the end of the chain, a node informs the new generated chain to every other node. By running the consensus protocol, the nodes can agree on whether officially adding the block to the chain or not. In addition, they only accept a chain as valid after verifying the correctness of links between existing blocks.

The consensus protocol adopted depends on the blockchain type that, in general, can be public or private [21]. A public blockchain is a permissionless network having no access control for nodes and participants, thus any uncertified, untrustworthy node or participant can read and record transactions, in addition to contributing for the task of inserting blocks into the chain. In turn, a private blockchain is a permissioned network that presents access control for participants and nodes, thus only permitted nodes can make part of the block insertion task into the chain. In Sec-Health, we employ a private blockchain, since only authorized healthcare organizations (e.g., hospitals) may be part of the system.

Compared to a traditional distributed database, blockchain presents significant advantages [22]. First, it is more resilient and scalable. In a blockchain network, there is no hierarchy among nodes, different from traditional architectures based on a main node that introduces performance bottlenecks,

single points of failure, and, in case of multiple main nodes, synchronization problems. Second, blockchain provides the materialization of smart contracts, which consist of custom and verifiable pieces of source code stored in the blockchain itself. When triggered, a smart contract may modify both its data and the state of the blockchain. Furthermore, it is estimated that modifying a block of a public blockchain is not computationally feasible if most of the network's CPU power belongs to honest nodes. This assumption ensures the blockchain integrity.

C. InterPlanetary FILE SYSTEM

InterPlanetary File System (IPFS) is a technology that affords decentralized networks in which all participating nodes use the same file system [10]. Data stored in an IPFS network is locally maintained by several nodes, removing the presence of single points of failure. Each stored file is addressed by its hash value, thus each file is unique in the system and the process of searching for files becomes more efficient. Retrieving a file requires performing a request using its hash. The nearest IPFS node having the file will provide it.

Since IPFS represents a decentralized network, it is employed in our work to allow for large-scale storage of health records while separating the storage of records (persisted in the IPFS network) from the storage of data required to verify record integrity (published on the blockchain). This enables a participating entity (e.g., a healthcare organization or a user) to opt for storing only small amounts of verification data (running a blockchain node) while leaving the task of storing health data (i.e., run an IPFS node) to entities that have more computing resources.

We adopt IPFS to store health records out of the blockchain as it was designed to be a decentralized WEB that exchange any type of data using well-known peer-to-peer protocols. Thus, it can share larger amounts of data, such as health records, in small or larger healthcare networks. Also, in IPFS, users can obtain distinct pieces of a health record from different nodes, improving efficiency, a requirement for some healthcare scenarios, such as emergency access.

IV. SEC-HEALTH: A NEW BLOCKCHAIN-BASED PROTOCOL FOR SECURING HEALTH RECORDS

As we discussed, proposals for secure storage and sharing of health records are limited, as they do not address the main properties established in current regulations, described in Section II. In order to work around such limitations, we now present Sec-Health, a new blockchain-based protocol that enables a storage and sharing system that meets health records properties.

A. PARTICIPANTS, ASSUMPTIONS, AND THREAT MODEL

Sec-Health comprises a set of participants, as follows. An authority is responsible for managing users registration and attesting that cryptographic material is valid, while a blockchain and an IPFS network store data related to health records. Several participants are considered the system users, as follows. A collaborator (e.g., physician, nurse) generates health records for patients and accesses, under patient consent, records generated by other collaborators. Research entities access anonymized health records to conduct researches. Attendants are call center professionals who handle initial steps of emergency sessions according to required information received from a phone caller. Finally, emergency servers are participants that assist collaborators to legitimately gain access to health records in emergency sessions.

We consider the following assumptions regarding Sec-Health operations. All employed hardware is secure, therefore hardware tampering is not possible. Furthermore, physical security of computing devices is guaranteed. We assume participants exchange messages over secure communication channels (e.g., using TLS). Malicious parties can intercept exchanged messages, but they do not launch denial-of-service attacks, since these are out of our scope. In addition, cryptographic primitives are semantically secure. For simplicity, we will consider a single authority while describing our proposal. However, we stress that the authority does not need to be trustworthy, since its services can be distributed [5].

As threat model, we assume that the protocol's participants and third parties may be attackers. An attacker uses its own resources to attack the system, without collusion. The attacker may attempt to read a health record without authorization, modify a record while proving the altered version is legitimate, or completely delete a record without authorization. Also, an attacker may try to access a health record after its access rights for such record have been revoked. Finally, an attacker may attempt to break the anonymity of a record provided for research purposes. We consider that a collaborator who legitimately gains access to a health record does not disclose information on it to other parties. Sec-Health's goal is to address the aforementioned malicious behaviors by detecting, preventing and/or mitigating them.

B. OVERVIEW

To ease the understanding of Sec-Health, we divide it into phases and complementary mechanisms. The set of four phases is composed of setup, storage, sharing, and emergency access. They include mechanisms that satisfy four health records properties, namely confidentiality, access control, integrity, and emergency access. In turn, the set of three complementary mechanisms fulfill three other properties: access revocation, anonymity, and interoperability. Figure 1 depicts an overview of Sec-Health with the participants that interact with each other in each phase and complementary mechanism. An overview explanation of the phases and mechanisms are given in the following.

The protocol begins with the setup phase, in which blockchain and IPFS networks are created, and users are registered in the system. The authority attests all the public data (e.g., participants' public key) necessary to run the

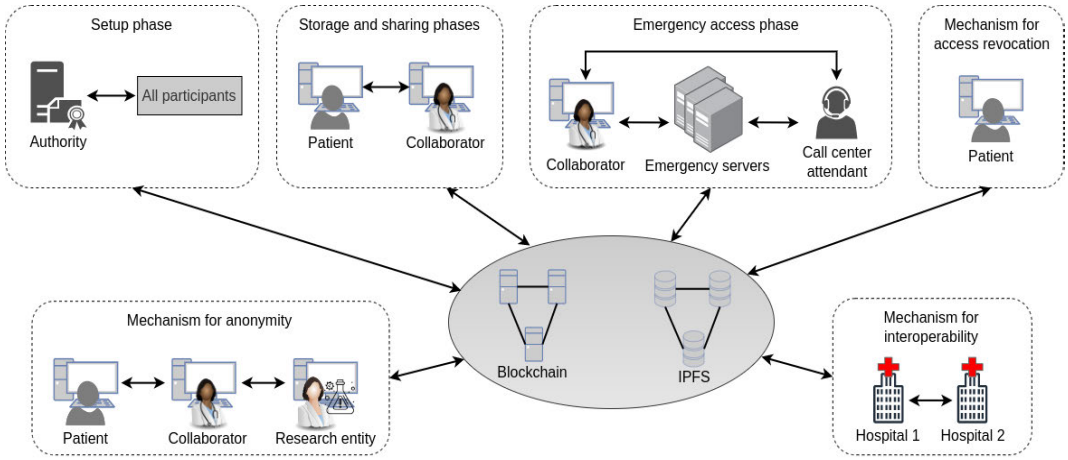


FIGURE 1. Overview of Sec-Health with its phases, complementary mechanisms and participants' interactions.

system, which are stored in the blockchain. In the storage phase, a collaborator generates a health record for a patient, who stores it in the IPFS network and its metadata in the blockchain in a secure fashion such that another collaborator can later access the record. After this, the designated collaborator can access the health record, in the sharing phase, if she owns the required cryptographic material for this purpose. The emergency access phase occurs when a patient needs emergency medical care and has no physical conditions (e.g., may be unconscious) to give available collaborators access permission to her health record. This phase runs in a collaborative way (between the emergency servers, collaborators and call center attendant) such that the patient can later verify who participated in the emergency session and had access to her health record.

After granting access to a health record for a collaborator, the patient can revoke this access right whenever she wants by means of the access revocation mechanism in an interaction with the blockchain network. With the anonymity mechanism, a collaborator can provide a health record for research studies under patient consent. The research entity responsible for the studies can access the record through data stored in the blockchain and IPFS. The patient may deny having given permission for disclosing her health record for research, while that the collaborator is able to prove when the patient is lying about this. Finally, the interoperability mechanism allows several healthcare organizations (e.g., hospitals) to securely exchange data, enabling patients to change organization whenever desired. In the next sections, we detail the design of every phase and complementary mechanism of Sec-Health. For each of them, we describe the operations performed by the protocol's participants and how they exchange data to achieve their objectives.

