IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 17, 2020, accepted November 1, 2020, date of publication November 9, 2020, date of current version November 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036811

Preserving Privacy in Mobile Health Systems Using Non-Interactive Zero-Knowledge Proof and Blockchain

ANTONIO EMERSON BARROS TOMAZ[1], JOSÉ CLÁUDIO DO NASCIMENTO[2], ABDELHAKIM SENHAJI HAFID[3], (Member, IEEE), AND JOSÉ NEUMAN DE SOUZA[1], (Senior Member, IEEE)

[1]Computer Science Department, Federal University of Ceara, Fortaleza 60.440-900, Brazil
[2]Electric Engineering Department - Campus Sobral, Federal University of Ceara, Sobral 62.010-560, Brazil
[3]Network Research Laboratory, Université de Montréal, Montreal, QC H3C 3J7, Canada

Corresponding author: Antonio Emerson Barros Tomaz (emerson@crateus.ufc.br)

This study was supported by the Federal University of Ceará, the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Brazil, the Natural Sciences and Engineering Research Council of Canada, and CNPq research funding (# 426701 / 2018-6).

ABSTRACT The advent of miniaturized mobile devices with wireless communication capability and integrated with biosensors has revolutionized healthcare systems. The devices can be used by individuals as wearable accessories to collect health data regularly. This type of medical assistance supported by mobile devices to monitor patients and offer health services remotely is known as mobile health (mHealth). Although mHealth provides many benefits and has become popular, it can pose severe privacy risks. Many features in mHealth are managed through a smartphone. Thus, one of the most worrying issues involves communication between the monitoring devices and the smartphone. When communication uses Bluetooth, it is standard for a device to be paired with the smartphone; but generally, it is not exclusively associated with a specific mHealth app. This characteristic can allow a data theft attack by a malicious app or fake data injection by an illegitimate device. To address this issue, we present an authentication scheme based on Non-Interactive Zero-Knowledge Proof that is lightweight enough to run on mHealth devices with minimal resources. Our scheme ensures that legitimate devices interact exclusively with the official mHealth application. To ensure the patient's privacy-preserving throughout the system, we address the issues of storing, managing, and sharing data using blockchain. Since there is no privacy in the standard blockchain, we present a scheme in which the health data transmitted, stored, or shared are protected by Attribute-Based Encryption. The outcome is a system with fine-grained access control, entirely managed by the patient, and an end-to-end privacy guarantee.

INDEX TERMS Authentication, blockchain, resource-limited devices, Internet of Things, mobile health, privacy-preserving.

I. INTRODUCTION

The Internet of Things (IoT) age has promoted technological progress in various social areas. Thanks to the advent of IoT, medical devices that were previously available only in hospitals can now be used by patients as technological accessories. Wearable or implantable, such devices have been massively adopted to monitor individuals and offer health services remotely. This field of telemedicine is called mobile health (mHealth). Mobile health has improved the quality of care for patients outside the traditional clinical environment. This technology allows individuals to continuously monitor their health, collect physiological data, and share it with healthcare professionals.

The typical architecture of a mHealth system includes miniaturized mobile devices that collect health data using body sensors. These devices communicate via Wireless Body Area Network (WBAN). Ideally, patients should easily manage their data to choose with whom the data is shared. Once chosen, authorized healthcare professionals can recommend treatments based on the data collected without the patient's need to go to the health center. The use of mHealth

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

technology facilitates health monitoring, diagnosis, and treatment of disease – by providing patients with greater convenience. However, mHealth brings several challenges to security and privacy. Mobile health systems require robust security schemes since personal health data is among the most sensitive. To help address these challenges, we propose to investigate two problems: (1) authentication of the monitoring devices; and (2) storage and access control of the data.

Authentication of the monitoring devices. The advent of mobile devices capable of collecting health data in real-time has boosted the healthcare industry. These devices are small and light enough to be worn on the body as an accessory. However, to make this possible, they are built with limited resources, such as a small amount of RAM, a slow processor, and limited storage. Consequently, the data collection devices typically use the patient's smartphone to pre-process the data before storing and sharing it with healthcare professionals. In this scenario, smartphones are considered essential devices. As they are designed as personal devices, it is usually assumed that the user is actually the owner of such a device. In this way, many resources in mHealth are managed through smartphones. At the heart of the problem is the communication between the monitoring devices and the smartphone. Although there is great ease of interaction between devices, it is in this communication that there is a critical vulnerability of the system. The reason is that Bluetooth technology only pairs the monitoring device with the smartphone, but does not associate the monitoring device with a specific application. Therefore, any application that gains permission to use the communication channel can access any device connected to that channel. The authors in [1] were the first to call attention to this vulnerability and, motivated by that, they implemented an attack called *external Device Mis-Bonding* (DMB). The attack allows a malicious application to steal the patient's health data. In a variation of the attack, a fake device can inject incorrect data into the system. Thus, we consider it essential to develop a mechanism that associates the monitoring device with the official mHealth application, not only with the smartphone.

Storage and access control of the data. In general, traditional electronic health systems (e-Health) store and share data using local or cloud databases under the healthcare provider's control. This type of centralized architecture requires the patient to trust a third party to manage his/her data. A drawback that stands out in this scenario is that the patient completely loses control of his/her data – causing severe privacy issues. If the healthcare provider is unreliable, it may illegally share patient data. Even in scenarios where the healthcare provider is reliable, it may be technically unable to maintain data security. Consequently, highly sensitive private data is at risk of being unduly exposed in the face of malicious attacks on the database. The blockchain becomes an integral part of this system to eliminate the need for a central entity that typically manages and shares the data. Essentially, a blockchain is a distributed ledger capable of maintaining an immutable log of transactions carried out on a network [2]. Blockchain was initially proposed as a technology underlying the Bitcoin cryptocurrency [3]. A public blockchain, formed by unknown nodes and without a central authority, requires a mechanism capable of maintaining consensus on the network. A consensus mechanism is a process of agreement between the nodes, mutually suspect, on the same transaction transmitted to the network or on the state of the blockchain. On the blockchain network, the interested stakeholders deploy a smart contract. A smart contract is a tiny executable program that is stored inside a blockchain. Once certain conditions are met, the program can run automatically [4].

The problems mentioned here make the mHealth ecosystem a challenging environment, especially for application developers. Indeed, mHealth systems have the potential to improve the quality of many traditional healthcare services, as long as security and privacy issues are adequately addressed.

A. OUR CONTRIBUTION

The paper makes the following two contributions.

- The main contribution is to prevent attacks from external devices that corrupt communication between mHealth devices and their official application. Our solution is built using an authentication mechanism, based on Non-Interactive Zero-Knowledge Proof, lightweight enough to run on devices with limited computational resources.
- Additionally, to address the issues of storing, managing, and sharing data, we propose a blockchain-based approach. However, there is no data privacy in the standard blockchain environment in transactions and storage. So, we present a scheme in which data transmission, storage, and sharing use Attribute-Based Encryption. The patient specifies an access policy, and he/she distributes the decryption keys to legitimate users of the system. In this way, we eliminate the risk that healthcare providers collect or share data without authorization. Thus, we have a system that guarantees privacy and is entirely managed by the patient.

Our proposal offers an end-to-end patient privacy solution, that is, from collection to data storage.

B. ROADMAP

The rest of this paper is organized as follows. In Section II, we describe related works focusing on privacy-preserving in personal health systems, especially in the mHealth system. Section III provides an overview of our approach. Section IV presents the initial system configuration. Section V presents the authentication mechanism for monitoring devices. Section VI describes how the patient has exclusive control of his/her data. Section VII describes the process flow of data access by healthcare professionals. Section VIII evaluates the impacts of the proposed approach. Finally, we present our conclusion in Section IX.

II. MOTIVATION AND RELATED WORK

To show the importance of authenticating monitoring devices on a mHealth system, let us the following scenario.

A. MOTIVATION

Generally, a health monitoring device reports the results to a mHealth application running on the patient's smartphone using a wireless communication channel, which is Bluetooth. This scenario requires devices with significantly lower power consumption; thus, the communication should preferably be carried out using Bluetooth Low Energy (BLE) technology. However, when a user pairs his/her smartphone to the mHealth device, via BLE, the data communicated between the two devices is accessible to all the user's smartphone applications. Ideally, only the mHealth application should be allowed to communicate with the monitoring device. However, the smartphone operating system (OS), specifically Android, is not able to establish the corresponding access control. Thus, the device's operating system allows any application with permission to the Bluetooth channel to communicate with the monitoring device [1]. Note that OS ensures that the smartphone is paired with a monitoring device, but does not guarantee that the monitoring device is associated only with the specific mHealth application.

The authors, in [1], studied this vulnerability and analyzed dozens of healthcare-related applications. They found that none of them is protected by any mechanism that authenticates the monitoring device to the mHealth application. To show the vulnerability of most devices in the market, they implemented an attack called *external Device Mis-Bonding* (DMB). This attack exploits a vulnerability in the Android security model used for communicating with an external device (such as Bluetooth devices). The attacker can steal data from an Android device or help an adversary to deploy a spoofed device that injects fake data into the mHealth system. The DMB attack's success is possible due to the lack of an exclusive association between the monitoring device and its corresponding mHealth application. In the absence of protection at the OS level, manufacturers of monitoring devices have no choice but to implement security at the application layer to protect data privacy.

In this case, there are at least two possible attack scenarios, as reported in [1]:

(i) *Data-Stealing Attack.* A malicious application on the smartphone can steal the patient's data, provided it has Bluetooth permissions. The malicious capture of data can happen when the official mHealth application (the one the user expects to connect to the device) is not connected to the monitoring device. In this scenario, the malicious application can determine the right time to download, using only its Bluetooth permission and side-channel information.

(ii) *Data-Injection Attack.* An attacker uses a malicious monitoring device to pair with the patient's smartphone. This attack can happen as follows: (1) the malicious application uses its Bluetooth permission to collect information about the legitimate monitoring device; (2) the attacker clones the device using the information collected – such as MAC address, UUID, and device name – and places the clone in the vicinity of the legitimate device; and (3) the smartphone pairs with the clone, instead of the legitimate device. Once this is done, the cloned device can inject fake data into the official mHealth application.

The authors in [1] argue that since the current Android design does not provide a means to link an application to an external device, manufacturers need to develop their own authentication mechanism – but this can be very challenging. The main reason is that the devices are generally elementary; they do not have sufficient resources to perform authentication operations, such as executing cryptographic functions.

B. AUTHENTICATION BETWEEN WEARABLE DEVICES AND A MOBILE TERMINAL

The communication method between the monitoring devices and the central node (smartphone) must provide strong security mechanisms to ensure that confidential patient data cannot be accessed by an attacker [5]. Liu *et al.* [6] report that there are severe attacks on wearable devices since the communication channels are exposed. To counter these attacks, they proposed a lightweight authentication protocol between wearable devices and the smartphone using a challenge-response scheme. However, the proposed protocol is designed to authenticate two wearable devices simultaneously; furthermore, after local authentication, a cloud server needs to verify the two wearable devices' legitimacy to complete the authentication process. The authors [6] provided a formal security analysis of the protocol; however, they did not conduct experiments that show the time spent or memory consumption of the scheme.

