Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8876807, 16 pages
https://doi.org/10.1155/2020/8876807

Research Article

Low-Cost Authentication Protocol for D2D Communication in m-Health with Trust Evaluation

Ana Paula G. Lopes and Paulo R. L. Gondim

Electrical Engineering Department, University of Brasilia, Brasília, Distrito Federal, Brazil

Correspondence should be addressed to Paulo R. L. Gondim; prgond@gmail.com

Received 24 May 2020; Revised 9 August 2020; Accepted 16 September 2020; Published 27 October 2020

Academic Editor: Yujin Lim

Relay-assisted device-to-device (D2D) communication serves users at the edge of system coverage of 5G networks, enabling communication among sensors and patients' mobile devices, and improving spectral and power efficiency. The security of D2D-based m-health applications requires attention due to the delicacy of the data treated in the collection, transmission, and storage of information on patients, whose devices must be adequately authenticated. However, traditional authentication and key agreement schemes are not suitable for D2D scenarios, since they might expose patients to security vulnerabilities and lead to an excessive use of resources. This article proposes a secure and lightweight scheme based on Shamir secret sharing for the mutual authentication of m-health devices in relay-assisted D2D communications, which provides security robustness and reduces resources (energy, processing) consumption. The manuscript also addresses the trustworthiness of devices involved in data relay and device discovery procedures.

1. Introduction

Mobile device communication has grown over the past few years due to the development of thousands of new applications and devices. The Internet of Things (IoT), the main responsible actor for such a revolution, enables the connection of several applications (e.g., those based on smartphones, smart watches, smart TVs, smart homes and vehicles, and smart metering). Mobile-health (m-health), which is an interesting human health-related application, provides the monitoring and evaluation of vital signs and other important health information on patients, preventing the escalation of diseases and affording immediate relief in emergencies.

The m-health system commonly works with a group of sensors coupled to a patient's body and a mobile device that receives the measurements from such sensors and sends the information to the respective health center. Huang et al. [1] observed high-quality healthcare services, such as remote monitoring, mobile telemedicine, remote disease diagnosis, and emergency care require the assurance of security of both the system and the communication channels through which messages are exchanged.

On the other hand, D2D communication refers to direct and low-power communication between two mobile devices [1]; it offers services based on their proximity, and its advantages include higher throughput, low latency, and instantaneous communications between devices [2]. Moreover, traffic offloading/traffic steering between cellular and D2D networks is an excellent alternative for the bandwidth demands imposed over cellular networks, increasing spectral efficiency, and reducing energy consumption [1].

Device-to-device communication (D2D) is a strong candidate for communication of devices involved in m-health applications. For example, in a scenario of remote telemonitoring of patients implemented on a large scale by cellular and wireless body area networks (WBAN), the high volume of data exchanged, jointly with concurrent data traffic from other applications, requires a new perspective on the communication of near devices for providing important health information on patients' health, 24 hours a day and seven days a week. As another example, in emergency care

situations in which information about a patient must be rapidly transmitted for the evaluation and acting by a health professional (a doctor, for example), D2D communications contribute to the transmission and reception of data in a timely manner.

Health services provided at the edge of communication networks, such as LTE (Long-Term Evolution) and LTE-A (LTE-Advanced) can benefit from technological resources and principles inherited from mobile edge computing (or, more recently, multiaccess edge computing (MEC)) [3–6] discussed the integration of MEC and D2D technologies. In particular, a possible use of relay D2D devices can promote coverage extension and a better removal of constraints related to computation resources, since the task of a device can be offloaded to an edge node and a nearby D2D device ([4]).

Nonetheless, Wang and Yan [2] highlighted the success of D2D communication depends on security, which has not been properly studied. D2D cannot work adequately to fulfill the application's expectations if security is not assured. Its requirements were addressed by Wang and Yan [2] and Haus et al. [7] and include authentication, privacy, anonymity, nonrepudiation, integrity, confidentiality, and resistance to attacks (e.g., man-in-the-middle, impersonation, and replay, among others).

Some of such security requirements might be fulfilled by a mutual authentication among devices and among devices and the core of the 3GPP (Third Generation Partnership Project) network. However, the traditional authentication and key agreement (AKA) standardized by 3GPP is not suitable for D2D authentication, since it leads to high consumption of processing and bandwidth resources as described by [8, 9]; moreover, D2D devices have commonly memory limitations and the energy (battery) consumption should also be reduced. Therefore, new applications that exchange critical data (e.g., m-health) require novel AKA schemes to fulfill such a demand [10].