C. SETUP PHASE

In this phase, the protocol's participants are set up with their corresponding cryptographic material, used to securely run the operations of the protocol. First, the authority is created by the healthcare organization(s) that will physically allocate the computer(s) that will run the authority role. Recall that the authority attests the validity of cryptographic keys and play the role of CP-ABE attribute authority. A number of CP-ABE attributes (e.g., *is_physician*, *is_nurse*, *medical_specialty*), managed by the authority are established, wherein there may be attributes that refer to a single healthcare organization or several ones at the same time. The authority also generates the master key to create CP-ABE secret keys for the system users. Next, several healthcare organizations (e.g., hospitals) cooperate to initialize an IPFS network and a blockchain network. Each organization runs at least one node that may be part of both IPFS and blockchain networks. New nodes may enter the networks later on.

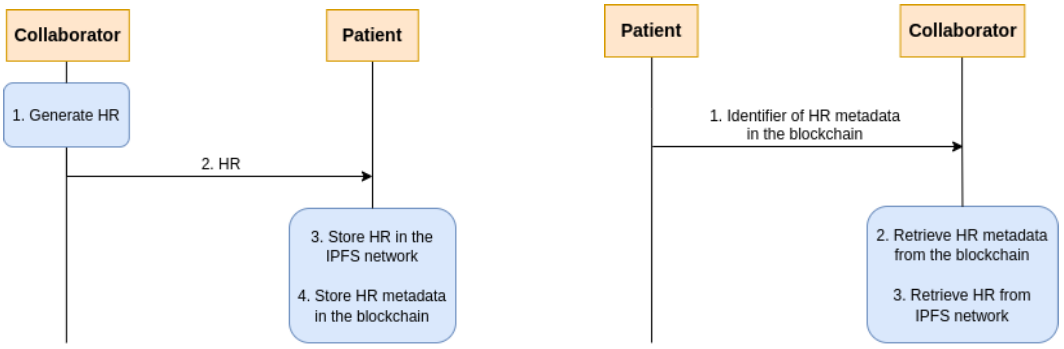
The registration of system users (patients, collaborators, call center attendants and research entities) follows common steps. Each of them generates an identifier and an encryption key pair to herself. In addition, the authority issues a CP-ABE secret key to the users that will access health records (patients, collaborators and research entities). Emergency servers follow a slightly different registration process in which they cooperatively generate a key pair from a threshold cryptosystem, wherein each server owns a share of the private key and an identifier. The identifiers and public keys of all participants are attested by the authority through the use of digital signatures and stored in the blockchain so that any interested party in the system may access them. A summary of the participants' material is shown in Table 1.

D. STORAGE AND SHARING PHASES

After the system is set up, the storage phase, which is illustrated in Figure 2a, takes place. Here, a collaborator can create a health record for a patient (step 1) and send it to her (step 2). Once a patient has a health record (HR), she may want to share it with a collaborator. To this end, she has to store it in the system so that the intended collaborator can access it in the future. First, the patient encrypts the HR with

TABLE 1. Cryptographic material of the protocol's participants.

Participant	Cryptographic material
Patient	Identifier $I(\mathsf{pat})$, public key $\mathsf{pk}_{\mathsf{pat}}$, private key $\mathsf{sk}_{\mathsf{pat}}$, and CP-ABE secret key $\mathsf{ck}_{\mathsf{pat}}$.
Collaborator	Identifier $I(\mathsf{col})$, public key $\mathsf{pk}_{\mathsf{col}}$, private key $\mathsf{sk}_{\mathsf{col}}$, and CP-ABE secret key $\mathsf{ck}_{\mathsf{col}}$.
Research entity	Identifier $I(\mathsf{re})$, public key $\mathsf{pk}_{\mathsf{re}}$, private key $\mathsf{sk}_{\mathsf{re}}$, and CP-ABE secret key $\mathsf{ck}_{\mathsf{re}}$.
Call center attendant	Identifier $I(\mathsf{att})$, public key $\mathsf{pk}_{\mathsf{att}}$, and private key $\mathsf{sk}_{\mathsf{att}}$.
Emergency server	Each server owns an identifier $I(\mathsf{ser}_i)$, and a private key share $\mathsf{sk}_{\mathsf{ser}_i}$. The public key $\mathsf{pk}_{\mathsf{ser}}$ belongs to all servers.



(a) Flowchart of Sec-Health storage phase.

(b) Flowchart of Sec-Health sharing phase.

FIGURE 2. Storage and sharing phases of Sec-Health.

CP-ABE and stores the resulting ciphertext $E(\mathsf{HR}, \mathsf{policy})$ in the IPFS network (step 3), which returns the URL to retrieve the ciphertext from the network. Afterwards, she computes the hash value $H(\mathsf{HR})$ over HR and constructs a transaction $\mathsf{TX}_{\mathsf{pat}} = \langle \mathsf{URL}, H(\mathsf{HR}) \rangle$ with information (metadata) related to HR. Next, the patient encrypts $\mathsf{TX}_{\mathsf{pat}}$ using the public key of the collaborator to whom she is willing to grant access to her health record. The resulting ciphertext $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{col}})$ is then stored in the blockchain (step 4), which returns an identifier $I(\mathsf{TX}_{\mathsf{pat}})$ for this transaction. As described in the next phase, the patient informs this identifier to the collaborator who will access the health record.

In the sharing phase, a collaborator requests access to a patient health record, as shown in Figure 2b. To allow the collaborator to create the request, the patient informs to her the identifier $I(\mathsf{TX}_{\mathsf{pat}})$ of the blockchain transaction $\mathsf{TX}_{\mathsf{pat}}$ containing the health record metadata (step 1). By having this identifier, the collaborator sends a request to retrieve $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{col}})$ from the blockchain and, then, uses her private key to compute the decryption $D(E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{col}}), \mathsf{sk}_{\mathsf{col}})$, which returns $\mathsf{TX}_{\mathsf{pat}} = \langle \mathsf{URL}, H(\mathsf{HR}) \rangle$ (step 2). Next, she employs URL to retrieve ciphertext $E(\mathsf{HR}, \mathsf{policy})$ from the IPFS network (step 3). Then, the collaborator uses her CP-ABE secret key $\mathsf{ck}_{\mathsf{col}}$ to attempt to decrypt $E(\mathsf{HR}, \mathsf{policy})$ by computing $D(E(\mathsf{HR}, \mathsf{policy}), \mathsf{ck}_{\mathsf{col}})$. If the decryption does not succeed, it means the collaborator does not own required attributes to access the patient health record. Otherwise, she obtains the record of interest HR in plaintext. The sharing phase ends with the collaborator verifying the health record integrity. For this, she first computes the hash value of HR, obtained from the IPFS network. Here, we denote it by $H(\mathsf{HR})'$. After this, she extracts $H(\mathsf{HR})$ from $\mathsf{TX}_{\mathsf{pat}}$ and verifies whether $H(\mathsf{HR}) = H(\mathsf{HR})'$. If this equality holds, then the HR obtained from the IPFS network is the same health record the patient sent to the network in the storage phase, which means the health record integrity was kept.

E. EMERGENCY ACCESS PHASE

Sec-Health fulfills the emergency access requirement of regulations with a collaborative protocol, as depicted in Figure 3. A patient executes an emergency setup to make her health record HR available for emergency cases. Before the patient executes the emergency setup, we assume she has already stored the corresponding CP-ABE ciphertext $E(\mathsf{HR}, \mathsf{policy})$ in the IPFS network and has the associated URL to access the ciphertext. With this, the patient can run the emergency setup by creating a transaction $\mathsf{TX}_{\mathsf{pat}} = \langle \mathsf{URL}, H(\mathsf{HR}) \rangle$ and encrypting it with the emergency servers' public key, resulting in ciphertext $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}})$. Then, she stores $\langle I(\mathsf{pat}), E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}}) \rangle$ in the blockchain (step 1), where $I(\mathsf{pat})$ is the patient identifier, which returns the identifier $I(\mathsf{TX}_{\mathsf{pat}})$ for this transaction.