Das *et al.* [7] proposed a lightweight authentication protocol between wearable devices and a mobile terminal. Once the mutual authentication is successful, data received by the mobile terminal can be uploaded to a cloud server. However, the authors did not address the issue of authentication among healthcare professionals, who wish to access data, and the cloud storage server. Similarly, Liu *et al.* [8] proposed an authentication scheme between wearable devices and a mobile terminal. However, unlike our proposal, the authors do not consider devices with limited resources. Their focus was on wearable devices, with considerable computational resources, able to generate and read QRCode as a part of the authentication process.

Le *et al.* [9] proposed a mutual authentication and access control based on Elliptic Curve Cryptography. The objective is to authenticate biosensors and mobile terminals in a healthcare environment. Their proposal [9] requires less computational overhead due to the use of ECC. However, it requires one or more trusted third parties (i.e., Key Distribution Center) to generate and control the key of the devices and users.

Huang *et al.* [10] proposed an integrated Personal Health Records (PHR) framework for privacy-preserving. The integrated PHR system collects patient data from multiple healthcare providers and stores it on a PHR Cloud Server to give the patient more control power. The proposed access control is based on ABE. However, their proposal requires all stakeholders to be registered with a trusted authority which generates and distributes users' keys. According to the authors, once a patient visits a healthcare provider, his/her related medical record is created and kept by that provider. Note that the healthcare providers collect data and create health records, and only then, PHR system collects the data from the records created by the providers. Therefore, nothing prevents healthcare providers from keeping a copy of patient records. Thus, the authors' efforts to provide patient-centered access control, through a centralized PHR, do not guarantee the privacy of the patients.

C. PRIVACY-PRESERVING USING BLOCKCHAIN

There are many efforts to preserve patients' privacy; several contributions have proposed integrating blockchain technology with personal health records systems [11]–[15]. In this section, we also describe some of these contributions that propose blockchain-based access control for mHealth systems.

Genestier *et al.* [16] presented a model in which patients manage access consent to their data in a decentralized way using blockchain. Although a smart contract performs access control, at least two entities are operating as centralized intermediaries between the patient's application and the blockchain: (1) a data management server; and (2) a consent management server. Upon data access request, the data management server consults the consent management server, which checks recorded authorizations in the blockchain. These servers are single points of failure that can impair system availability. Also, the authors do not appear to employ any data protection mechanism while the data is stored on the server. An attack on that server can expose the patient's sensitive data.

Liang *et al.* [17] proposed a mobile healthcare system integrated with blockchain for sharing health data. Each data access request is sent to the blockchain, where it is processed to obtain permission from the data owner. However, the system's implementation relies on a trusted third party, which receives storage or data query requests. The system requires the user to register with a cloud storage service provider to synchronize data. Although the authors have developed a Merkle tree-based method to ensure data integrity, the data is stored without any cryptographic scheme that guarantees the data's confidentiality in an eventual attack on the server.

Silva *et al.* [18] presented a cryptographic scheme to guarantee confidentiality, integrity, and authenticity of data in mHealth applications. The authors proposed a hybrid approach using symmetric and asymmetric encryption algorithms. However, RSA – the asymmetric algorithm employed by them – requires a very large key to provide an adequate level of security. For example, to achieve the same level of security as an elliptic curve cryptosystem with a 256-bit key (used in our paper), RSA needs a 3072-bit key [19], [20]. Due to the size of the key and the time required for processing, algorithms based on modular arithmetic, such as RSA, may make the scheme proposed in [18] unsuitable for mHealth systems with resource-limited devices.

The authors, in [21], proposed an access control model for personal health record systems. As in our approach, the authors store metadata corresponding to health records in the blockchain. Health records are stored encrypted on a cloud server. However, access control is performed using a Proxy Reencryption Scheme. This approach makes the data sharing process dependent on an intermediary, which is the proxy server responsible for re-encryption. Thus, the encryption keys and other information necessary for an authentication process are under the proxy server's control. This approach [21] suffers from the problem of single point of failure since it relies on centralized third party to control part of the system operations. Dagher *et al.* [22] proposed a framework that uses smart contracts in an Ethereum-based blockchain for access control. However, similar to the proposal in [21], the authors in [22] also use proxy re-encryption technique, making the system dependent on a third party.

Li *et al.* [23], proposed a fine-grained access control for mHealth systems, which uses multi-authority on Attribute-Based Encryption (ABE) scheme. They argue that the multi-authority model in ABE has advantages over the single-authority model. However, there is a dependency on a trusted third party to generate and distribute the decryption keys on both models. Similar to the model in [23], Rahulamathavan *et al.* [24] presented an approach to privacy-preserving in IoT ecosystems which employs ABE scheme with multi-authority integrated with blockchain. Our approach has a simpler architecture when compared to [23] and [24]. In our approach, there is no dependency on third parties; furthermore, the patient himself, assisted by his smartphone, can generate and distribute ABE scheme's keys to the system's users.

Lunardi *et al.* [25] proposed the architecture of a ledger-based access control scheme for IoT. Their focus is not on mHealth systems; however, it is related to our proposal since they investigate the use of Blockchain in the context of resources-limited IoT devices. The authors implement cryptographic algorithms on an Arduino device similar to the one we used in our experiments to evaluate our approach. They show that the Arduino was able to run the RSA and AES algorithms with an acceptable response time. However, it is not possible to evaluate the implemented algorithms' security level since the authors did not specify the size of the keys used. Thus, we cannot make a direct comparison (of the security level and of the execution time) between the algorithms implemented in [25] and the algorithms implemented in our paper. Nevertheless, our experiments show that our security scheme for resource-limited devices has an acceptable response time and a high security level.

D. LIMITED SOLUTIONS FOR PRIVACY-PRESERVING IN mHealth

The scientific community has massively investigated the security and privacy in healthcare systems; indeed, several surveys have been published in recent years [5], [26]–[30]. However, not all existing contributions did address privacy and security issues holistically. In the context of healthcare systems assisted by wearable devices, several contributions [16], [31]–[35] [36] did not address the authentication between these devices and the smartphone (or some type of a gateway). They assume that data that arrives at the smartphone is intact and is sent by legitimate devices. However, this assumption does not hold in most cases [1], [6]. In this paper, we propose a holistic solution for mHealth systems. We protect data from collection to storage/sharing. We are concerned with providing secure interactions between the smartphone, controlled by the user, and wearable devices with limited resources. Our proposal creates an exclusive association between the wearable device and the official mHealth application to solve the problems presented in [1].

Generally, patients are very concerned about the privacy of their data that is taken care by third-party cloud providers [37]. In contrast to existing contributions [16]–[18], [21]–[24], our proposal does not rely on trusted third parties or intermediate servers. Instead of storing patient health data on centralized servers, we integrate the mHealth architecture with blockchain/IPFS to maintain a distributed database where data can be managed exclusively by the patient. Our approach eliminates the risk of DDoS attacks, does not suffer from the problem of single point of failure, and guarantees availability. By comparing our approach, for example to the approach in [10], we eliminate healthcare providers' power to control patient data. Healthcare providers do not maintain the patient data, and therefore cannot share it with third parties; they are limited to analyzing the data.

III. MODEL OVERVIEW

The model proposed in this paper has a distributed architecture that integrates mHealth technology with blockchain technology to preserve patient privacy.

A. PARTICIPATING ENTITIES

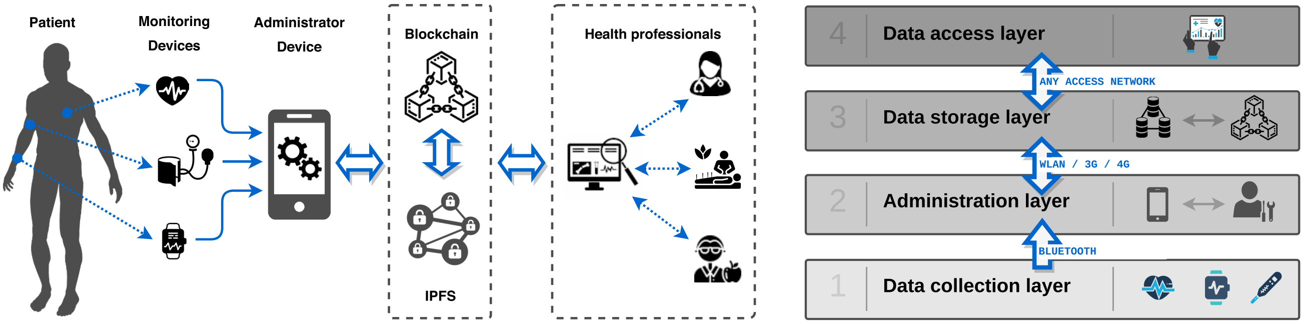
Our model considers six players as the entities of the system (see Fig. 1(a)).

- *Patient*. The data owner. He/she is responsible for administering the system and may grant or deny access to healthcare professionals.
- *Monitoring devices*. They are miniaturized devices equipped with biosensors, microcontrollers, and wireless data transmission. These devices can be incorporated into clothing or worn on the body as accessories. They can capture the patient's physiological signals, such as blood pressure, blood sugar rate, heart rate, sleep conditions, breathing patterns, among others. The collected data will be sent to a storage service (which, in our case, is the blockchain/IPFS), where it is available for analysis by authorized healthcare professionals.
- *Administrator device*. Due to limited processing and memory resources, monitoring devices must transmit the collected data to a more robust processing device. The patient may use their smartphone as a trusted device to configure and administer the system. The administrator device is equipped with an application capable of receiving, formatting, and encrypting the data before sending it for storage/sharing.
- *Blockchain*. Blockchain is an append-only, shared, fault-tolerant, and distributed database which maintains a set of records in the form of blocks. The blocks are transparent and are accessible by every blockchain node; however, they cannot be modified or deleted [38]. The blockchain network has three functions in the proposed system: (1) through a smart contract, it verifies the legitimacy of health professionals; (2) record the metadata of patients' health records and ensure that they are accessed only by authorized healthcare professionals; and (3) provide robustness against availability failures and data breach attacks.
- *IPFS*. InterPlanetary File System [39] is a peer-to-peer protocol for storing distributed data. In an IPFS-based network, stored files are referenced through a hash that is calculated exclusively based on their content. Files stored on an IPFS network are immutable – if the file is changed, IPFS considers the changed file as a new object, so a new hash is calculated. The IPFS network is used here because the financial cost of storing large files on the blockchain is very high. In this case, an IPFS network can be used to store health records, while the blockchain stores only the data hash and metadata.
- *Healthcare professionals*. They are the users of the data, adequately authorized by the patient, such as doctors, dentists, nutritionists, specialized clinics, among others. These healthcare providers can analyze the data and provide guidance or indicate treatments.

B. THE LOGICAL DATA ARCHITECTURE

Given the logical architecture of the data, the model is structured in four layers (see Fig. 1(b)). The architecture represents how data is handled from collection by monitoring devices to analysis by healthcare professionals. Regarding the communication between layers, our model follows the standard layered communication architecture of a WBAN, as presented in [40], [41].

1) *Data collection layer.* The data collected by the body sensors are used to monitor the patient's health. At a predefined time, the devices collect the patient's physiological data and transmit it to the next layer. Here, a WBAN using Bluetooth Low Energy (BLE) is



(a) Overview of the mHealth system integrated with the blockchain.

(b) Logical data architecture.

FIGURE 1. **Mobile health system architecture.**

employed to connect the monitoring devices to the administrator device.