This article proposes a novel mutual authentication and key agreement scheme (protocol) for D2D devices in m-health that enables patients to securely send their medical information to a health center and doctors. In comparison to other authentication protocols, our scheme reduces energy consumption and the use of processing resources.

An important issue regarding m-health is trust among devices supported by D2D communication. Whenever a patient must send data and no direct connection with the 3GPP infrastructure is provided, such data are sent through relay and device to device until the network infrastructure has been reached. The problem is that not all devices are trustworthy to perform such a task; consequently, trust assurance and evaluation become a critical problem for D2D m-health applications.

The protocol has been designed to forecast the relay of data when devices are outside the 3GPP coverage area, or inside it, but with no access to the network, regarding the necessity of computational trust.

The main contributions of the present study include the following:

(i) A secure secret sharing scheme for D2D m-health applications that fulfills all security aspects discussed in 3GPP D2D security specification TS 33.303 [11]

(ii) A mutual authentication scheme for D2D m-health groups of devices focused on the 3GPP architecture for proximity-based services

(iii) An adaptation of the trust mechanism based on the local trust concept proposed by Yan et al. [12] that enables D2D devices to choose the most reliable device in their proximity to perform the relay of their data; in the proposed protocol, the local trust secret key encryption is based on symmetric cryptography, thus reducing computational costs when compared with [12], which is based on asymmetric cryptography

(iv) An evaluation of computational and energy costs of our scheme, which revealed its superior performance, in comparison to two other protocols

(v) An assessment of the security properties of the scheme and possible protection against attacks and threats

(vi) A semiautomated formal validation of the protocol

The remainder of the manuscript is organized as follows: Section 2 discusses some related work. Section 3 presents the 3GPP reference architecture for proximity services. Section 4 introduces the protocol, its phases, the authentication and key agreement process, and a trust evaluation. Section 5 is devoted to a security analysis and comparisons with other protocols. Section 6 reports an evaluation of computational and energy costs. Section 7 discusses AVISPA verification. Section 8 provides the conclusions and suggests some future work.

2. Related Work

m-health security has been the focus of several studies. Zhang et al. [13] developed an efficient certificateless generalized signcryption (CLGSC) scheme based on the Elliptic Curve Discrete Logarithm Problem (ECDLP) and a lightweight and robust security-aware (LRSA) D2D-assisted data transmission protocol for m-health based on CLGSC. However, according to Zhou [14], the scheme proposed by [15] shows some security weaknesses, such as vulnerability to an insider attack, which affects its confidentiality. Zhou enhanced it by improving the CLGSC scheme and proposed a certificateless signcryption scheme for m-health [16], towards correcting the abovementioned vulnerabilities in CLGSC scheme. According to the author, it uses some extra variables in the authentication procedure in comparison to [15], which enables attackers to obtain some authentication parameters through queries.

Harn [15] presented three authentication schemes based on Shamir's secret sharing [17], which enables the generation of a common secret for a group of entities. According to Shamir's secret sharing, a previously established system manager chooses a random polynomial and generates a secret based on the secret tokens of each entity participating in the system. The tokens are then securely exchanged among the entities so that they reconstruct the secret through the Lagrange

interpolating formula and authenticate each other by comparing the secret generated with the secret received from the system manager.

Harn [15] designed the Asynchronous (t; m; n) Group Authentication Scheme (GAS) with Multiple Authentications, which authenticates n members of m groups and is resilient until t tokens have been compromised. Each entity has two tokens generated by the system manager through two different polynomials, which must remain secret. The system manager also generates a secret based on the tokens of the entities. Using its own two tokens, each member generates two Lagrange components, which are based on the Lagrange interpolating formula. The entities then exchange their Lagrange components to obtain a secret to be compared with that received from the system manager.