Once emergency setup occurs, emergency access sessions take place. A session is initiated by a caller (e.g., patient's relative), who contacts a call center attendant by phone requesting emergency medical care to a patient. The caller informs to the attendant patient information, including her name and emergency situation details, which we denote by ED. The attendant proceeds with the session running a blockchain function to find the patient identifier $I(\mathsf{pat})$ by name. Next, she creates a transaction $\mathsf{TX}_{\mathsf{att}} = \langle I(\mathsf{pat}), \mathsf{ED}, I(\mathsf{col}_1) \rangle$, where $I(\mathsf{col}_1)$ is the identifier of the

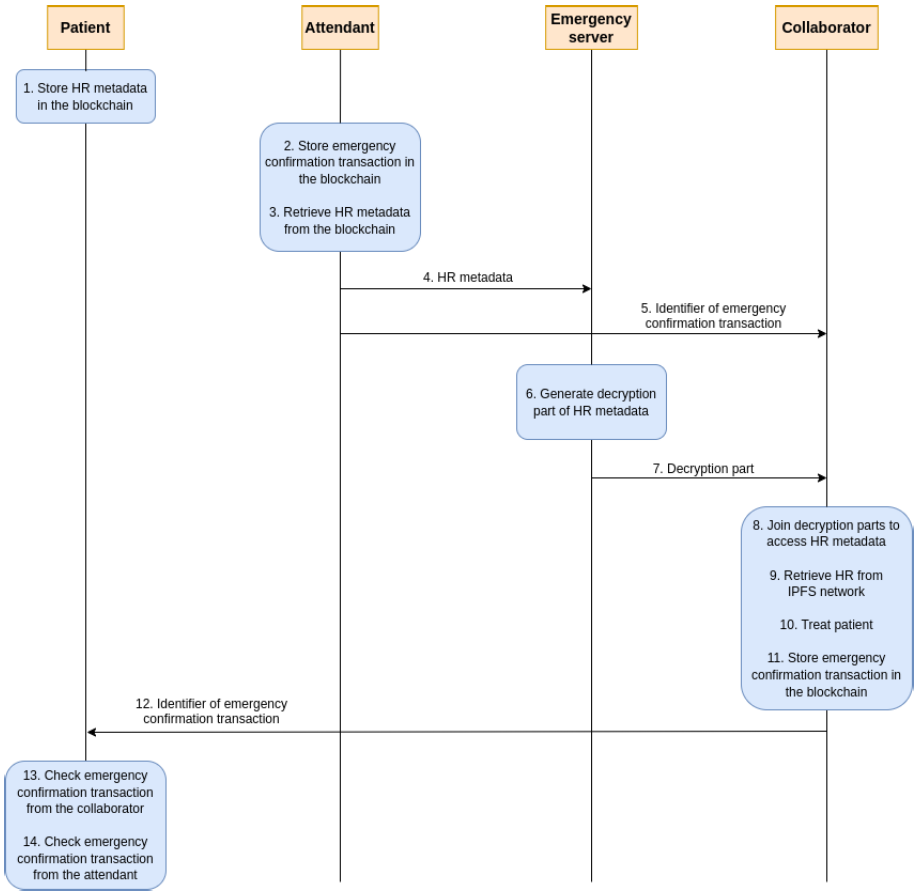


FIGURE 3. Flowchart of the the process to perform emergency setup, emergency access and emergency session audit.

ambulance's collaborator who will cover the emergency. Then, the attendant encrypts $\mathsf{TX}_{\mathsf{att}}$ with the patient's public key $\mathsf{pk}_{\mathsf{pat}}$, resulting in ciphertext $E(\mathsf{TX}_{\mathsf{att}}, \mathsf{pk}_{\mathsf{pat}})$, which she stores in the blockchain (step 2). This transaction, which has identifier $I(\mathsf{TX}_{\mathsf{att}})$, is called an emergency confirmation transaction and will provide transparency for the patient regarding the participants and contents of the emergency session, as we will see.

After publishing her emergency confirmation transaction, the attendant uses $I(\mathsf{pat})$ in a blockchain function to retrieve $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}})$ from the blockchain (step 3). Next, she sends two data tuples. The first is sent to every emergency server (step 4), comprising $\langle I(\mathsf{col}_1), E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}})\rangle$. The second, composed of $I(\mathsf{TX}_{\mathsf{att}})$, is sent to the ambulance's collaborator (step 5) whose identifier is $I(\mathsf{col}_1)$. Thereafter, the emergency servers proceed with the session. Each of them performs a decryption operation of $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}})$ using its secret key share $\mathsf{sk}_{\mathsf{ser}_i}$ from a threshold cryptosystem (step 6), resulting in a decryption part $D(E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}}), \mathsf{sk}_{\mathsf{ser}_i})$, which it sends to the collaborator col_1 (step 7).

Upon receiving decryption parts $D(E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}}), \mathsf{sk}_{\mathsf{ser}_i})$ from the emergency servers, the collaborator col_1 joins the decryption parts (step 8) to finally decrypt $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{ser}})$ and obtains $\mathsf{TX}_{\mathsf{pat}} = \langle \mathsf{URL}, H(\mathsf{HR})\rangle$. After this, she executes the steps of the sharing phase (see Section IV-D) to securely access the health record HR (step 9). After using the record to treat the patient (step 10), the collaborator computes her emergency confirmation transaction $\mathsf{TX}_{\mathsf{col}_1} = \langle I(\mathsf{TX}_{\mathsf{att}}), I(\mathsf{col}_2)\rangle$, where $I(\mathsf{col}_2)$ is the identifier of the hospital's collaborator who will proceed with the patient treatment. Then, the collaborator col_1 encrypts $\mathsf{TX}_{\mathsf{col}_1}$ with the patient's public key $\mathsf{pk}_{\mathsf{pat}}$, resulting in ciphertext $E(\mathsf{TX}_{\mathsf{col}_1}, \mathsf{pk}_{\mathsf{pat}})$. This ciphertext is stored in the blockchain (step 11), which returns transaction identifier $I(\mathsf{TX}_{\mathsf{col}_1})$. She finally interacts with emergency servers and collaborator col_2 the same way as the attendant did to allow the hospital's collaborator col_2 to proceed with the session.

Once an emergency session is finished, the patient may audit it to verify the participants and contents of the session. The identifier of the last emergency confirmation transaction stored in the blockchain is the information required to audit a session. Let $I(\mathsf{TX}_{\mathsf{col}_1})$ be the transaction identifier of the single collaborator col_1 who treated the patient in a session. To audit the session, the patient receives $I(\mathsf{TX}_{\mathsf{col}_1})$ from collaborator col_1 (step 12), employs it to retrieve

$E(\mathsf{TX}_{\mathsf{col}_1}, \mathsf{pk}_{\mathsf{pat}})$ from the blockchain and uses $\mathsf{sk}_{\mathsf{pat}}$ to compute $D(E(\mathsf{TX}_{\mathsf{col}_1}, \mathsf{pk}_{\mathsf{pat}}), \mathsf{sk}_{\mathsf{pat}})$ and to obtain transaction $\mathsf{TX}_{\mathsf{col}_1} = \langle I(\mathsf{TX}_{\mathsf{att}})\rangle$. With this, she has verified that collaborator col_1 in fact participated in the emergency session and accessed her health record (step 13). After this, she extracts $I(\mathsf{TX}_{\mathsf{att}})$ from $\mathsf{TX}_{\mathsf{col}_1}$, uses it to obtain $E(\mathsf{TX}_{\mathsf{att}}, \mathsf{pk}_{\mathsf{pat}})$ from the blockchain and employs $\mathsf{sk}_{\mathsf{pat}}$ to compute $D(E(\mathsf{TX}_{\mathsf{att}}, \mathsf{pk}_{\mathsf{pat}}), \mathsf{sk}_{\mathsf{pat}})$ and to obtain transaction $\mathsf{TX}_{\mathsf{att}} = \langle I(\mathsf{pat}), \mathsf{ED}, I(\mathsf{col}_1)\rangle$. With this transaction, the patient can verify the attendant who participated in the emergency session and the contents of ED (step 14). Also, she can check that the attendant explicitly indicated who would be the collaborator to treat the patient. This approach provides full transparency of the session for the patient.

F. MECHANISM FOR ANONYMITY

In Sec-Health, health records are anonymized before being provided for researches. A patient record can only be provided for research by the collaborator col who generated the record under patient consent. Before the process of sharing a health record with a research entity is executed, we assume the collaborator has already generated a record to a patient pat, who afterwards stored the encrypted record in the IPFS network. Also, we assume the patient already stored a transaction in the blockchain granting access to the record for the collaborator, who, therefore, knows all the record metadata: $\langle \mathsf{URL}, H(\mathsf{HR})\rangle$. These assumptions are satisfied by executing the storage phase steps (see Section IV-D).