2) *Administration layer*. This layer receives data coming from Layer 1 for processing before sending it to the storage service. The device used for this is usually the patient's smartphone, which acts as a gateway between the monitoring devices and the blockchain. Here, the patient has an application that provides system settings and access control functions. The communication between this layer and Layer 3 takes place through traditional home wireless local area networks (WLAN) or 3G/4G mobile networks.
3) *Data storage layer*. This layer refers to the data storage infrastructure using the blockchain and an IPFS network. Our approach proposes a wholly distributed architecture where no centralized third parties manipulate or store patient data. However, storing all health data on the blockchain is not adequate since it is expensive [21]. Thus, we decided to use a distributed storage system, such as IPFS, to store most health data. The data stored in IPFS is referenced by its hash, which is immutably stored in the blockchain's smart contract. Users interested in the data can consult the smart contract, discover the hash, and request the corresponding health record from IPFS, which provides secure and immutable data storage. More details about the interactions between blockchain and IPFS can be found in [42].
4) *Data access layer*. Once collected and properly stored, the data is ready to be analyzed by authorized healthcare professionals. Users view data through applications available on their personal or institutional devices. Access to data by healthcare professionals is defined by an access policy based on Attribute-Based Encryption (see Section IV-A).

C. DISTRIBUTED APPLICATION

Our approach proposes a distributed application composed of three parts: an administrator application, a smart contract, and a data access application. The administrator application, installed on the administrator device, is for the patient's exclusive use. The smart contract is deployed on the blockchain and is responsible for controlling access to data. The data access application runs on the devices of authorized healthcare professionals.

IV. INITIAL SYSTEM CONFIGURATION

To ensure patient privacy, our approach proposes the use of Attribute-Based Encryption at various points in the system, starting with the initial configuration.

A. ATTRIBUTE-BASED ENCRYPTION (ABE)

The concept of Attribute-Based Cryptography (ABE), initially proposed in [43], is a cryptographic primitive that supports confidentiality and fine-grained access control over encrypted data. The decryption of data is authorized based on an access policy, defined by the data owner, considering a set of descriptive attributes.

ABE supports two types: key-policy ABE and ciphertext-policy ABE. In the key-policy ABE scheme (KP-ABE) [44], the ciphertext is labeled with a set of descriptive attributes, while a user's private key is associated with an access policy, which specifies the types of ciphertext that the key can decrypt. In the ciphertext-policy ABE scheme (CP-ABE) [45], a set of attributes is assigned to the user's private key, while the access policy is associated with the ciphertext. Only users whose attributes satisfy the access policy can access the data. An access policy $\mathbb{A}$ is a rule that returns 0 or 1 given a set of attributes $\mathbb{L}$. We say that $\mathbb{L}$ satisfies $\mathbb{A}$ if and only if $\mathbb{A}$ answers 1 in $\mathbb{L}$.

In traditional CP-ABE schemes, there are three types of entities: key generation center (KGC), the encryptor (data owner), the decryptor (data users) [46], [47]. KGC generates the private key to configure the system and generates/distributes the secret key for each data user according to their attributes. The user can then use his/her secret key to decrypt the ciphertext; however, he/she will only be successful if his/her attributes satisfy the corresponding access policy.

A drawback of the traditional scheme is that it depends on a trusted third party: the key generation center. Usually, for the proper functioning of the system, we assume that KGC is an honest entity. However, in reality, there are many problems involved in this type of architecture, as indicated in [42]. Note that as KGC generates users' secret keys, it has the ability to decrypt all stored data. If KGC does not behave as an honest entity, it can abuse keys and leak private data. As such, the data owner loses the ability to control their own data. In practice, it is not easy to find KGC that is reliable.

Our proposal adopts the CP-ABE scheme involving four entities: the data owner, the data users, the key generation center, and a storage provider.

- *Data owner.* Here, the owner is the patient. He/she is responsible for encrypting the information to be shared according to an access policy.
- *Data users.* They are the healthcare professionals who can decrypt the information only if their attributes satisfy the corresponding access policy.
- *Key generation center.* In our approach, this center is represented by the administrator device of the data owner. In this case, the patient himself/herself can generate and distribute the decryption keys to the data users. Thus, the patient has greater control over his data.
- *Storage provider.* It is the entity responsible for storing encrypted data. In our approach, storage is accomplished by combining the blockchain network with the IPFS network. IPFS stores the health data set in an encrypted form while the blockchain is responsible for access control. The blockchain also stores metadata that will be used by data users.

A CP-ABE scheme consists of four algorithms, mathematically formalized in [45], and described below.

- $\texttt{Setup}\ (\rho)$. The algorithm that takes a predefined security parameter ρ and outputs a public key $\mathcal{P}_{\mathcal{K}}$ and a private master key $\mathcal{M}_{\mathcal{K}}$.
- $\texttt{KeyGen}\ (\mathcal{M}_{\mathcal{K}}, \mathbb{L})$. The Key generator algorithm takes the master key $\mathcal{M}_{\mathcal{K}}$ and a set of attributes $\mathbb{L}$ as input. The output is a secret key $\mathcal{S}_{\mathbb{L}}$ corresponding to $\mathbb{L}$. Note that the attribute set describes one or more data users. Each data user will have their own secret key $\mathcal{S}_{\mathbb{L}}$.
- $\texttt{Enc}_{\texttt{ABE}}\ (\mathcal{P}_{\mathcal{K}}, \gamma, \mathbb{A})$. The encryption algorithm takes the public key $\mathcal{P}_{\mathcal{K}}$, the information to be encrypted γ, and the access policy $\mathbb{A}$ as input. The output is the ciphertext $\gamma_{\mathbb{A}}$, which implicitly contains the policy $\mathbb{A}$.
- $\texttt{Dec}_{\texttt{ABE}}\ (\mathcal{P}_{\mathcal{K}}, \gamma_{\mathbb{A}}, \mathcal{S}_{\mathbb{L}})$. The decryption algorithm takes the public key $\mathcal{P}_{\mathcal{K}}$, the encrypted information $\gamma_{\mathbb{A}}$, and the secret key of the data user $\mathcal{S}_{\mathbb{L}}$ as input. If $\mathbb{L}$, the set of user attributes, satisfies $\mathbb{A}$, then the algorithm decrypts the ciphertext, and the output is γ.

ABE is a one-to-many public key cryptographic primitive. The data owner encrypts their data and defines various user groups who can decrypt it, as long as they satisfy the access policy. This property makes ABE very attractive for implementing fine-grained access controls.

B. INITIAL CONFIGURATION PROCESS

An initial configuration process is required to prepare the system for use. The configuration is performed only once by the data owner (patient) using the administrator device. Note that the administrator device assumes the role of the *key generation center*, required in traditional systems. The configuration consists of the following steps.

1) The administrator device must run the algorithm $\texttt{Setup}(\rho) \rightarrow (\mathcal{M}_{\mathcal{K}}, \mathcal{P}_{\mathcal{K}})$. The algorithm returns the system configuration parameters: the private master key $\mathcal{M}_{\mathcal{K}}$, which will be known only by the data owner, and the public key $\mathcal{P}_{\mathcal{K}}$ of the ABE scheme.
2) The administrator device deploys the smart contract, previously coded in the mHealth application, on the blockchain. After deployment, the mHealth application obtains the public address of the smart contract.

The smart contract stores the blockchain address of the mHealth application to ensure that only the administrator device can write data to the blockchain. The smart contract must verify the legitimacy of the administrator device before recording data sent by it (see Section VI).

V. DATA COLLECTION SUBSYSTEM

The feasibility of remote health monitoring relies, fundamentally, on the correct and safe data collection by healthcare devices. Upon receiving data, the administrator device processes and transmits them to storage services, where healthcare professionals access them. However, the system's functioning may be impaired if the communication between the monitoring devices and the other players is not secure. Thus, it is crucial to authenticate these devices. Therefore, in this section, we present an authentication scheme for monitoring devices.

A. MATHEMATICAL PRELIMINARIES

We propose an authentication scheme based on a Non-Interactive Zero-Knowledge Proofs (NIZKP) for the Elliptic Curve Discrete Logarithm Problem (ECDLP). This subsection presents a brief explanation of the corresponding cryptographic primitives.

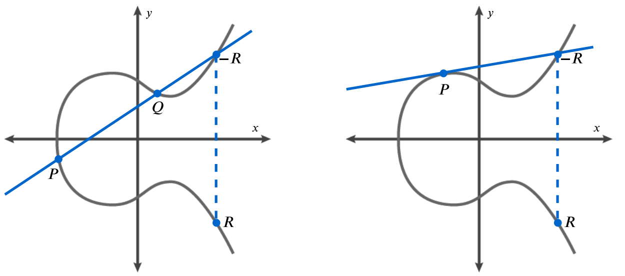
1) ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

An elliptic curve E defined over a field $\mathbb{F}$, denoted by $E(\mathbb{F})$, is a collection of points $(x, y) \in \mathbb{F} \times \mathbb{F}$ that satisfy the equation

$$y^2 = x^3 + ax + b$$

where a and $b \in \mathbb{F}$ and $4a^3 + 27b^2 \neq 0$. This condition indicates that the polynomial $x^3 + ax + b$ has no repeated roots. In addition to the points (x, y), the curve must include an identity element, called *point at infinity* and denoted by ∞.

For cryptographic applications, there is a special interest in elliptic curves over finite fields. On an elliptic curve E defined over a finite field $\mathbb{F}_p = \{0, 1, \cdots, p-1\}$, in which p is a sufficiently large prime number, all variables and coefficients assume values in the set of integers $\mathbb{F}_p$.



(a) Addition of two points. (b) Point doubling.

FIGURE 2. Point addition and point doubling operations on an elliptic curve.

All operations must be calculated modulo p, such as

$$y^2 \equiv x^3 + ax + b \ (mod\ p)$$

with $4a^3 + 27b^2 \not\equiv 0 \ (mod\ p)$. All the points $P_i = (x_i, y_i)$ that satisfy this condition are said to belong to the elliptic curve.

The addition operation between two points, P and Q, on a curve can be defined as follows: $P + Q$ is a point $R \in E$. Geometrically, R is defined by drawing a line that passes through the points P and Q and intersects the curve at the third point $-R$ (see Fig. 2(a)). The point $-R$ is the reflection over the x axis of the point R. Thus, we have

$$P + Q = R.$$

To duplicate a point $P \in E$, a tangent line is drawn in P, which intersects the curve at a second point $-R$ (see Fig. 2(b)). Thus,

$$R = P + P = 2P.$$

2) ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (ECDLP)

The use of Elliptic Curve Cryptography (ECC) was initially proposed in [48] and [49]. ECC security is based on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) over finite fields. Informally, we can say that the discrete logarithm operation is the inverse of point multiplication by a scalar.

The multiplication of a point $P \in E(\mathbb{F}_p)$ by a scalar $n \in \mathbb{N}$ is defined as adding P to itself n times. This operation results in a point $T \in E(\mathbb{F}_p)$, such as

$$T = P + P + \cdots + P = nP.$$

Thus, the discrete elliptic logarithm of T with respect to P is the integer n such that $T = nP$.

With that in mind, ECDLP can be defined more specifically as follows. Given an elliptic curve E over some finite field $\mathbb{F}_p$, and two points $P, T \in E(\mathbb{F}_p)$, it is computationally infeasible to find an integer n such that $T = nP$. If p, the order of the prime field $\mathbb{F}$, is sufficiently large, no efficient algorithm is known to solve this problem. However, knowing n and point P, it is computationally fast to calculate nP.