Mustafa and Philip [18] discussed the way a scheme of group key exchange for D2D medical IoT communication with cryptographic secret sharing must be designed to be efficient. Although it uses Shamir secret sharing [17], the authors do not detail the calculations and messages exchanged for the authentication of the devices, and only describe the procedure. A device is required to be a supernode that calculates the key generation process and distributes the key shares (tokens) to each device. The node is considered a single point of failure, since all devices rely upon it for the creation of the group-based session key. As future work, the authors propose the creation of a distributed key exchange approach. However, the development of a trust scheme for the D2D m-health environment has not been considered.

Yan et al. [12] designed a scheme for secure D2D communications that operates over the 3GPP infrastructure, based on two-dimensional trust levels, namely, Local Trust (LT), controlled by the communicating devices, and General Trust (GT), controlled by the 3GPP infrastructure. It considers D2D communication in general and presents the following three coverage scenarios: coverage, relay coverage, and out of coverage. The devices obtain support from ProSe Function Server (PFS) and ProSe App Server (PAS) to perform a trust evaluation. The scheme is composed of algorithms that authenticate and measure the trust level of devices in three situations, i.e., when only LT (local trust) level, or only GT (global trust), or both levels are used for the trust measurement, and has been partially used for the construction of our trust mechanism.

Last, but not least, we considered several technical reports and specifications of 3GPP regarding D2D communication and ProSe to strengthen the technical foundation of this study, including 3GPP TS 33.303 [11], 3GPP TS 23.303 [19], and TR 36.843 [20]. The former describes the security aspects to be considered when ProSe is used in the Evolved Packet System (EPS) and comprises the security procedures involving interfaces among network entities, the configuration of ProSe-enabled User Equipment (UE), and data transfer between ProSe Function and ProSe-enabled UE. The second specification [11] regards the ProSe features in EPS, i.e., ProSe discovery (identification of UEs in the proximity) and ProSe Direct Communication, which enables the establishment of communication paths between two or more UEs in a direct communication range. The technical specifications in [20] address enhancements for ProSe UE-to-network relay for commercial and public safety applications, as wearables and IoT devices.

3. 3GPP Reference Architecture for ProSe Services

3.1. Functional Description. Since the development of applications and equipment can benefit from the adoption of standardized architectures, we considered the architecture adopted by a normative organization (in this case, the 3GPP) for proximity-based services.

The following entities of the 3GPP reference architecture for ProSe services have been considered:

(i) Home Subscriber Server (HSS): part of the Evolved Packet Core (EPC) of LTE (Long-Term Evolution) networks that contains users' and subscribers' information, supports authentication and authorization of devices, and manages mobility

(ii) ProSe Function Server (PFS): the logical function used for network-related actions required for ProSe that plays different roles for each of its feature [11], such as generation of trust tokens and identities in the management of D2D communication

(iii) ProSe App Server (PAS): an entity that stores and manages ProSe User IDs and maintains permission information for restricted ProSe Direct Discovery

(iv) User Equipment (UE): a mobile device associated with each user

(v) Evolved NodeB (eNodeB): an entity that provides a wireless connection with UE and enables its connection with the core network

Figure 1 shows the reference architecture proposed by 3GPP for Proximity Services [20]. Domain A is inside the red dotted circle and comprises the security domain of EPC, PFS, and PAS. Domain B is defined by the lilac dotted circle and refers to the security domain of UE and PAS. Finally, Domain C defines the security domain comprised only of users' equipment.

3.2. Reference Points. Below is a list of reference points of 3GPP TS 23 303 [19], as shown in Figure 1:

(i) PC1: the reference point between the ProSe application in the UE and the ProSe Application Server that defines application-level signalizing requirements

(ii) PC2: the reference point (PC2) between the ProSe Function Server (PFS) and the ProSe Application Server (PAS) that defines the interaction between PFS and PAS. PFS receives a proximity request from an originating UE and sends a proximity map request to PAS to obtain the identity of the