The process of sharing the HR with a research entity occurs according to the workflow of Figure 4. Here, the patient pat creates a proof $\mathsf{PR} = \mathsf{HR} \parallel r$ that consists of concatenating the HR with a generated random value r (step 1). This proof represents patient consent with respect to making her health record available for research purposes. Next, she digitally signs PR and sends the resulting signature $S(\mathsf{PR})$ along with r to the collaborator col (step 2). Upon receiving $\langle S(\mathsf{PR}), r\rangle$ from the patient, the collaborator starts the creation of a blockchain transaction that will allow the research entity to access the HR without learning its corresponding patient. To this end, the collaborator first computes the hash value $H(S(\mathsf{PR}))$ over $S(\mathsf{PR})$ to hide the identifier of the patient who created $S(\mathsf{PR})$ (step 3). Now, recall that the collaborator knows $\langle \mathsf{URL}, H(\mathsf{HR})\rangle$. Then, she constructs a transaction $\mathsf{TX}_{\mathsf{col}} = \langle \mathsf{URL}, H(\mathsf{HR})\rangle$ and encrypts it with the research entity's public key $\mathsf{pk}_{\mathsf{re}}$, resulting in ciphertext $E(\mathsf{TX}_{\mathsf{col}}, \mathsf{pk}_{\mathsf{re}})$. Afterwards, she stores $\langle E(\mathsf{TX}_{\mathsf{col}}, \mathsf{pk}_{\mathsf{re}}), H(S(\mathsf{PR}))\rangle$ in the blockchain (step 4), which returns the transaction identifier $I(\mathsf{TX}_{\mathsf{col}})$. After this, the collaborator informs $I(\mathsf{TX}_{\mathsf{col}})$ to the research entity (step 5) that will be able to access the HR (step 6) by executing the steps of the sharing phase (see Section IV-D).

Suppose now the patient claims she did not allow the collaborator to provide the HR for research. The collaborator can prove that the patient is lying. First, the collaborator discloses tuple $\langle \mathsf{HR}, r, S(\mathsf{PR}), I(\mathsf{TX}_{\mathsf{col}})\rangle$ to a verifier (step 7). To check that the patient generated the required proof (step 8), the verifier performs as follows. She reconstructs the proof $\mathsf{PR} = \mathsf{HR} \parallel r$. Next, she uses $I(\mathsf{TX}_{\mathsf{col}})$ to retrieve $\langle E(\mathsf{TX}_{\mathsf{col}}, \mathsf{pk}_{\mathsf{re}}), H(S(\mathsf{PR}))\rangle$ from the blockchain. The verifier then employs $S(\mathsf{PR})$ and PR to check whether $H(S(\mathsf{PR}))$ is the hash of a valid patient signature over PR indeed. The signature validity proves that the patient allowed the collaborator to provide the record for research. If the signature is not valid, then the collaborator stored a false signature in the blockchain.

G. MECHANISMS FOR ACCESS REVOCATION AND INTEROPERABILITY

To revoke access rights from a collaborator col to access a health record, the patient, owner of the record, requests deletion of the blockchain transaction $E(\mathsf{TX}_{\mathsf{pat}}, \mathsf{pk}_{\mathsf{col}})$, encrypted with the collaborator's public key. This prevents the collaborator from obtaining the transaction $\mathsf{TX}_{\mathsf{pat}} = \langle \mathsf{URL}, H(\mathsf{HR})\rangle$, which contains the URL required to retrieve the record from the IPFS network. In Sec-Health, we approach this mechanism by means of a functionality that current popular blockchains, such as Hyperledger Fabric [23], provide, consisting of separating the actual private data of a transaction from the hash value of the transaction. Thus, deleted data comprise only the health record metadata $\langle \mathsf{URL}, H(\mathsf{HR})\rangle$, while the transaction's hash, required to verify the chain integrity, is maintained by blockchain nodes. To delete the private data of her transaction, the patient must take a single action: inform the corresponding transaction identifier $I(\mathsf{TX}_{\mathsf{pat}})$ to the blockchain network, which will be responsible for the deletion. Therefore, we stress that revoking access does not require performing cryptographic operations. In addition, revoking access from one collaborator does not interfere on the access granted for other collaborators.

Interoperability allows for the utilization of a health record for patient care in several healthcare institutions (e.g., multiple hospitals) with patient consent and through a system that enables secure record transfer. Most of the times, interoperability between various institutions is achieved by means of a centralized entity which is responsible for enforcing policies that rule access to records. Such an entity is subject to several security threats, as it represents a single point of failure. In Sec-Health, applying access policies in the encrypted record itself enables a distinct approach for interoperability, wherein employing a centralized entity to control access to records is not necessary. Consequently, records can be stored in different physical locations without critical risks related to unauthorized access. In this context, Sec-Health affords secure interoperability through the participation of healthcare institutions in the blockchain and IPFS networks, where each institution is responsible for managing and running at least one node in each network. Given the high degree of sensitivity related to health records, private versions of those networks must be employed. With this, only authorized institutions and users will be able to participate in the networks and gain access to the data. Although all institutions may have access to encrypted records and

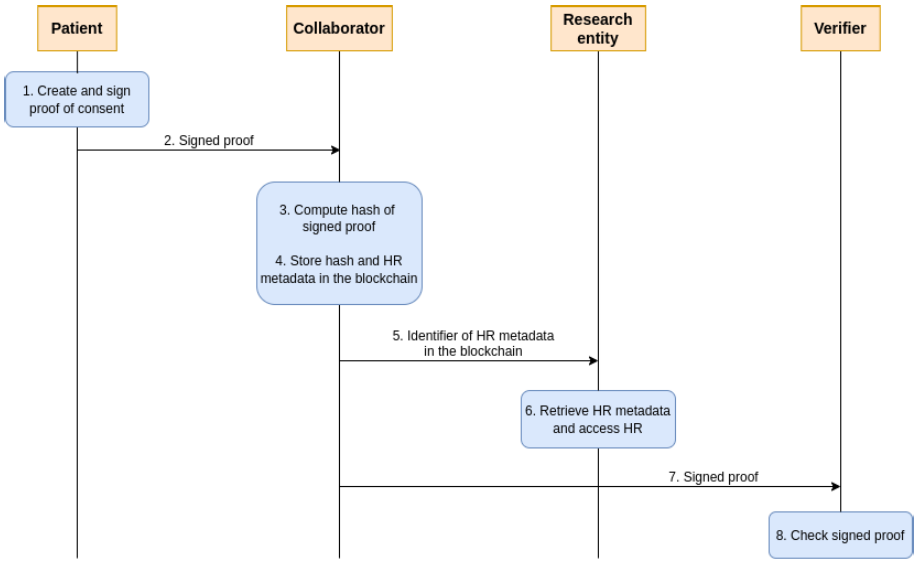


FIGURE 4. Flowchart of the process to set up and access an anonymized record, and to verify the patient has given consent to disclose her record for research.

transactions, only authorized users, with appropriate private and CP-ABE keys, will gain access to the data in plaintext.

V. RELATED WORK

The literature presents several proposals that aim to ensure health records security. However, there is a lack of proposals that approach the main health records properties. Instead, they address only a subset of the concerns [2]. In general, these solutions employ centralized approaches (e.g., based on clouds) or decentralized ones (e.g., blockchain-based). To delineate a state-of-the-art overview, we now present security proposals for health records.

Ganiga et al. [6] elaborated a healthcare security framework for cloud-based systems, which addresses confidentiality and access control by employing attribute-based encryption. However, it is assumed that the cloud will behave honestly by following the permissions that a patient assigns for collaborators. Masud et al. [24] provided a secure lightweight scheme to access health records in cloud-based systems. The security of the system relies mainly on master keys, employed by hospital's administrators to generate sub-keys to patients, adopted to secure communications and health records. Thus, compromising the cloud or the administrator of a hospital is sufficient to break the system. Ganiga et al. and Masud et al. do not address properties such as emergency access and access revocation.

The protocol for mobile devices based on CP-ABE due to Liu et al. [25] generates the largest part of a health record ciphertext in offline mode. Thereby, an online device saves battery by running lightweight cryptographic operations. The authors, however, assume a trustworthy cloud and do not address properties like integrity and emergency access. Kumar and Chand [26] designed a healthcare information system to address the vulnerabilities of a trusted central cloud provider. Their system employs a novel identity-based encryption scheme to secure health data collected from sensors implanted on the patient's body. The drawback is that unauthorized deletion of health records from the cloud is not addressed. Moreover, some properties are not satisfied, such as interoperability and emergency access.