ECC has attracted special attention because it achieves the same level of security requiring less computational resources, when compared to traditional public key cryptography, such as RSA [50]. This makes it ideal for security implementations on low-resource devices, such as the devices used in mHealth. For example, an implementation using ECC requires only a 256-b it key, while RSA needs to use a 3072-bit key to achieve the same level of security [19], [20].

3) ELLIPTIC CURVE DIFFIE–HELLMAN (ECDH)

ECDH is based on the classic Diffie-Hellman secret key exchange protocol [51]. This protocol allows two users to share a secret key using an insecure communication channel. The original Diffie-Hellman (DH) protocol is based on the Discrete Logarithm Problem defined in terms of modular arithmetic, while ECDH is based on ECDLP.

Consider that Alice and Bob want to share a secret key between themselves; they agree to use ECDH protocol. Initially, Alice and Bob agree on three public parameters: a sufficiently large prime number p, an elliptic curve E over $\mathbb{F}_p$, and a point $G \in E$. Then, the parameters are defined:

1) Alice chooses a number $\kappa_a \in \mathbb{F}_p$ and calculates a point $Q_a = \kappa_a G$, where κ_a is the private key and Q_a is the public key of Alice. Likewise, Bob chooses an integer κ_b and calculates the point $Q_b = \kappa_b G$. Then, they send these points to each other.
2) Alice calculates the shared secret key, using her private key and Bob's public key, $S = \kappa_a Q_b$. Likewise, Bob calculates the secret key, using his private key and Alice's public key, $S = \kappa_b Q_a$. Note that S is the same point for Alice and Bob, such that

$$S = \kappa_a Q_b = \kappa_a(\kappa_b G) = \kappa_b(\kappa_a G) = \kappa_b Q_a.$$

Even though intruder Eve intercepts the points Q_a and Q_b that travel through the unsafe channel, it is computationally difficult to find the secret key S since she needs to know κ_a or κ_b. However, to compute κ_a or κ_b, she needs to solve the discrete logarithm problem.

4) NON-INTERACTIVE ZERO-KNOWLEDGE PROOF (NIZKP)

In the late 1980s, the concept of Zero-Knowledge Proof (ZKP) was introduced in [52]. A ZKP system is a protocol that enables one party, called **prover**, to prove that some statement is true to another party, called **verifier**, but without revealing anything but the truth of the statement. In general, these systems are formulated as a decision problem, where the statements are associated with a language of the $\mathcal{NP}$ class, so the prover must prove to the verifier that a statement x belongs to a certain language L of the class $\mathcal{NP}$.

A ZKP system is an interactive protocol that involves exchanging messages between the prover and the verifier for a specified number of rounds. At the end of these exchanges, if the statement is true, the verifier must be convinced of this truth. However, if the statement is false, the verifier will discover the lie with a high probability. Each round is made up of 3-moves, which are three messages called *commitment*, *challenge*, and *response*. Initially, the prover generates a first message (commitment), which is the statement to be proved and sends it to the verifier. Then, the verifier randomly chooses a challenge and sends it to the prover. Finally, the prover calculates the response based on the challenge and sends it to the verifier.

However, in some scenarios, interactions are not desirable. In these situations, the standard approach is to use a variant of the ZKP, called Non-Interactive Zero-Knowledge Proofs (NIZKP). In this case, the challenge-response process is non-interactive; indeed, a single message is sent from the prover to the verifier.

A NIZKP system can be formulated as follows. Let R be a binary relation and let $L = \{x \mid \exists w : (x, w) \in R\}$ be the language where x is a statement and w is a *witness* for the membership of $x \in L$, so that it is possible to efficiently check whether or not $(x, w) \in R$. A non-interactive proof system $(\mathcal{P}, \mathcal{V})$ between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for a language L with a binary relation R is a pair of algorithms, modeled as Turing Machines, such that $\mathcal{P}$ runs in probabilistic polynomial time and $\mathcal{V}$ runs in deterministic polynomial time satisfying the following properties:

1) *Completeness*. It is the capacity of $\mathcal{P}$ to convince $\mathcal{V}$ that the statement x is true, as long as $\mathcal{P}$ has a witness w of it. Therefore, for any $x \in L$ and polynomial $p(\cdot)$, x is accepted by $\mathcal{V}$ with high probability, such as
$$\Pr[\pi \leftarrow \mathcal{P}(x, w) : \mathcal{V}(x, \pi) = 1] \geq 1 - \frac{1}{p(|x|)}.$$
2) *Soundness*. It is the capacity of $\mathcal{V}$ to protect himself from being convinced of false statements, except with a very small probability. Therefore, for any $x \notin L$, a polynomial $p(\cdot)$ and a malicious prover $\mathcal{P}'$, the probability that $\mathcal{V}$ will accept x is negligible in terms of $|x|$, given by
$$\Pr\left[\pi \leftarrow \mathcal{P}'(x) : \mathcal{V}(x, \pi) = 1\right] < \frac{1}{p(|x|)}.$$
3) *Zero-knowledge*. For any statement $x \in L$, provided by $\mathcal{P}$, no information is revealed from x to $\mathcal{V}$ that it could not compute alone before running the protocol. Given a simulator $\mathcal{S}\,im$, that runs in probabilistic polynomial time, any information calculated from a real proof can also be calculated from a simulated proof. In this way, the results produced by $(\mathcal{P}, \mathcal{V})$ and $\mathcal{S}im$ are computationally indistinguishable,
$$\{\mathcal{V}(x, \mathcal{P}(x, w))\} \approx \{\mathcal{S}im(x)\}.$$

The transformation of a ZKP to a NIZKP is achieved by the Fiat-Shamir heuristic [53], where a secure cryptographic hash function generates the challenge. In this case, $\mathcal{P}$ calculates the committed statement normally. However, the challenge, instead of chosen by $\mathcal{V}$, is replaced by a hash.

B. AUTHENTICATION OF MONITORING DEVICES

In this section, to address the issues presented in the Section II, we present the design of the device authentication mechanism.

1) THE NIZKP-BASED AUTHENTICATION SCHEME

We propose a lightweight mechanism to verify the legitimacy of the monitoring devices. The mechanism is based on NIZKP due to its security guarantees.

To build any NIZKP system, the choice of the mathematical problem that forms its base is a fundamental element. In this paper, the basic problem chosen is the Elliptic Curve Discrete Logarithm Problem (ECDLP). Systems based on elliptic curves, when compared to RSA, require less computing power and less memory consumption while providing the same level of security. Our scheme replaces heavy asymmetric cryptography used in traditional Public Key Infrastructure (PKI) with one that is more suitable for resource-limited devices.

There are many ways to build zero-knowledge proof systems. In 1991, Schnorr presented a zero-knowledge proof protocol based on the traditional discrete logarithm problem, known as Schnorr protocol [54]. To implement the NIZKP system used in this paper, we adopted a variation of the Schnorr protocol developed based on ECDLP. However, we are especially interested in the non-interactive form of the protocol [55]. The fundamental difference is in generating the challenge; instead of being produced by the verifier, it is produced through the Fiat-Shamir transformation [53], which uses a secure cryptographic hash function.

Typically, a NIZKP system consists of two steps:

1) The first stage involves a configuration process and, therefore, still requires some interactions between the parties. More specifically, the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ must share some information. In our case, one of the agreed public information is the elliptic curve `secp256k1` [56], the same used by the Bitcoin system. In practice, we assume that this proposal is implemented using the `secp256k1` curve for all ECC-based protocols. Specifically, the applications present on the monitoring devices, the administrator device, and the blockchain are programmed to use the parameters defined in `secp256k1`. The second public information agreed between the parties is the public key of $\mathcal{P}$ that will be used in the authentication process. Finally, we include the third information agreed between the parties, exclusively for this proposal – the shared secret key for data encryption, which is known only to $\mathcal{P}$ and $\mathcal{V}$. This last two information is generated in the registration phase (see Section V-B2).
2) The second stage is where NIZKP actually occurs. This is the authentication phase of the device, which must occur quickly and entirely in a non-interactive way. Here the proof is generated by $\mathcal{P}$ and validated by $\mathcal{V}$. The proof is generated based on the first phase information. The prover $\mathcal{P}$ sends the proof in a single message. Upon receiving the message, $\mathcal{V}$ processes the information and decides whether to accept or reject the proof.

Our authentication scheme requires that monitoring devices go through a registration process first. Thus, our

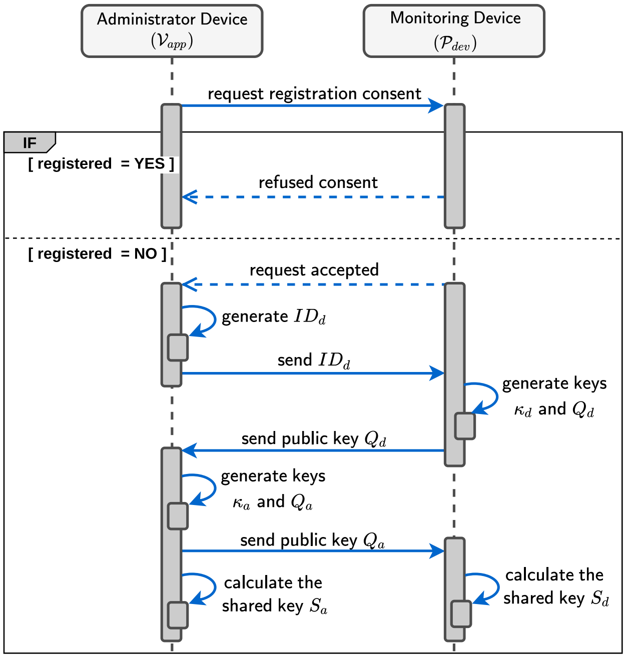

Administrator Device ($\mathcal{V}_{app}$)
Monitoring Device ($\mathcal{P}_{dev}$)
request registration consent
IF
[registered = YES]
refused consent
[registered = NO]
request accepted
generate ID_d
send ID_d
generate keys κ_d and Q_d
send public key Q_d
generate keys κ_a and Q_a
send public key Q_a
calculate the shared key S_a
calculate the shared key S_d

FIGURE 3. The workflow of the registration phase of the monitoring devices.

scheme consists of two phases: registration and authentication.

2) DEVICE REGISTRATION PHASE

Initially, the monitoring device expects the administrator device to initiate a connection. After that, the parties must execute Protocol 1 (see Fig. 3). Protocol 1 is based on the Elliptic Curve Diffie–Hellman (ECDH) [48], [49]. The shared secret key can be used in symmetric cryptographic systems to encrypt communication between the parties.

We assume that the registration process is carried out in a secure mode. For example, the patient must register a device in a private place where he/she is sure that only legitimate devices are present. With the device in hand, the patient can pair and register the device. This precaution minimizes the possibility of a malicious device getting registered.