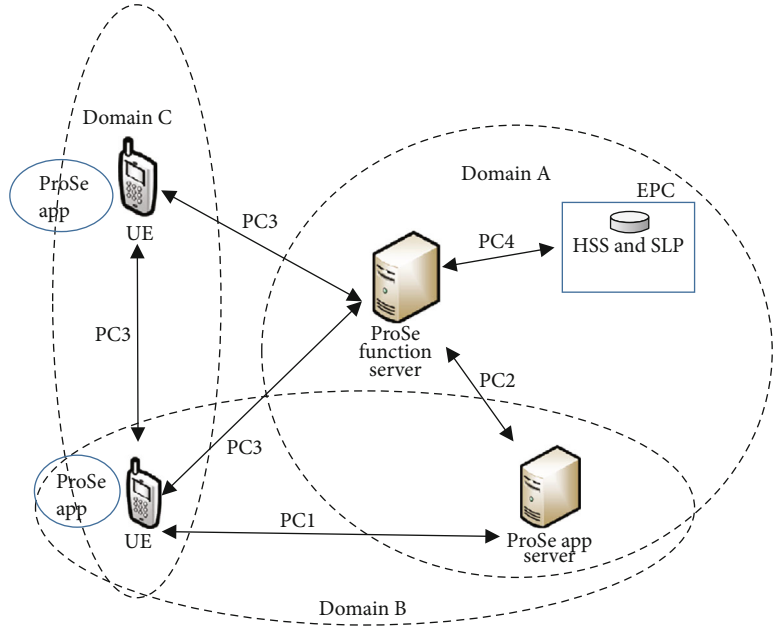

Figure 1: 3GPP ProSe reference architecture (based on Yan et al. [11]).

targeted application user. PAS determines whether the originating UE is allowed to discover the targeted UE

(iii) PC3: the reference point between the UE and the ProSe Function that authorizes discovery requests in the EPC level and allocates the identities used in discovery procedures

(iv) PC4: the reference point between HSS and PFS used by the latter to retrieve EPC-level discovery-related subscriber data

(v) PC5: the reference point between UEs used for control and user plane for direct discovery

3.3. 3GPP ProSe Device Discovery. The 3GPP device discovery is detailed in technical specification TS 23.303 [19] and involves the detection and identification of other devices (UEs) located in proximities using E-UTRAN or WLAN direct radio signals. The device discovery can be open if no permission is required by the UE being discovered, or restricted, otherwise. It can also be used by applications to initiate ProSe Direct Communication.

It has the following two models for operations:

(i) Model A ("I am here"): interested devices announce certain information in a predefined discovery interval, which could be used by devices nearby to obtain permission to discover their existence. They monitor the devices that showed interest in the messages, read, and process them

(ii) Model B ("who is there?"/"are you there"): devices transmit a request with the information on what they are interested in discovering. The addressed devices respond with information related to the source device's requests

Our scheme has adopted Model A of device discovery. First, each device must obtain authorization for direct discovery and direct communication from the PFS. Prior to announcing the information, they must send a discovery request to the PFS; if it succeeds, they can start announcing on the PC5 interface. Next, they send a request to the PFS to be authorized to monitor. If they succeed and have a Discovery Filter, they can start the monitoring. Finally, when the monitoring devices detect one or more devices that have matched the filter, they report them to the PFS. For a more detailed description, readers should consult 3GPP TS 23.303 [19].

3.4. Security Requirements. The several security requirements and aspects expected by the 3GPP standardization [19] for D2D communication that uses the ProSe architecture include the following:

(i) Avoidance of attacks: the systems must resist several attacks, e.g., replay and impersonation

(ii) Authorization of devices: the system must allow only currently authorized devices to be discovered by other UEs

(iii) Tracking of devices: the tracking of devices based on their discovery messages should be minimized

(iv) Authentication of devices and PFS: the devices involved must authenticate the source of the data received. UE and PFS must authenticate each other

(v) Integrity and confidentiality: both integrity and confidentiality of data exchanged among the entities must be guaranteed

(vi) Privacy: the privacy of users must be provided

4. Proposed Scheme

Our scheme considers situations in which devices are outside the coverage area, in the coverage area, and directly connected with the 3GPP infrastructure, or in the coverage area, but with no direct access to 3GPP. In the second case, D2D nodes operate as the relay of a network, as proposed by Zhang et al. [13] and Zhou [14]. Moreover, computational trust is fundamental for a proper operation of the system.

HSS manages the device authentication and key distribution, whereas PFS and PAS manage the trust of devices. D2D communication involves patient's devices willing to perform the relay of data. Finally, the health center infrastructure receives the patients' data and forwards them to doctors, nurses, and physicians. Figure 2 shows the architecture of the protocol, derived from 3GPP ProSe [11] standards, with all entities involved.

Table 1 shows the main symbols and parameters used in the proposal.