The main goal of Qiu et al. [27] was to secure health records even if both cloud servers and cryptographic keys, used for confidentiality and access control, are compromised. However, the authors do not address the properties of access revocation, emergency access and interoperability. The solution due to Varadharajan et al. [28] aims to preserve patient privacy with a proxy re-encryption mechanism that enables collaborators to temporarily access records. Revocation occurs automatically after a predefined time frame. Mhatre and Nimkar [5] introduced a federated clouds model to share health records among different domains. It fulfills several properties of health records, namely confidentiality, access control, revocation and interoperability, but does not address other key properties.

Several solutions employ blockchain. Abunadi and Ramasamy [29] proposed a blockchain framework in which patients share cryptographic keys with the collaborator such that she can gain access to health records. However, by having access to patients' cryptographic keys, a collaborator may easily leak them to other parties. Patel [7] proposed a framework to secure medical images that addresses only some of the health record properties (access control, integrity, interoperability, and anonymity) through running access control lists on blockchains. More complete protocols were proposed (e.g., [8], [30]). Shen et al. [8], for example, addressed the efficiency of securing health data with blockchain. However, the protocols still lack fulfillment of certain properties.

Rahulamathavan et al. [31] used CP-ABE and blockchain to secure the sharing of health records. However, because public blockchains are employed, the proposal is limited, since this type of blockchain does not support storage of large records or insert large records in the chain by charging high reward taxes. Zghaibeh et al. [32] provided another blockchain-based healthcare information system that heavily relies on smart contracts, which are responsible for registering the data related to patient treatment since the moment a patient visits a physician until the moment she receives results of medical exams. However, the system lacks solutions for several properties, namely emergency access, access revocation, anonymity and confidentiality, in case a malicious entity gains read permissions to a record.

The work due to da Costa et al. [12] employs CP-ABE and blockchain to fulfill the health record properties of confidentiality, access control, and integrity, preventing unauthorized modification with blockchain and unauthorized deletion storing records in decentralized networks. However, in addition to presenting security problems regarding the provided mechanisms (e.g., the use of a single symmetric key for several collaborators to access a record), their solution lacks approaches to fulfill other health record properties, such as emergency access, anonymity, and interoperability. Thus, Sec-Health overcomes the existing problems of this work.

The most common aspect of the related work is the focus on basic properties (confidentiality, access control, and integrity). Also, all presented solutions lack mechanism to securely provide emergency access. Sec-Health differs from related work by addressing the challenging issue of fulfilling all the properties defined in Section II. Our blockchain-based proposal has decentralized mechanisms that give patients control over the records, and prevent a single malicious entity from compromising the system. This is highly advantageous over proposals based on centralized servers. Compared to other blockchain-based solutions, Sec-Health goes a step further by covering basic properties (confidentiality, access control, and integrity), and more challenging ones (e.g., emergency access, revocation, and anonymity).

VI. EVALUATION

In this section, we evaluate Sec-Health by first analyzing its security, i.e., describing how it protects health records while fulfilling the properties defined in Section II. We compare Sec-Health with related work based on the properties fulfillment (a summary is presented in Table 2), detailing our contributions. Second, we present an experimental evaluation of Sec-Health, comparing our results with related work and showing the practical feasibility of our protocol.

A. ANALYZING AND COMPARING SEC-HEALTH WITH RELATED WORK

1) CONFIDENTIALITY AND ACCESS CONTROL

Although some proposals adopt CP-ABE to fulfill confidentiality and access control (e.g., [5], [6], [25]), they use centralized servers to store health records and evaluate access permissions. We instead provide a two-layer security, wherein the first layer has a decentralized network, with no single points of failure, which stores records metadata encrypted with public keys, and the second consists of records encrypted with CP-ABE. Rahulamathavan et al. [31] use CP-ABE and blockchain to secure health records. However, they consider the storage of records, encrypted with CP-ABE, in a public blockchain, which can be expensive. We, instead, adopt a private blockchain that requires no rewards for nodes. Sec-Health uses an approach similar to the one due to da Costa et al. [12]. However, the latter adopts symmetric keys to encrypt a blockchain transaction, which can be easily leaked and are not associated top any collaborator. Other proposals (e.g., [7], [32]) lack discussions on how confidentiality should be achieved.

2) INTEGRITY

Several proposals lack schemes to address integrity of health records [5], [25], [28]. Other solutions (e.g., [6]) adopt a centralized cloud assuming that it will never alter or delete health records. To protect against unauthorized alteration, we store the record metadata (e.g., its hash value) in a blockchain, which is very hard to tamper with. If an attacker tries to change the hash value of a health record, she will need to change all the blocks of the chain added after the block that contains the hash value. By knowing the true chain, the network nodes can easily detect such an attack. Other protocols (e.g., [7], [8]) also address unauthorized alteration by employing blockchain. However, they propose storing records off-chain in a centralized server, thus not addressing unauthorized record deletion. We overcome this issue by storing encrypted health records in a decentralized network (IPFS). Rahulamathavan et al. [31] and da Costa et al. [12] propose a public blockchain for providing integrity, which is not suitable for healthcare scenarios.

3) EMERGENCY ACCESS

In Sec-Health, emergency access is permitted by a set of emergency servers within a distributed approach, wherein each of them provides a decryption part of the transaction containing the health record metadata. Therefore, an attacker attempting to maliciously access a record in our scheme would need to corrupt the majority of the servers in order to obtain a sufficient number of decryption parts. Sec-Health also affords collaboration between the emergency session participants, since each of them explicitly indicates who will be the next participant in the session through emergency confirmation transactions. This collaboration provides transparency for the patient, who can verify, once the session is ended, who had access to her health record through the transactions. The related proposals do not address emergency access. In general, this property is addressed individually or along with basic properties (i.e., confidentiality and access control), such as in the work due to Oliveira et al. [33].

TABLE 2. Summary of the solutions and their fulfilled properties, wherein PR1 is confidentiality, PR2 is access control, PR3 is integrity, PR4 is emergency access, PR5 is access revocation, PR6 is interoperability, and PR7 is anonymity.

Proposal	PR1	PR2	PR3	PR4	PR5	PR6	PR7
Ganiga et al. [6]	✓	✓	✓	✗	✗	✓	✗
Masud et al. [24]	✓	✓	✓	✗	✗	✗	✓
Liu et al. [25]	✓	✓	✗	✗	✗	✗	✗
Kumar and Chand [26]	✓	✓	✓	✗	✓	✗	✓
Qiu et al. [27]	✓	✓	✓	✗	✗	✗	✓
Varadharajan et al. [28]	✓	✓	✗	✗	✓	✗	✗
Mhatre and Nimkar [5]	✓	✓	✗	✗	✓	✓	✗
Abunadi and Ramasamy [29]	✓	✓	✓	✗	✗	✓	✗
Patel [7]	✗	✓	✓	✗	✗	✓	✓
Shen et al. [8]	✓	✓	✓	✗	✓	✓	✗
Zhuang et al. [30]	✓	✓	✓	✗	✗	✓	✗
Rahulamathavan et al. [31]	✓	✓	✓	✗	✗	✗	✓
Zghaibeh et al. [32]	✗	✓	✓	✗	✗	✓	✗
da Costa et al. [12]	✓	✓	✓	✗	✓	✗	✗
Sec-Health	✓	✓	✓	✓	✓	✓	✓

4) ACCESS REVOCATION

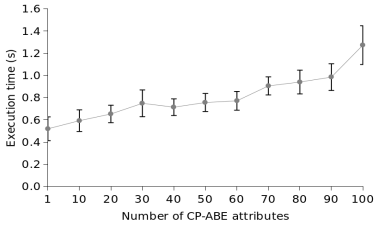
Access revocation is addressed by various literature proposals. Some of them update the health record ciphertext or the users secret key to apply revocation [5], [12], [26], thus not satisfying the requirements of access revocation defined in Section II. Varadharajan et al. [28] apply revocation to records automatically after a certain time period. This is not a suitable solution as the patient needs to predict time periods in which she thinks collaborators may need access to her records. In addition, ciphertexts need to be frequently updated with new time periods once revocation takes place. There are also blockchain-based solutions (e.g., [8]) in which smart contracts are responsible for verifying and revoking the access permissions of every collaborator. Our solution is simpler as it requires no execution of transactions by smart contracts to perform revocation. In Sec-Health, the patient just needs to request the data deletion of a single transaction, which immediately revokes a single collaborator, does not require generating any new cryptographic material and does not interfere in access rights of other collaborators. Therefore, our scheme meets the requirements described in Section II.