Protocol 1: Device Registration

- ***Goal:*** *register a monitoring device.*
- ***Players:*** *the monitoring device $\mathcal{P}_{dev}$, operating as a provider, and the mHealth App $\mathcal{V}_{app}$ installed on the administrator device, operating as a verifier.*
- ***Common input:*** *curve $E(\mathbb{F}_p)$; generator $G \in E(\mathbb{F}_p)$.*

Steps:

1) *Assuming there is a Bluetooth connection between the devices, $\mathcal{V}_{app}$ sends to $\mathcal{P}_{dev}$ a registration consent message.*
2) *$\mathcal{P}_{dev}$ checks if there is already a previous association with another $\mathcal{V}_{app}$. If the answer is YES, the registration request is rejected because the device can not be linked to another App. Otherwise, the process continues at step 3.*
3) *$\mathcal{V}_{app}$ generates and sends an identifier ID_d to $\mathcal{P}_{dev}$.*
4) *$\mathcal{P}_{dev}$ generates its private key by choosing a secret integer $\kappa_d \in \mathbb{F}_p$ at random. It then generates its public key by calculating $Q_d = \kappa_d G$, in which $Q_d \in E(\mathbb{F}_p)$.*
5) *$\mathcal{P}_{dev}$ sends Q_d to $\mathcal{V}_{app}$ to allow $\mathcal{V}_{app}$ to calculate the shared secret key.*
6) *$\mathcal{V}_{app}$ generates its private key by choosing a secret integer $\kappa_a \in \mathbb{F}_p$ at random. It then generates its public key $Q_a = \kappa_a G$, in which $Q_a \in E(\mathbb{F}_p)$.*
7) *$\mathcal{V}_{app}$ sends its public key Q_a to $\mathcal{P}_{dev}$.*
8) *Now, both can calculate the shared secret key. $\mathcal{V}_{app}$ calculates $S_a = \kappa_a Q_d$ and $\mathcal{P}_{dev}$ calculates $S_d = \kappa_d Q_a$. So, the secret key is $S_a = S_d$.*

3) DEVICE AUTHENTICATION PHASE

In an ECDLP-based NIZKP, each monitoring device uses a public key, point Q_d (generated by Protocol 1), to prevent a malicious prover from proving false statements. Thus, any proof prepared by a legitimate prover must be constructed based on Q_d. Authentication occurs according to Protocol 2 (see Fig. 4).

Protocol 2: Generation of NIZKP

- ***Goal:*** *authenticate a monitoring device to carry out the data transmission.*
- ***Players:*** *the monitoring device $\mathcal{P}_{dev}$, operating as a prover; the mHealth App $\mathcal{V}_{app}$ installed on the administrator device, operating as a verifier.*
- ***Common input:*** *curve $E(\mathbb{F}_p)$; generator $G \in E(\mathbb{F}_p)$; public key Q_d and ID_d of $\mathcal{P}_{dev}$.*
- ***Private input:*** *secret key S_d of $\mathcal{P}_{dev}$.*

Steps:

1) *$\mathcal{P}_{dev}$ collects, through body sensors, the health data set, denoted by data_{sim}.*
2) *$\mathcal{P}_{dev}$ chooses an integer $\upsilon \in \mathbb{F}_p$ at random and then calculates the point $A = \upsilon G$.*
3) *$\mathcal{P}_{dev}$ calculates the challenge σ using a cryptographic hash function $\mathtt{H}$, such as $\sigma = \mathtt{H}(G||Q_d||A||ID_d)$.*
4) *$\mathcal{P}_{dev}$ calculates the response π to the challenge σ, such that $\pi = \upsilon + \sigma \cdot \kappa_d \pmod p$.*
5) *$\mathcal{P}_{dev}$ encrypts health data. To do this, it runs the algorithm $\mathtt{AES}(S_d, \mathsf{data}_{sim}) = \mathsf{data}_{enc}$, where $\mathtt{AES}$ is the symmetric encryption algorithm, S_d is the shared key generated by Protocol 1 and data_{enc} is the encrypted data.*
6) *$\mathcal{P}_{dev}$ generates a package pac_{nizkp} containing NIZKP and encrypted health data data_{enc}. The package pac_{nizkp} is logically partitioned into four segments, containing the following information:*
    a) *The first contains the point A, calculated in step 2;*
    b) *The second contains the response π, generated in step 4;*
    c) *The third contains the device identifier ID_d;*
    d) *The fourth contains the encrypted health data data_{enc}.*

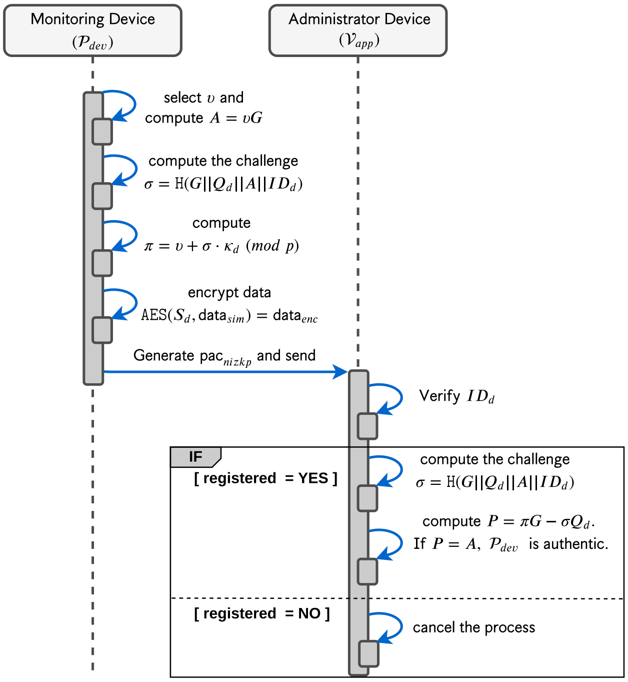

Monitoring Device
$(\mathcal{P}_{dev})$
Administrator Device
$(\mathcal{V}_{app})$
select υ and
compute $A = \upsilon G$
compute the challenge
$\sigma = \mathtt{H}(G||Q_d||A||ID_d)$
compute
$\pi = \upsilon + \sigma \cdot \kappa_d \ (mod\ p)$
encrypt data
$\mathtt{AES}(S_d, \text{data}_{sim}) = \text{data}_{enc}$
Generate pac_{nizkp} and send
Verify ID_d
IF
[registered = YES]
compute the challenge
$\sigma = \mathtt{H}(G||Q_d||A||ID_d)$
compute $P = \pi G - \sigma Q_d$.
If $P = A$, $\mathcal{P}_{dev}$ is authentic.
[registered = NO]
cancel the process

FIGURE 4. Authentication process: generation of NIZKP and sending of data by the monitoring device; verification of NIZKP by the mHealth App to ensure the legitimacy of the device.

7) *Finally, $\mathcal{P}_{dev}$ sends the package pac_{nizkp} to $\mathcal{V}_{app}$.*

To ensure that the authentication package came from a legitimate device, the verifier must execute Protocol 3 (see Fig. 4).

Protocol 3: Verification of NIZKP

- ***Goal:** verify the legitimacy of the monitoring device and receive the data.*
- ***Players:** the monitoring device $\mathcal{P}_{dev}$, operating as a prover; the mHealth App $\mathcal{V}_{app}$ installed on the administrator device, operating as a verifier.*
- ***Common input:** curve $E(\mathbb{F}_p)$; generator $G \in E(\mathbb{F}_p)$; public key Q_d and ID_d of $\mathcal{P}_{dev}$; package pac_{nizkp}.*

Steps:

1) *$\mathcal{V}_{app}$ receives the package pac_{nizkp} formed by: point A; response π; ID_d; encrypted data data_{enc}.*
2) *$\mathcal{V}_{app}$ checks if ID_d is from a registered device. If true, the process continues at step 3. Otherwise, the process is canceled.*
3) *$\mathcal{V}_{app}$ uses the public key Q_d associated with ID_d to calculate the challenge, as calculated by $\mathcal{P}_{dev}$, $\sigma = \mathtt{H}(G||Q_d||A||ID_d)$.*
4) *$\mathcal{V}_{app}$ calculates a point $P = \pi G - \sigma Q_d$ and checks if $P = A$.*
    - *If $P = A$, then $\mathcal{P}_{dev}$ is a legitimate device and Protocol 4 can be run to process the data.*
    - *If authentication fails, the process is terminated.*

When a security protocol is based on Schnorr's NIZKP, the threat of replay attacks must be considered. To avoid this specific attack, we can add more information to compose the hash function input that calculates the challenge. Such information must be a sequential number that identifies the package pac_{nizkp}. The verifier must observe whether the ID_d of the device and the package number form a unique identification.

C. EXPERIMENTAL EVALUATION

We conducted experiments to evaluate whether the protocols proposed in this section are suitable for running on devices with limited resources. The experiments' goal is to evaluate the algorithms' performance under two aspects: computational cost to generate NIZKP and the consumption of RAM and flash memories in the monitoring devices.

1) EXPERIMENTS SETUP

The environment of the experiments involves the following aspects:

- **Implementation.** The functions that involve ECC, within the protocols proposed in this section, were implemented based on the library $\mathtt{micro-ecc}$ [57]. This library allows implementing the ECDH and ECDSA algorithms on 8-bit processors using the C language. We use some functions from this library to implement a part of our protocols.
- **Elliptic curve.** We chose to implement the scheme using the elliptic curve `secp256k1` [56], whose parameters are recommended by Standards for Efficient Cryptography Group (SECG) [58].
- **Hardware.** We chose a very limited device to represent the monitoring device: the Arduino Nano. It is a small prototyping board based on the Atmel ATMega 328P microcontroller (8-bit) clocked at 16MHz and only 2KB of RAM and 32KB of flash memory. To send the data, we added to the Arduino a Bluetooth module of type BLE V4.0 HM-10. As an administrator device, to receive and process data, we used an Android 9 smartphone with a quad-core 1.8 GHz processor and 4GB of RAM.

2) RESULTS

In this subsection, we present the experiment performed from the monitoring device. Table 1 shows the time required for the devices to perform each of the operations related to the registration process (Protocol 1). Table 2 shows the time required to perform operations related to the generation and verification of NIZKP (Protocol 2 and Protocol 3, respectively). Table 3 shows the amount of memory needed to run the proposed scheme on the monitoring device.

Based on these results, we found that the Arduino Nano is able to run the Device Registration algorithm (Protocol 1) and the NIZKP Generation algorithm (Protocol 2) while consuming few resources. At runtime, our scheme occupies only 63.5% of RAM available on this device. As for flash memory, note that the compiled code from Protocols 1 and 2

TABLE 1. Average registration protocol runtime (Protocol 1).

Operations	Arduino	Smartphone
Key pair generation	3.784 s	0.195 s
Initial data exchange and processing (e.g., public key)	4.139 s	-
Generation of shared secret	3.785 s	0.045 s
Total execution time (including transmission): **13.144 s**		

TABLE 2. Average NIZKP protocols runtime (Protocols 2 and 3).

Operations	Arduino	Smartphone
Encryption of health data	0.055 s	-
Challenge generation	0.033 s	-
Generation of NIZKP (including the challenge)	4.300 s	-
Formation of the data package (Protocol 2, step 6)	3.747 s	-
Verification of NIZKP	-	0.213 s
Decryption of data	-	0.014 s
Total execution time (including transmission): **9.384 s**		

TABLE 3. Memory used by the authentication scheme on the monitoring device.

Data	Used Memory
Private key	32 Bytes
Public key	64 Bytes
Shared secret	32 Bytes
Complete data package	113 Bytes
Runtime algorithms	1.3 KB
Compiled algorithm (in flash memory)	24.5 KB

occupy only 24.5KB. The data encryption/decryption times refer to a 16-byte data block.