Some basic assumptions are as follows:

(i) The health center infrastructure is considered trustworthy and secure

(ii) The entities of the 3GPP infrastructure and their communication channels are considered trustworthy and secure

(iii) The channel between the patients' device and their respective body sensors is considered safe

(iv) The D2D communication channels and the channel between devices and eNB are considered unsafe (they are the focus of this study)

The domain considered covers one or more 3GPP cells. Several groups can be inside the system domain of operation and are formed according to the patients' needs regarding the sending of their data.

The protocol uses asymmetric and symmetric cryptography. The former is used in the generation of private keys and temporary identities for mutual authentication, while symmetric cryptography is employed in the trust evaluation for reducing costs, when compared to [12]. It is based on the Asynchronous (t; m; n) Group Authentication Scheme (GAS) with Multiple Authentications, proposed by Harn [15], since it provides a way of sharing a secret in a group of entities that might be used in the generation of secret keys. Timestamps and random variables are freshly generated in each session for avoiding attacks. A session key is generated among devices, as well as among devices and HSS at the end of the mutual authentication phase, and a local trust key is generated whenever a local trust evaluation is required from one device to another. New keys are generated at every single execution of the protocol.

The following 5 (five) phases, described in the next subsections, must be executed for a patient outside the coverage area to send their data in a protected manner:

(i) Initialization

(ii) Registration

(iii) Mutual authentication

(iv) Trust evaluation

(v) Encryption/decryption

4.1. Initialization. Some important system parameters are generated in this phase, and all devices accredited by the health center server must perform the phase offline.

Initially, HSS selects two random prime numbers p and q that satisfy condition $q/(p-1)$ and defines a finite field $Zp*$ and a secure elliptic curve $E(Zp*)$. Next, it selects a group Gp of order p, Gq that is a subgroup of Gp, g as the generation point of Gq, and $Zq*$ as a prime field of order q. Then, it selects a random number $z \epsilon Zq*$ as the master private key and calculates $M_{k\text{pub}} = z * g$ to obtain the master public key.

HSS selects three hash functions ($h1(.)$, $h2(.)$, and $h3(.)$) (described in Table 1) for the mutual authentication phase and generates j random numbers, Rj, ($j = 1, 2, \cdots, i$), for each device and for itself for the calculation of a set of temporary identities TID_{D_i} :

$$TID_{Di} = h_1\left(ID_{Di} \| R_j * z\right). \tag{1}$$

It also selects its own TID_{HSS}:

$$TID_{HSS} = h_1\left(ID_{HSS} \| R_j * z\right). \tag{2}$$

Next, it sends each device its respective set. A different TID_{D_i} is used whenever a new session has been established to provide a relay of data to a specific device. When the last TID available is being used, the device must notify the HSS after the authentication procedure. Then, HSS sends a new set of temporary identities encrypted with the freshly generated session key.

HSS generates a piece of each device's private key (similar to [15]), chooses a $y \epsilon Zq$, and calculates the following:

$$Y_{Di} = h_2(TID_{Di} \| y) * M_{Kpub}. \tag{3}$$

Finally, it sets the partial key calculated for each respective device and publishes the following parameters: $\{g, G, E(Zp*), M_{k\text{pub}}, \text{TID}_{\text{HSS}}, \text{h1}(.), h2(.), h3(.), H(.)\}$.

4.2. Registration. The ProSe device discovery mechanism is initially applied in this phase for the discovery of nearby devices, as described in [11]. The phase is performed over an insecure channel, and the main steps are described below.

Each user generates a share of its private key (based on [13]) choosing $x \epsilon Zq*$ and calculates its public key:

$$\text{PK}_{D_i} = x^* \, Y_{D_i} * g. \tag{4}$$

Then, it sends TID_{D_i}, PK_{D_i}, and a timestamp $T_{D_i}{}^{-1}$ to the other devices and the nearby device sends HSS all the information received from relay devices.