5) INTEROPERABILITY

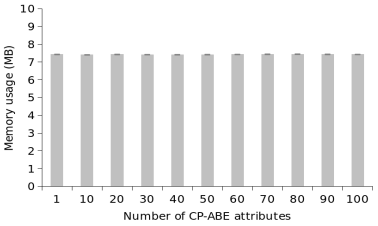
Sec-Health fulfills the property of interoperability by allowing a cooperation of healthcare organizations to jointly execute a system composed of blockchain and IPFS networks, wherein each organization can run a node within the networks. By running an IPFS node, the organization can obtain any patient record that has been generated by a collaborator within this cooperation of healthcare organizations. Note that this does not mean that any collaborator can access all health records. Since every health record is encrypted with a CP-ABE access policy elaborated by the patient, only authorized collaborators will have access to the patient record. The same applies to blockchain transactions, which are encrypted with public key cryptography. Also, we stress that an organization does not need to run a node in both networks. If the organization faces the problem of limited storage resources, it can choose to run only a blockchain node, which stores less amounts of data compared to the IPFS network. With the blockchain data, the organization is able to securely access records stored in the IPFS network. Sec-Health provides decentralized and secure interoperability, the same way as addressed by other blockchain-based solutions (e.g., [29], [30], [32]). By proposing a decentralized system, we overcome issues regarding single points of failure from cloud-based proposals (e.g., [24], [27]).

6) ANONYMITY

Sec-Health solution for providing anonymized health records for research protects the patient. First, a malicious collaborator cannot construct the proof to provide a record for research, since this requires a patient signature over the proof. Second, the patient elaborates the proof concatenating her record with a random value. If a malicious collaborator gains access to a proof that the patient generated to another collaborator, and uses it to illegitimately disclose the record to an arbitrary research entity, the patient can claim that this collaborator is malicious. Since the patient has not informed the random value concatenated with the record to the malicious collaborator, the latter will fail to open the proof to a verifier. Some proposals address anonymity in healthcare. Patel [7] and Rahulamathavan et al. [31] assign, in public blockchains, anonymous public keys to users such that



(a) Setup phase execution time by number of CP-ABE secret key attributes.



(b) Setup phase memory usage by number of CP-ABE secret key attributes.

FIGURE 5. Evaluation results of the setup phase.

the transaction creators (i.e., patients) cannot be identified. However, since healthcare systems need to be accessed only by authorized entities, public blockchains are not suitable. Although presenting solutions for anonymity, cloud-based schemes (e.g., [24], [27]) do not afford a concrete mechanism for providing anonymized health records for research entities with verification of patient consent, as Sec-Health does.

B. LIMITATIONS OF SEC-HEALTH

As discussed in Section VI-A, Sec-Health presents several advantages over related work. Nevertheless, the design of our protocol to secure health records also leads to some trade-offs that result in limitations. The first limitation is due to the fact that a single instance of a health record is stored in the IPFS network encrypted with CP-ABE. When looking to access this record, all collaborators will retrieve such instance from the network by using a single URL which is encrypted with public key and stored for each collaborator in the blockchain. Since a single URL is employed for all collaborators, an arbitrary collaborator may leak the URL without being identified. This issue could be mitigated by adopting a different URL for each collaborator, but this would require storing the same record several times in the IPFS network, which would significantly increase the storage overhead and the number of duplicated records.

Another limitation relates to encrypting health records with CP-ABE. Since attributes are used for encryption, CP-ABE requires patients to have a prior idea of the collaborators (or type of collaborators) who will access their records. There should exist cases in which a collaborator is authorized to access a health record by the patient but the collaborator's attributes do not satisfy the CP-ABE access policy used to encrypt the record. In such cases, the patient would need to encrypt the health record again including an access policy that matches the attributes of the corresponding collaborator. This would also result in increasing the storage overhead and the number of duplicated records in the IPFS network.

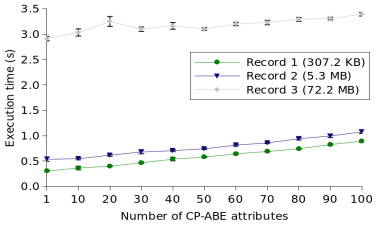
Sec-Health mechanism for access revocation fulfills the requirements described in Section II. However, blockchain nodes may be eventually compromised. Thus, after a patient requests the blockchain to revoke a collaborator and delete the corresponding transaction, malicious nodes may keep the transaction and return it to malicious collaborators when requested. This issue could also arise in the IPFS network in the case a patient requests the deletion of a health record. Such limitation of decentralized systems could be mitigated with a proper collaborative scheme that allows honest nodes to communicate to each other that malicious nodes are delivering unappropriated data. With this, the network would know when to remove a malicious node from the system. We leave the proposal of such a scheme as future work.

C. EXPERIMENTAL EVALUATION

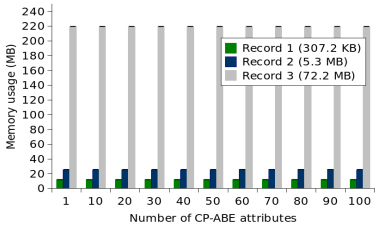
To show the feasibility of Sec-Health, we developed a PoC of the protocol. Cryptographic operations were implemented in C++. More specifically, we employed CP-ABE, RSA [17] for public key encryption (3072-bit keys), SHA-256 hash function [18], and Elliptic Curve Digital Signature Algorithm (ECDSA) [20] for digital signatures. Hyperledger Fabric was used to build a testing blockchain with 4 nodes. The testbed was built on top of a virtual machine with Ubuntu 18.04, Intel Core i7 processor and 2 GB of RAM.

The PoC includes three Sec-Health phases: setup, storage, and sharing. For each phase, we evaluated two metrics, execution time and client-side memory usage, to understand Sec-Health's performance and feasibility in a practical scenario. We also evaluated the storage overhead of CP-ABE ciphertexts to understand the relation between the sizes of ciphertext and plaintext health records. To evaluate execution time and memory usage, we did not consider the time for exchanging messages, as it may vary according to the bandwidth available to users. Aiming to evaluate the PoC under different health record sizes, our tests used: Record 1 with 307.2 KB; Record 2 containing 5.3 MB; and Record 3 with 72.2 MB. As CP-ABE is a crucial primitive, used for record encryption, the tests varied the number of attributes of CP-ABE access policies and secret keys. Each test scenario was executed 50 times, and metric values are the average among the executions, with 95% confidence interval.

Figure 5a depicts the setup phase execution times for the registration of a user (see Section IV-C). The execution times ranged between 0.52 s (with 1 attribute) and 1.27 s (with 100 attributes). The number of CP-ABE attributes did not cause significant increase in time. The RSA key pair generation was the main cause for time variation due to its random nature to find proper key generation parameters.

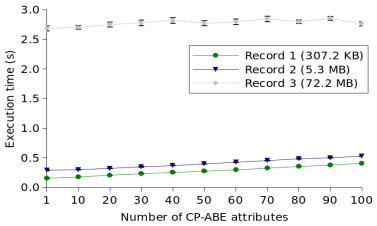


(a) Storage phase execution time by number of CP-ABE access policy attributes.

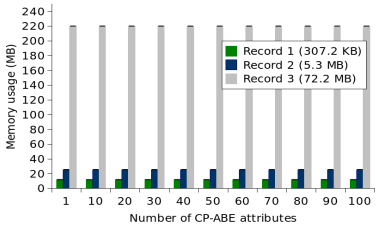


(b) Storage phase memory usage by number of CP-ABE access policy attributes.

FIGURE 6. Evaluation results of the storage phase.



(a) Sharing phase execution time by number of CP-ABE access policy attributes.



(b) Sharing phase memory usage by number of CP-ABE access policy attributes.

FIGURE 7. Evaluation results of the sharing phase.

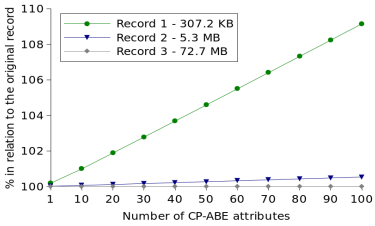


FIGURE 8. Size of the health record ciphertext (encrypted with CP-ABE) in relation to the size of the record in plaintext.