We believe that the execution times of the proposed are acceptable for a real mHealth environment. Especially, if we consider the device's low processing capacity and the level of security offered by the scheme. The level of security is compared to an RSA-based scheme with a 3072-bit key. It is difficult to make a broad/comprehensive comparison of our approach with related work (see Section II); most existing contributions do not address authentication between wearable devices and smartphones (or other types of gateways). To the best of our knowledge, our proposal is unique in the set of security characteristics it supports. However, we found that our proposal can be qualitatively compared with [59]; the comparison concerns the authentication process. In [59], the authors proposed an authentication scheme between medical devices and a smart e-health gateway. They developed a public key-based handshake protocol. To evaluate their proposal, they used a medical device equipped with a 16MHz MSP430 microcontroller, 128KB of ROM, and 16KB of RAM. The authentication process between the device and the gateway takes approximately 15 seconds. Note that the device used in the experiments [59] is similar to the one we used in our experiments; however, our authentication scheme achieves a shorter response time of approximately 9 seconds. Furthermore, in our scheme, the transmission overhead is only 113 bytes, whereas in the proposal in [59], it is 1190 bytes.

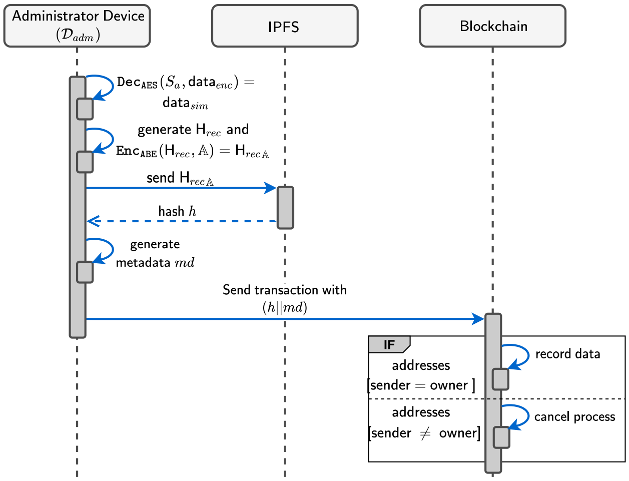

Administrator Device
($\mathcal{D}_{adm}$)
IPFS
Blockchain
$\texttt{Dec}_{\texttt{AES}}(S_a, \mathsf{data}_{enc}) = \mathsf{data}_{sim}$
generate H_{rec} and
$\texttt{Enc}_{\texttt{ABE}}(\mathsf{H}_{rec}, \mathbb{A}) = \mathsf{H}_{rec\mathbb{A}}$
send $\mathsf{H}_{rec\mathbb{A}}$
hash h
generate
metadata md
Send transaction with
$(h||md)$
IF
addresses
[sender = owner]
record data
addresses
[sender $\neq$ owner]
cancel process

FIGURE 5. Data processing: formatting of data received from monitoring devices, administrator device authentication, and ABE-based data sharing.

The idea of using the Arduino Nano is to show that our scheme is capable of running on most current mHealth devices. This is confirmed by the results obtained here, which show that our scheme consumes fewer resources than the minimum suggested in many reference projects for the mHealth industry [60]–[62]. The results of Table 1, Table 2 and Table 3 allow us to say that our security scheme is suitable for resources limited mHealth devices.

VI. DATA ADMINISTRATION SUBSYSTEM

In this section, we present the details of the administration layer. Here, the data owner uses the smartphone to receive the data from the monitoring devices. Once the authenticity of the monitoring device, that sent the package pac_{nizkp} is verified, the administrator device can execute Protocol 4 to process the data. The workflow for this subsystem is illustrated in Fig. 5.

Protocol 4: Data Processing

- ***Goal:*** *processing and sharing of data.*
- ***Players:*** *the administrator device $\mathcal{D}_{adm}$ and the smart contract.*
- ***Common input:*** *the package pac_{nizkp}.*
- ***Private input:*** *shared secret key S_a of $\mathcal{D}_{adm}$.*

Steps:

1) *$\mathcal{D}_{adm}$ decrypts the package data_{enc} running the algorithm $\texttt{Dec}_{\texttt{AES}}(S_a, \mathsf{data}_{enc}) = \mathsf{data}_{sim}$, where $\texttt{Dec}_{\texttt{AES}}$ is the symmetric decryption algorithm, S_a is the shared key generated by the Protocol 1 and data_{sim} is the decrypted data package.*
2) *$\mathcal{D}_{adm}$ uses data_{sim} to generate a file H_{rec} that corresponds to a health record.*

3) *$\mathcal{D}_{adm}$ encrypts H_{rec} using an ABE algorithm for a given access policy $\mathbb{A}$, such as $\mathtt{Enc}_{\mathtt{ABE}}(\mathsf{H}_{rec}, \mathbb{A}) = \mathsf{H}_{rec\mathbb{A}}$.*
4) *$\mathcal{D}_{adm}$ sends the encrypted file $\mathsf{H}_{rec\mathbb{A}}$ to the IPFS network.*
5) *IPFS generates a hash $\mathtt{h}$ for the uploaded file and returns $\mathtt{h}$ to $\mathcal{D}_{adm}$. In IPFS, the hash is used as the location of the file.*
6) *$\mathcal{D}_{adm}$ generates the metadata $\mathtt{md}$ corresponding to the health record H_{rec}. Metadata will be used to search health records by healthcare professionals.*
7) *$\mathcal{D}_{adm}$ sends a blockchain transaction carrying the following information:*
    a) *hash $\mathtt{h}$, which represents $\mathsf{H}_{rec\mathbb{A}}$ on the IPFS;*
    b) *metadata $\mathtt{md}$ linked with the health record.*
8) *After receiving the transaction, the smart contract verifies that the transaction was sent by $\mathcal{D}_{adm}$, comparing the sender's address with the contract owner's address.*
    - *If the addresses are the same, then $\mathcal{D}_{adm}$ is the legitimate device; and then the data (metadata and file hash) are recorded on the blockchain.*
    - *If not, the data are rejected.*

A. EXPERIMENTS WITH BLOCKCHAIN

The experiments in this section are intended to evaluate the operations involving the administrator device and the blockchain. We consider two aspects for evaluation: (1) the time spent by the administrator device to perform the operations described in Table 4; and (2) the Ether cost of each transaction in Table 5.

When evaluating the time cost of operations, we do not consider communication costs between the administrator device and the blockchain. The reason for this is that we use the Ethereum blockchain, where we have no control over the processing time of its operations. Thus, we focus on the impact that our approach has on administrator device.

1) EXPERIMENTS SETUP

The environment of the experiments involves the following aspects:

- **Blockchain platform.** We use the Ethereum[1] blockchain to conduct the experiments. For the interactions between the administrator device and the blockchain, we used the Rinkeby[2] network, an Ethereum tool for testing and development. It allows calls to the blockchain at no financial cost for transactions.
- **Implementation.** We developed the mHealth administrator application using the Android platform. To implement the operations related to Attribute-Based Encryption, we use the libraries java `cpABE` [63] and `jPBC` [64]. We developed the smart contract, which operates as a legitimacy checker, using the Solidity[3] language. The smart contract is deployed on the Ethereum blockchain by sending the transaction *deploy()*. All interactions between the Android application and the smart contract were implemented using the `web3j`[4] library. Although the smart contract address is public, only the administrative device can write data to the blockchain, according to Protocol 4. In order to interact with the IPFS network, our application uses the `IPFS − lite`[5] library to instantiate and run an IPFS client. Thus, it is possible to send the data and receives the hash of the file sent in return.
- **Hardware.** We use an Android 9 smartphone with a quad-core 1.8 GHz processor and 4GB of RAM as an administrator device.

TABLE 4. Time spent by the administrator device to perform operations on the data management subsystem.

Operation	Average Time
The time required to encrypt data using ABE	0.912 s
The time required to generate the metadata	0.054 s
The time required to deliver data to the IPFS network	0.239 s

TABLE 5. Estimated cost by transaction.

Operation	Gas Used	Price (Ether)
Deployment of the smart contract	971,548	0.0039833468
Sending data	1,055,691	0.0043283350

2) RESULTS

Table 4 shows the average time spent by the device administrator to perform each of the operations related to data processing or sharing. Table 5 shows an estimate of the cost, incurred by the patient, to execute the blockchain transactions. Although we have presented the price of transactions in Ether, it is possible to convert that price to the dollar or another currency.

VII. DATA ACCESS SUBSYSTEM

In our approach, the player who wants to access patient data is equipped with an application capable of interacting with the smart contract on the blockchain. The smart contract verifies the legitimacy of the user.

A. HEALTHCARE PROFESSIONAL AUTHENTICATION

In this section, we will refer to healthcare professionals as *data users*, which is the term traditionally used in ABE-based schemes. Our approach ensures that patient data is shared only with duly authorized users. To do this, the scheme that authenticates users consists of two phases: the user registration phase (see Fig. 6) and the authentication phase (see Fig. 7).

[1] https://ethereum.org
[2] https://www.rinkeby.io/
[3] https://solidity.readthedocs.io
[4] https://github.com/web3j/web3j
[5] https://github.com/textileio/android-ipfs-lite

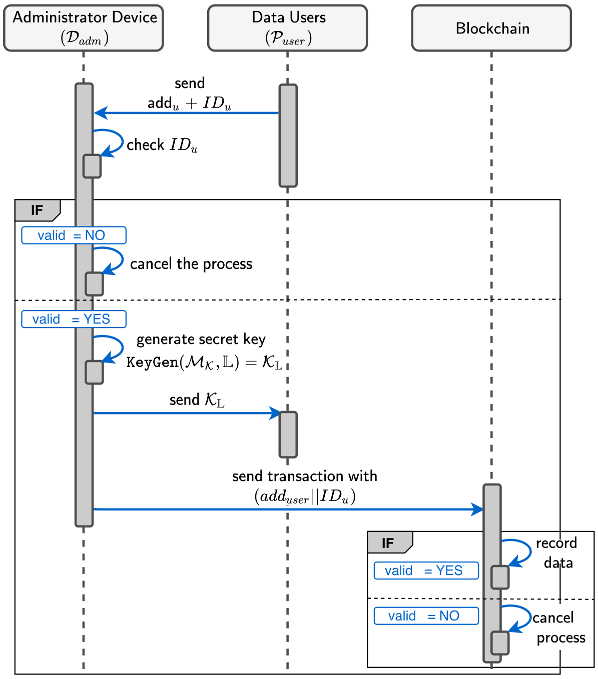

Administrator Device ($\mathcal{D}_{adm}$)
Data Users ($\mathcal{P}_{user}$)
Blockchain
send $add_u + ID_u$
check ID_u
IF
valid = NO
cancel the process
valid = YES
generate secret key
$\mathtt{KeyGen}(\mathcal{M}_{\mathcal{K}}, \mathbb{L}) = \mathcal{K}_{\mathbb{L}}$
send $\mathcal{K}_{\mathbb{L}}$
send transaction with $(add_{user}||ID_u)$
IF
valid = YES
record data
valid = NO
cancel process

FIGURE 6. The workflow of the registration phase of data users.

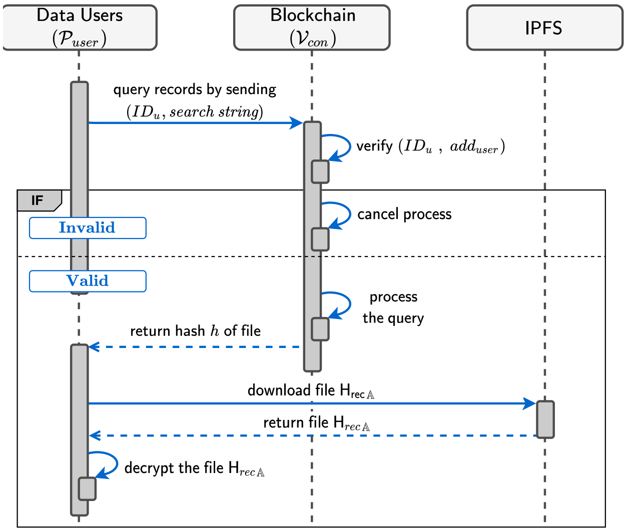

Data Users ($\mathcal{P}_{user}$)
Blockchain ($\mathcal{V}_{con}$)
IPFS
query records by sending
$(ID_u, search\ string)$
verify $(ID_u\ ,\ add_{user})$
IF
Invalid
cancel process
Valid
process the query
return hash h of file
download file $\mathsf{H}_{rec\,\mathbb{A}}$
return file $\mathsf{H}_{rec\,\mathbb{A}}$
decrypt the file $\mathsf{H}_{rec\,\mathbb{A}}$

FIGURE 7. The workflow of the authentication phase of data users.

1) USER REGISTRATION PHASE

Our approach requires an interactive step between the patient and the healthcare professional before the registration phase – a type of *pre-authentication*. In this interaction, the patient must transmit to the healthcare professional the address of the smart contract, denoted by $\mathtt{add}_{con}$, and the professional's identifier ID_u, randomly generated. Note that, usually, the first contact between the patient and the health professional is in person; therefore, the patient can share ID_u and $\mathtt{add}_{con}$ during this meeting. Then, the data owner registers the user according to the Protocol 5.

Protocol 5: Healthcare Professional Registration

- ***Goal:*** *register a data user.*
- ***Players:*** *data user $\mathcal{P}_{user}$; administrator device $\mathcal{D}_{adm}$; and smart contract.*
- ***Secret input:*** *private key $\mathcal{M}_{\mathcal{K}}$, generated in the initial system configuration.*

Steps:

1) *$\mathcal{P}_{user}$ sends his/her blockchain address $\mathtt{add}_{user}$ and his/her ID_u received from the patient previously.*
2) *$\mathcal{D}_{adm}$ checks if ID_u received is the same as the one sent in the pre-authentication step.*
    - *If true, the process continues at step 3.*
    - *Otherwise, the registration process is canceled.*
3) *$\mathcal{D}_{adm}$ generates the user's secret key by running the algorithm $\mathtt{KeyGen}(\mathcal{M}_{\mathcal{K}}, \mathbb{L}) = \mathcal{K}_{\mathbb{L}}$; which takes the master key $\mathcal{M}_{\mathcal{K}}$ and the attribute set $\mathbb{L}$ as input.*
4) *$\mathcal{D}_{adm}$ sends to $\mathcal{P}_{user}$, via a secure channel, the secret key $\mathcal{K}_{\mathbb{L}}$. Note that at this stage, both $\mathcal{P}_{user}$ and $\mathcal{D}_{adm}$ have the computational power to use a communication channel that implements, for example, secure asymmetric encryption techniques.*
5) *$\mathcal{D}_{adm}$ sends a transaction to the smart contract, in order to record the user registration data, containing the user's ID_u and $\mathtt{add}_{user}$.*
6) *After receiving the transaction, the smart contract verifies that the transaction was sent by $\mathcal{D}_{adm}$ (as in Protocol 4).*
    - *If true, user data is recorded in the smart contract.*
    - *If not, data are rejected.*

2) DATA ACCESS PHASE

In this phase, the users duly registered by the patient can access the data. Healthcare professionals can monitor patient health through applications available on their computing devices. In each access request, a user must be authenticated according to Protocol 6.

Protocol 6: Data Access

- ***Goal:*** *allow access to data to monitor patient health.*
- ***Players:*** *data user $\mathcal{P}_{user}$; smart contract; IPFS network.*
- ***private input:*** *user's secret key $\mathcal{K}_{\mathbb{L}}$ for the ABE scheme.*

Steps:

1) *To search for patient health records, $\mathcal{P}_{user}$ sends a blockchain transaction containing ID_u and the search string:*
2) *The smart contract checks if the sender's address and ID correspond to any pair $(\mathtt{add}_{user}, ID_u)$ stored in the smart contract. If so, the process continues at step 3. Otherwise, the process is canceled.*
3) *The smart contract processes the query and returns the hash h of all records matching the search.*
4) *$\mathcal{P}_{user}$ downloads the file $\mathsf{H}_{rec\,\mathbb{A}}$ corresponding to h.*
5) *$\mathcal{P}_{user}$ decrypts the health record using algorithm $\mathtt{Dec}_{\mathtt{ABE}}(\mathsf{H}_{rec\,\mathbb{A}}, \mathcal{K}_{\mathbb{L}}) = \mathsf{H}_{rec}$, which takes the encrypted file $\mathsf{H}_{rec\,\mathbb{A}}$ and the secret key $\mathcal{K}_{\mathbb{L}}$ as input. The output is*

the file H_{rec}. But only if $\mathbb{L}$ satisfies $\mathbb{A}$, where $\mathbb{L}$ is the set of attributes associated with the user's key $\mathcal{K}_{\mathbb{L}}$.

After examining the patient's health record, the healthcare professional may need to register observations or guidelines for the patient. In this case, the professional sends a message to the patient through a secure channel. The patient analyzes whether the message received will be registered in the system or not. If he/she decides to register, then he/she runs the data processing protocol (Protocol 4) from step 2. In this way, the professional's message is available to all users of the system, as long as their attributes satisfy the health record access policy.

VIII. DISCUSSION

In this section, we will discuss each layer of our approach to point out its impacts.

The data collection layer (Section V) consists of preparing and transmitting the data. Note that we are not interested in the type or form in which health data is collected. We are interested when the data is already available for transmission. For the first part of the approach, we present an authentication scheme for the monitoring devices. The scheme is based on NIZKP, having the ECDLP as the mathematical problem. With this, we achieve the following outcomes:

- An exclusive association between the monitoring device and the mHealth application; this is, a monitoring device can be paired only with the official mHealth application. Thus, the scheme prevents a malicious application, eventually installed on the administrator device, from communicating with the monitoring device and stealing the data. Besides, we can also prevent an illegitimate device from injecting false data into the system. This improvement is because the fake device is not able to discover the private key of a legitimate device to perform authentication.
- Data traffic between the monitoring device and the smartphone is symmetrically encrypted. The algorithm implemented in the monitoring devices encrypts the data, and only the mHealth application can decipher it. This means that a malicious application, even if it can establish communication with the monitoring device, will receive the stream of encrypted data; therefore, it will not be able to read it. As this scenario requires sharing a secret key between the mHealth application and the device, we propose using the ECDH protocol to generate and share the key.
- As mentioned earlier, ECC offers the same level of security when compared to other asymmetric encryption systems, using a significantly smaller key. Our scheme uses a 256-bit key only to provide the same security level as RSA with a 3072-bit key [19], [20]. Even with a high level of security, we were able to implement this scheme on resource-limited devices, requiring low execution time and little memory space, as shown in the results of Section V-C2.

Note that the security of the proposed scheme, for the data collection layer, is based on the difficulty of solving the ECDLP. Thus, respecting the elliptic curve parameters, as recommended by SECG [58], no algorithm solves the ECDLP in polynomial time. Thus, with the results shown above, we can say that our authentication scheme solves the security problems presented in Section II.

At the data administration layer, we employ a combination of ABE, blockchain, and IPFS. This combination results in an efficient administration of the system by the patient. More specifically, we achieve the following outcomes on this layer:

- We eliminate the need for a trusted third party, which is very common in cryptographic systems, responsible for generating and distributing the encryption/decryption keys to users. The administrator device assumes the role of the trusted authority. In this way, the patient himself/herself can generate the keys and allow access to the data only to the desired users.
- We assume that a particular administrator device controls the system. Therefore, to prevent other devices from writing data to the blockchain, the system requires authentication of the administrator device before allowing health data storage.

In the data storage layer, we chose to include a decentralized storage system, which involves the blockchain and the IPFS network. With that, we achieve the following:

- We have eliminated the problem of single point of failure, which is one of the biggest concerns in traditional centralized storage systems.
- The data is stored encrypted using the ABE scheme. Thus, we were able to guarantee fine-grained access control. Indeed, the data owner chooses who can access and what data can be accessed based on an access policy.
- The data cannot be changed or deleted due to the immutability property. Each transaction stored in the blockchain has a corresponding hash, and a Merkle tree is generated from the hashes of the transactions included in the block. The Merkle tree's hash value is stored in the block header together with a timestamp and the hash of the previous block. Therefore, if an attacker wants to tamper with a record in the blockchain, he/she needs not only to modify the hash of the block, but also to modify the hash of all subsequent blocks which are nearly impossible to achieve [65]. Note that, with the guarantee of immutability and ABE scheme, our approach eliminates the risk of data being unduly exposed or tampered with in the event of attacks.

In the data access layer, a user who wants to access the data must obey the following mechanisms:

- The first step is to search on metadata associated with health records stored in the blockchain. The smart contract is responsible for ensuring the legitimacy of the data users.
- Once authenticated, the user obtains the encrypted health record using the ABE scheme. To decipher the

record, the user needs a decryption key that satisfies the access policy defined by the data owner. Note that, even if an attacker randomly gets the hash that identifies a health record in IPFS, the file cannot be decrypted without the decryption key that satisfies the access policy.

Our proposal results in a system that guarantees the patient's privacy from end to end, that is, from collection to data storage. All players in the system must go through an authentication process. In the case of monitoring devices, the authentication process is done on the smartphone. In the case of data users, authentication is done in the smart contract on the blockchain. In case of attacks, an attacker would only be able to subvert the authentication scheme if he/she can resolve ECDLP. However, this is considered a computationally hard problem, and there is no polynomial-time algorithm to solve it.

IX. CONCLUSION

We propose an approach for mHealth systems integrated with the blockchain that offers a high level of security and guarantee of patient privacy. We present an authentication scheme that associates each monitoring device exclusively with the official mHealth application. With this, we eliminate the risk of spoofed devices or malicious applications infiltrating the system. The experiments show that our NIZKP-based scheme over ECDLP is safe and, at the same time, sufficiently lightweight to run on resource-limited devices.

Our access control scheme, based on ABE and integrated with the blockchain, results in a significant improvement in patient privacy. To access data, users submit to two levels of security. Initially, an authentication process is performed by a smart contract on the blockchain. Once authenticated, users obtain the encrypted data and only decrypt it with a secret key that satisfies the patient's access policy. Our approach eliminates the need for a trusted central authority. Here, the patient's administrator device is responsible for generating and distributing secret keys to users duly registered in the system. This management method provides full control of the system to the data owner.

As the data collected by a mHealth system is extremely sensitive, some security and privacy requirements are essential; these include confidentiality, integrity, access control, availability, and patient-centered data control. We proposed solutions for all these security requirements to address the challenges of privacy-preserving in mHealth systems.