As shown in Figure 5b, there is no significant client-side memory usage when registering a user in Sec-Health as the amount of memory is generally near 7.4 MB. Sec-Health does not cause any additional memory overhead when increasing the number of attributes of a CP-ABE secret key.

For Record 1, the storage phase (see Section IV-D) presented efficient execution times, as depicted in Figure 6a. For instance, with a CP-ABE access policy with 100 attributes, the average time was 0.89 s. When Record 2 was used, a slight time increase occurred. For example, with Record 2 and access policies with 100 attributes, the average time was 1.07 s. Record 3 (the largest one) caused a higher impact on execution time, which ranged from 2.91 s and 3.39 s. This is mainly due to the execution of the hash function and CP-ABE. Digital signatures overhead on execution time was negligible. Figure 6b presents the storage phase memory usage. For Records 1 and 2, less than 30 MB were reserved in the user device. For the larger record (Record 3), more than 200 MB was used. We can observe that increasing the number of access policy attributes does not introduce extra overhead, thus more complex policies are feasible.

Figure 7a depicts execution times for the sharing phase. Compared to the storage phase, the sharing phase is more efficient. For example, with 100 attributes, the times of the latter phase with Records 1, 2 and 3 were 0.4, 0.52 and 2.76 s, respectively, while for the former phase, the times were 0.89, 1.07 and 3.39. Thus, accessing a health record is more efficient than storing it in the system. This is fundamental, since record access may occur in critical scenarios, such as emergency sessions. Also, the curves from Figure 7a show that the number of attributes does not considerably impact the sharing phase execution time. This characteristic is clearly notable with larger health records. Figure 7b illustrates memory usage of the sharing phase. The amount of memory used here is very similar to the amount consumed in the storage phase for all scenarios (see Figure 6b), i.e., there is no significant client-side memory overhead.

Figure 8 shows that in Sec-Health the CP-ABE ciphertext produces an increase in relation to its plaintext. Record 1 (307.2 KB) presented the highest percentages of size increase. For instance, with an access policy of 100 attributes, the encrypted record produced 9% of increase. This is a small addition of data (27.6 KB). For larger records (Records 2 and 3), the increase was smaller than 1% for all scenarios. Therefore, storing the encrypted health record in the IPFS network instead of the corresponding record in plaintext introduces no significant storage overhead.

D. COMPARING THE EXPERIMENTAL RESULTS WITH RELATED WORK

Most of the related work (see Section V) do not present load test results (e.g., execution time and memory usage). To perform a fair comparison between our results and those from related work, we consider proposals that were tested

TABLE 3. Execution time comparison between Sec-Health and related work using health records with sizes smaller than 1 MB.

	Storage phase		Sharing phase	
	10 attributes	100 attributes	10 attributes	100 attributes
Rahulamathavan et al. [31]	0.5 s	–	0.23 s	–
Liu et al. [25]	0.01 s	0.01 s	10 s	100 s
da Costa et al. [12]	0.19 s	1.28 s	0.38 s	0.79 s
Sec-Health	0.36 s	0.89 s	0.17 s	0.4 s

TABLE 4. Client-side memory usage comparison between Sec-Health and the protocol proposed by da Costa et al. [12].

	Storage phase			Sharing phase		
	Rec. 1	Rec. 2	Rec. 3	Rec. 1	Rec. 2	Rec. 3
da Costa et al. [12]	7 MB	17 MB	147 MB	44 MB	53 MB	184 MB
Sec-Health	12 MB	26 MB	220 MB	12 MB	26 MB	220 MB

by organizing them into operations for storing and sharing (accessing) a health record. In addition, we particularly focus on comparing Sec-Health with proposals that employed CP-ABE by ranging the number of used attributes.

Table 3 shows comparison results of execution time. Rahulamathavan et al. [31] presented evaluations varying the number of CP-ABE attributes. However, the highest number of attributes was 10, and only health records with sizes smaller than 1 MB were used. Thus, we compare their results to our Record 1 (307.2 KB) results. In their worst case tested (10 attributes), their storage phase executed in 0.5 s. Sec-Health was more efficient, with a time of 0.36 s. Our sharing phase is also more efficient than theirs. While in the scenario with 10 attributes their sharing phase executed in 0.23 s, Sec-Health executed in 0.17 s. Thus, Sec-Health outperformed the related proposal.

Liu et al. [25] performed tests by ranging the number of CP-ABE attributes from 1 to 100, and using records with sizes smaller than 1 MB. Their storage phase was more efficient than ours in all scenarios, as their scheme executes in a constant time (0.01 s). This constant behavior is due to the costly security operations being run before the actual storage phase. In contrast, the execution time of Sec-Health storage phase varies according to the number of CP-ABE attributes. However, their sharing phase was always less efficient than ours. While their scheme executed this phase in approximately 100 s in the worst case (100 attributes), Sec-Health performed it in an average time of 0.4 s.

When compared to the protocol due to da Costa et al. [12], Sec-Health storage phase performed less efficiently than their storage phase when Record 1 was used with a small number of attributes (10 attributes), wherein their protocol executed in 0.19 s, while Sec-Health performed in 0.36 s. This is due to the fact that we employ public key encryption and Hyperledger Fabric blockchain, while they use symmetric encryption and Ethereum. Sec-Health, however, became more efficient when more CP-ABE attributes were employed (100 attributes), wherein Sec-Health executed in 0.89 s, while their work ran in 1.28 s. This shows Sec-Health storage phase does not have significant impact with the increase on the number of attributes as their proposal does.

Increasing the number of attributes introduces even lower impact on the execution times of Sec-Health sharing phase. This, combined to the fact that Ethereum takes more time than Hyperledger Fabric to search for blockchain transactions, makes our sharing phase more efficient than the same phase due to da Costa et al. [12]. Table 3 shows that our execution times for sharing a record were always lower than the times obtained by them for Record 1. While Sec-Health executed in 0.17 s and 0.4 s with 10 and 100 attributes, respectively, their proposal ran in 0.38 s and 0.79 s.

As shown in Table 4, we only compare memory usage with the work due to da Costa et al. [12], since other authors did not provide client-side memory overhead of their schemes. In the storage phase, Sec-Health introduced more memory overhead will all tested records. However, Sec-Health sharing phase used less memory than the related proposal for Records 1 and 2. This occurred because Ethereum reserves larger amounts of memory to search for blockchain transactions than Hyperledger Fabric. This characteristic, however, did not make Sec-Health more efficient with Record 3. Thus, for larger health records, Sec-Health produces more memory overhead than the related proposal.

The comparison results show that Sec-Health sharing phase, in general, executes in less time and uses less memory than the same phase from related work. More specifically, Sec-Health can reduce at least 26% the time to execute the sharing phase. There are cases in which the reduction achieves percentages of approximately 90%. Besides, Sec-Health reduces up to 50% the memory used in the sharing phase. There was only one tested scenario in which related work employed less memory in this phase. Regarding the storage phase, Sec-Health showed to be less efficient than related work in terms of memory usage. In addition, there

are scenarios in which our storage phase takes more time to execute compared to related work. However, in some scenarios, Sec-Health storage phase can be more efficient, reducing up to 30% the execution time.

VII. CONCLUSION

In this work, we proposed Sec-Health, a blockchain-based protocol that secures health records while addressing all of their main properties, namely confidentiality, access control, integrity, access revocation, emergency access, interoperability, and anonymity. Sec-Health shows security advantages compared to related proposals that present highly centralized mechanisms. While those proposals are generally based on a trusted or semi trusted server, Sec-Health affords several decentralized features, preventing one single entity from compromising the healthcare system. Furthermore, compared to decentralized solutions, our protocol addresses the challenging problem of fulfilling all the main properties of health records, whereas other solutions focus on offering mechanisms for specific properties only.

Experimental evaluations of a Sec-Health PoC demonstrated the practical feasibility of our protocol. Sec-Health sharing phase is, in general, more efficient than related work, reducing at least 26% of the time to execute the phase. There even cases in which our sharing phase can reduce nearly 90% of the execution time. The results also showed that Sec-Health sharing phase can reduce up to 50% the memory usage of related work. These results demonstrated that, with Sec-Health, health records can be accessed quicker and with less overhead to clients. In terms of storage phase, Sec-Health introduces more memory overhead and, in some scenarios, executes in more time than the same phase of related work. However, in some cases, our storage phase can also reduce up to 30% the execution time of related work.