REFERENCES

[1] M. Naveed, X. Zhou, S. Demetriou, X. Wang, and C. A. Gunter, "Inside job: Understanding and mitigating the threat of external device mis-binding on android," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014.

[2] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

[3] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] M. Yang, T. Zhu, K. Liang, W. Zhou, and R. H. Deng, "A blockchain-based location privacy-preserving crowdsensing system," *Future Gener. Comput. Syst.*, vol. 94, pp. 408–418, May 2019.

[5] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[6] W. Liu, H. Liu, Y. Wan, H. Kong, and H. Ning, "The yoking-proof-based authentication protocol for cloud-assisted wearable devices," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 469–479, Jun. 2016.

[7] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K.-K.-R. Choo, and Y. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 1310–1322, Jul. 2018.

[8] S. Liu, S. Hu, J. Weng, S. Zhu, and Z. Chen, "A novel asymmetric three-party based authentication scheme in wearable devices environment," *J. Netw. Comput. Appl.*, vol. 60, pp. 144–154, Jan. 2016.

[9] X. H. Le, M. Khalid, R. Sankar, and S. Lee, "An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare," *J. Netw.*, vol. 6, no. 3, pp. 355–364, Mar. 2011.

[10] C. Huang, K. Yan, S. Wei, and D. H. Lee, "A privacy-preserving data sharing solution for mobile healthcare," in *Proc. Int. Conf. Prog. Informat. Comput. (PIC)*, Dec. 2017, pp. 260–265.

[11] A. Roehrs, C. A. da Costa, and R. da Rosa Righi, "OmniPHR: A distributed architecture model to integrate personal health records," *J. Biomed. Informat.*, vol. 71, pp. 70–81, Jul. 2017.

[12] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, p. 218, Oct. 2016.

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

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

[15] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, p. 326, Jan. 2019.

[16] P. Genestier, S. Zouarhi, P. Limeux, D. Excoffier, A. Prola, S. Sandon, and J.-M. Temerson, "Blockchain for consent management in the ehealth environment: A nugget for privacy and security challenges," *J. Int. Soc. Telemedicine eHealth*, vol. 5, p. GKR-e24, Apr. 2017.

[17] 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)*, Oct. 2017, pp. 1–5.

[18] B. M. C. Silva, J. J. P. C. Rodrigues, F. Canelo, I. M. C. Lopes, and J. Lloret, "Towards a cooperative security system for mobile-health applications," *Electron. Commerce Res.*, vol. 19, no. 3, pp. 629–654, Sep. 2019.

[19] M. Bafandehkar, S. M. Yasin, R. Mahmod, and Z. M. Hanapi, "Comparison of ECC and RSA algorithm in resource constrained devices," in *Proc. Int. Conf. IT Converg. Secur. (ICITCS)*, Dec. 2013, pp. 1–3.

[20] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery, "On the security of 1024-bit RSA and 160-bit elliptic curve cryptography," Cryptol. ePrint Arch., Tech. Rep. 2009/389, 2009. [Online]. Available: https://eprint.iacr.org/2009/389

[21] 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.

[22] 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.

[23] Q. Li, H. Zhu, J. Xiong, R. Mo, Z. Ying, and H. Wang, "Fine-grained multi-authority access control in IoT-enabled mHealth," *Ann. Telecommun.*, vol. 74, nos. 7–8, pp. 389–400, Aug. 2019.

[24] 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)*, Dec. 2017, pp. 1–6.

[25] R. C. Lunardi, R. A. Michelin, C. V. Neu, and A. F. Zorzo, "Distributed access control on IoT ledger-based architecture," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–7.

[26] N. Vithanwattana, G. Mapp, and C. George, "MHealth–investigating an information security framework for mHealth data: Challenges and possible solutions," in *Proc. 12th Int. Conf. Intell. Environ. (IE)*, Sep. 2016, pp. 258–261.

[27] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Gener. Comput. Syst.*, vol. 90, pp. 62–78, Jan. 2019.

[28] J. J. Hathaliya and S. Tanwar, "An exhaustive survey on security and privacy issues in healthcare 4.0," *Comput. Commun.*, vol. 153, pp. 311–335, Mar. 2020.

[29] T. McGhin, K.-K.-R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, Jun. 2019.

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

[31] T. Nguyen Gia, M. Jiang, V. K. Sarker, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 1765–1770.

[32] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, and S. Lee, "Health fog: A novel framework for health and wellness applications," *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, Oct. 2016.

[33] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.

[34] L. Yang, Q. Zheng, and X. Fan, "RSPP: A reliable, searchable and privacy-preserving e-healthcare system for cloud-assisted body area networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[35] K. Zhang, K. Yang, X. Liang, Z. Su, X. Shen, and H. H. Luo, "Security and privacy for mobile healthcare networks: From a quality of protection perspective," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 104–112, Aug. 2015.

[36] D. Sathya and P. Ganesh Kumar, "Secured remote health monitoring system," *Healthcare Technol. Lett.*, vol. 4, no. 6, pp. 228–232, Dec. 2017.

[37] J. Vora, P. DevMurari, S. Tanwar, S. Tyagi, N. Kumar, and M. S. Obaidat, "Blind signatures based secured E-Healthcare system," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2018, pp. 1–5.

[38] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions," *Future Gener. Comput. Syst.*, vol. 97, pp. 512–529, Aug. 2019.

[39] J. Benet, "IPFS–content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*. [Online]. Available: http://arxiv.org/abs/1407.3561

[40] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, and S. Shamshirband, "Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications," *Egyptian Informat. J.*, vol. 18, no. 2, pp. 113–122, Jul. 2017.

[41] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Netw.*, vol. 17, no. 1, pp. 1–18, Jan. 2011.

[42] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[43] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2005, pp. 457–473.

[44] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 89–98.

[45] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[46] J. Zhang, X. A. Wang, and J. Ma, "Data owner based attribute based encryption," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2015, pp. 144–148.

[47] Y. Song, H. Wang, X. Wei, and L. Wu, "Efficient attribute-based encryption with privacy-preserving key generation and its application in industrial cloud," *Secur. Commun. Netw.*, vol. 2019, pp. 1–9, May 2019.

[48] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conf. Appl. Cryptograph. Techn.* Springer, 1985, pp. 417–426.

[49] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[50] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[51] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[52] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM J. Comput.*, vol. 18, no. 1, pp. 186–208, Feb. 1989.

[53] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 1987, pp. 186–194.

[54] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, no. 3, pp. 161–174, Jan. 1991.

[55] F. Hao. *Schnorr Non-Interactive Zero-Knowledge Proof.* Accessed: 2017. [Online]. Available: https://tools.ietf.org/html/rfc8235

[56] Secp256k1. *Bitcoin WiKi*. Accessed: 2019. [Online]. Available: https://en.bitcoin.it/wiki/Secp256k1

[57] K. MacKay. *Micro-ECC*. Accessed: 2017. [Online]. Available: https://www.arduinolibraries.info/libraries/micro-ecc

[58] Standards for Efficient Cryptography Group. *Recommended Elliptic Curve Domain Parameters*. Accessed: 2010. [Online]. Available: https://www.secg.org/sec2-v2.pdf

[59] S. R. Moosavi, T. N. Gia, A.-M. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, and H. Tenhunen, "SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways," *Procedia Comput. Sci.*, vol. 52, pp. 452–459, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915008133, doi: 10.1016/j.procs.2015.05.013.

[60] Texas Instruments. *Blood Pressure Monitor*. Accessed: 2020. [Online]. Available: https://www.ti.com/solution/blood-pressure-monitor?variantid=33745&subsystemid=14125

[61] Texas Instruments. *Electronic Thermometer*. Accessed: 2020. [Online]. Available: https://www.ti.com/solution/electronic-hermometer?variantid=t34261&subsystemid=25456

[62] Microchip. *Wearable Heart Rate Monitor*. Accessed: 2020. [Online]. Available: https://www.microchip.com/design-centers/medical/applications/wearable-activity-monitors/design-files-demo-boards/wearable-heart-rate-monitor-demo

[63] J. Wang, *Java Realization For Ciphertext-Policy Attribute-Based Encryption*. Github, 2012. [Online]. Available: https://github.com/junwei-wang/cpabe/

[64] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. 16th IEEE Symp. Comput. Commun. (ISCC)*. Corfu, Greece, Jun./Jul. 2011, pp. 850–855. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/

[65] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Gener. Comput. Syst.*, vol. 96, pp. 185–195, Jul. 2019.

ANTONIO EMERSON BARROS TOMAZ received the B.Sc. degree in computer science from the State University Vale do Acaraú, Brazil, in 2007, and the M.Sc. degree in teleinformatic engineering from the Federal University of Ceará (UFC), Fortaleza, Brazil, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include privacy preserving, network security, the Internet of Things, and Blockchain.

JOSÉ CLÁUDIO DO NASCIMENTO received the B.Sc. degree in electrical engineering, the M.Sc. degree in teleinformatic engineering, and the Ph.D. degree in teleinformatic engineering from the Universidade Federal do Ceará (UFC), Brazil, in 2005, 2006, and 2009, respectively. He is currently a Professor of quantum information and optics communications with the Electric Engineering Department, UFC - Campus Sobral, Brazil.

ABDELHAKIM SENHAJI HAFID (Member, IEEE) spent several years as a Senior Research Scientist with Bell Communications Research (Bellcore), NJ, USA, working in the context of major research projects on the management of next-generation networks. He was also an Assistant Professor with Western University (WU), Canada; the Research Director of the Advanced Communication Engineering Center (venture established by WU, Bell Canada, and Bay Networks), Canada; a Researcher with CRIM, Canada; a Visiting Scientist with GMD-Fokus, Germany; and a Visiting Professor with the University of Evry, France. He is currently a Full Professor with the University of Montreal. He is also the Founding Director of the Network Research Laboratory and Montreal Blockchain Laboratory. He is a Research Fellow with CIRRELT, Montreal, Canada. He has extensive academic and industrial research experience in the area of the management and design of next-generation networks. His current research interests include the IoT, fog/edge computing, blockchain, and intelligent transport systems.

JOSÉ NEUMAN DE SOUZA (Senior Member, IEEE) received the Ph.D. degree from Pierre and Marie Curie University (PARIS VI/MASI Laboratory), France, in 1994. From 1999 to 2005, he was a Board Member (Directory) of the Computer Networks National Laboratory. He has been a First Class Invited Professor UMR CNRS 8144, PRISM, Universite de Versailles Saint Quentin-en-Yvelines, France, in 2001; also with UMR CNRS 7030, LIPN, Universite de Paris 13, France, in 2005, 2006, 2008, and 2009, respectively; also with the IMAGINE Lab, University of Ottawa, ON, Canada, in 2007; and also with the IBISC Laboratory, Universite d'Evry Val d'Essonne, Evry, France, in 2011. He has been a CNRS Invited Researcher with the LABRI Laboratory, Bordeaux 1 University, France, in 2010. He spent a year (2008–2009) at the National Laboratory for Scientific Computing, Petropólis, Brazil, developing Senior Postdoctoral activities. Since 1999, he has been the Brazilian Representative at the IFIP TC6 (Communication Systems). He is currently a Researcher Full Professor with the Computer Science Department, Federal University of Ceara. He is also involved in the teleinformatics engineering course as a Lecturer and a Researcher with the Teleinformatics Engineering Department.