As future work, we plan to implement the other Sec-Health mechanisms (e.g., those related to emergency access, anonymity) and evaluate them in different scenarios. Furthermore, we intend to test Sec-Health with other blockchain platforms to learn the trade-offs of using each of them and select the best one to implement our protocol. We will also investigate what types of modification we can apply to Sec-Health such that it can execute more efficiently along with the selected blockchain platform. Another future work is to design a collaborative scheme to detect and remove malicious nodes from the blockchain and IPFS networks. Finally, we plan to investigate a new version of Sec-Health in the context of quantum networks/internet.

REFERENCES

[1] C. S. Kruse, A. Stein, H. Thomas, and H. Kaur, "The use of electronic health records to support population health: A systematic review of the literature," *J. Med. Syst.*, vol. 42, no. 11, p. 214, Nov. 2018.

[2] Y. Al-Issa, M. A. Ottom, and A. Tamrawi, "EHealth cloud security challenges: A survey," *J. Healthcare Eng.*, vol. 2019, pp. 1–15, Sep. 2019.

[3] HIPAA Journal. *December 2021 Healthcare Data Breach Report*. Accessed: Sep. 2, 2022. [Online]. Available: https://www.hipaajournal.com/december-2021-healthcare-data-breach-report/

[4] I. M. Lopes, T. Guarda, and P. Oliveira, "General data protection regulation in health clinics," *J. Med. Syst.*, vol. 44, no. 2, p. 53, Feb. 2020.

[5] S. Mhatre and A. V. Nimkar, "Secure cloud-based federation for EHR using multi-authority ABE," *Progress in Advanced Computing and Intelligent Engineering* (Advances in Intelligent Systems and Computing), vol. 714. Singapore: Springer, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-13-0224-4_1

[6] R. Ganiga, R. Pai, M. Pai, and R. Sinha, "Security framework for cloud based electronic health record (EHR) system," *Int. J. Electr. Comput. Eng.*, vol. 10, pp. 455–466, Feb. 2020.

[7] 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.

[8] B. Shen, J. Guo, and Y. Yang, "MedChain: Efficient healthcare data sharing via blockchain," *Appl. Sci.*, vol. 9, no. 6, p. 1207, Mar. 2019.

[9] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Sep. 7, 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] J. Benet, "IPFS—Content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, Oakland, CA, USA, Dec. 2007, pp. 321–334.

[12] L. da Costa, B. Pinheiro, R. Araujo, and A. Abelem, "A decentralized protocol for securely storing and sharing health records," in *Proc. IEEE Int. Conf. E-Health Netw., Appl. Services (HealthCom)*, Bogotá, Colombia, Oct. 2019, pp. 1–6.

[13] W. B. Lee and C. D. Lee, "A cryptographic key management solution for HIPAA privacy/security regulations," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 34–41, Jan. 2008.

[14] LGPD. (2018). *Lei no 13.709, de 14 de Agosto de 2018 (in Portuguese)*. Accessed: Sep. 7, 2022. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709.htm

[15] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proc. 8th ACM Symp. Inf., Comput. Commun. Secur. (ASIA CCS)*, K. Chen, Q. Xie, W. Qiu, N. Li, W.-G. Tzeng, Eds. Hangzhou, China, May 2013, pp. 523–528.

[16] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Inf. Sci.*, vol. 479, pp. 567–592, Apr. 2019.

[17] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems (reprint)," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, 1983.

[18] FIPS. (2002). *Secure Hash Standard*. [Online]. Available: https://csrc.nist.gov/csrc/media/publications/fips/180/2/archive/2002-08-01/documents/fips180-2.pdf

[19] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 1976, T. Okamoto, Ed. Kyoto, Japan: Springer, Dec. 2000, pp. 162–177.

[20] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

[21] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, Feb. 2019.

[22] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the Internet of Money," 2015, *arXiv:1511.05740*.

[23] Hyperledger. *Hyperledger Fabric*. Accessed: Sep. 7, 2022. [Online]. Available: https://www.hyperledger.org/use/fabric

[24] M. Masud, G. S. Gaba, K. Choudhary, R. Alroobaea, and M. S. Hossain, "A robust and lightweight secure access scheme for cloud based E-healthcare services," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 3043–3057, Sep. 2021.

[25] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on E-healthcare records in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 1020–1026, Jan. 2018.

[26] M. Kumar and S. Chand, "A secure and efficient cloud-centric Internet-of-Medical-things-enabled smart healthcare system with public verifiability," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10650–10659, Oct. 2020.

[27] H. Qiu, M. Qiu, M. Liu, and G. Memmi, "Secure health data sharing for medical cyber-physical systems for the healthcare 4.0," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2499–2505, Sep. 2020.

[28] V. Vijayakumar, M. K. Priyan, G. Ushadevi, R. Varatharajan, G. Manogaran, and P. V. Tarare, "E-health cloud security using timing enabled proxy re-encryption," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 1034–1045, Nov. 2022.
[29] I. Abunadi and R. Kumar, "BSF-EHR: Blockchain security framework for electronic health records of patients," *Sensors*, vol. 21, no. 8, p. 2865, Apr. 2021.
[30] Y. Zhuang, L. R. Sheets, Y.-W. Chen, Z.-Y. Shae, J. J. P. Tsai, and C.-R. Shyu, "A patient-centric health information exchange framework using blockchain technology," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 8, pp. 2169–2176, Aug. 2020.
[31] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Bhubaneswar, India, Dec. 2017, pp. 1–6.
[32] M. Zghaibeh, U. Farooq, N. U. Hasan, and I. Baig, "SHealth: A blockchain-based health system with smart contracts capabilities," *IEEE Access*, vol. 8, pp. 70030–70043, 2020.
[33] M. T. de Oliveira, A. Bakas, E. Frimpong, A. E. D. Groot, H. A. Marquering, A. Michalas, and S. D. Olabarriaga, "A break-glass protocol based on ciphertext-policy attribute-based encryption to access medical records in the cloud," *Ann. Telecommun.*, vol. 75, nos. 3–4, pp. 103–119, Apr. 2020.

LEONARDO DA COSTA received the first M.Sc. degree in computer science from the Stevens Institute of Technology, USA, and the second M.Sc. degree in computer science from Universidade Federal do Pará (UFPA), Brazil, where he is currently pursuing the Ph.D. degree. Since 2020, he has been working as a Cybersecurity Researcher with the Samsung Research and Development Institute Brazil (SRBR). His main research interests include cryptographic protocols, blockchain, health records security, network security, Android security, and machine learning applied to security.

BILLY PINHEIRO received the Ph.D. degree in electrical engineering (PPGEE) from the Federal University of Pará (UFPA) with an emphasis on applied computing. He worked as a Postdoctoral Researcher with the Novel Enablers for Cloud Slicing Project (NECOS). He is currently a Researcher and a CTO with Amazônia Blockchain Solutions (Amachains). He has experience in computer science, with an emphasis on computer networks and distributed systems, working mainly on the following topics: virtualization, SDN, wireless mesh networks, cloud computing, blockchain, and architectures for microservices.

WEVERTON CORDEIRO is currently an Associate Professor with UFRGS. His research is broadly focused in the field of networking. His current research interests include community networks, SDN, PDPs, and NFV.

ROBERTO ARAÚJO received the master's degree in computer science from the Universidade Federal de Santa Catarina and the Ph.D. degree in informatics from the Theoretical Computer Science Group, TU-Darmstadt (Germany). He is currently an Associate Professor with Universidade Federal do Pará (Brazil). His current research interests include secure voting, secure protocols, and cryptography.

ANTÔNIO ABELÉM received the Ph.D. degree in computer science from the Catholic University of Rio de Janeiro (PUC-Rio, 2003). He is currently a Full Professor with the Computer Science Faculty, Federal University of Pará (UFPA), Belém, capital of the Brazilian state of Pará. In 2020, he was a Visiting Professor with the University of Massachusetts (UMass), in Amherst-MA, where he carried out research on quantum networks. He coordinates the Research Laboratory in Computer Network and Multimedia Communication (GERCOM), from which participates and coordinates in several national and international projects. His research is focused on future internet, software-defined networking, network function virtualization, the Internet of Things, performance analysis, and network security. He is an Associate Member of the Board of the Brazilian Computer Society (SBC). He is also a TPC member and a reviewer of national and international conferences and journals.