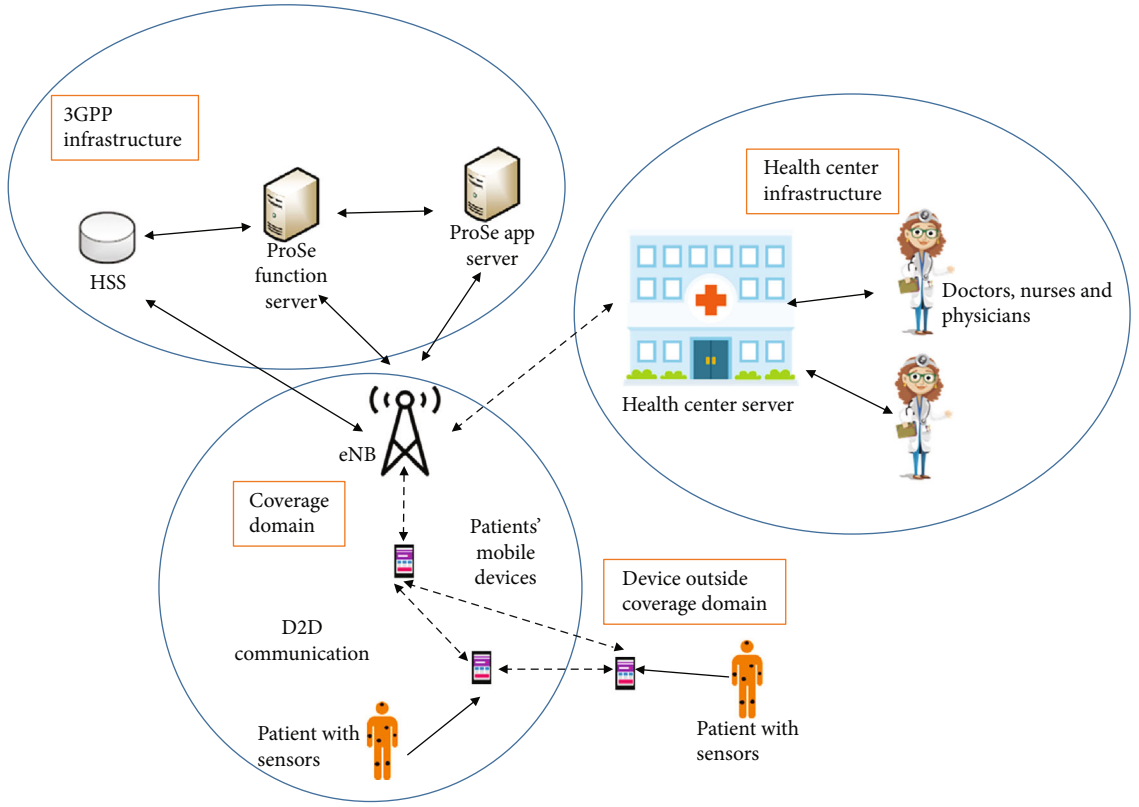

Figure 2: System architecture.

The device sets its private key as pair $SK_{D_i} = (x, Y_{D_i})$ using the other share of its private key received from the HSS in the initialization phase.

Each device chooses an integer $v_{D_i-D_j} \epsilon Zq*$ as a secret value to be sent to other devices and HSS, encrypted with its public key, as follows:

$$V_{D_i-D_j} = \mathrm{EPK}_{D_i}\left(v_{D_i-D_j}\right), \tag{5}$$

where j represents either a device D_j or HSS.

Consequently, only the correct device can decrypt the message and obtain the secret token. The secret values are broadcast to the entities involved in the communication, which find and decrypt them to obtain all secret values necessary for the generation of session keys.

The asynchronous mode of the group authentication protocol designed by Harn [15] is considered for providing multiple authentications in a t-secure m-user n-group authentication scheme (GAS). In other words, for a group with n members, m-users are authenticated at once, with at most $(t-1)$ compromised tokens; a unique token is assigned for each user of a group by the group manager for determining the membership of a user to a group. Therefore, considering what is proposed by [15], we have designed our authentication scheme as follows:

First, HSS selects two random polynomials $f_l(x), l = 1, 2$ of degree $t-1$ each, where $t \le n$ is the number of devices involved in the relay (i.e., number of tokens necessary for the recovery of secret S):

$$f_l(x) = \sum_{i=0}^{t-1} a_i x^i \mod p. \tag{6}$$

All coefficients are in finite field $Zp*$.

HSS generates two tokens for each device calculating $f_l(\mathrm{TID}_{D_i})$. Each TID_{D_i} has its respective token. HSS also calculates its own two tokens $f_l(\mathrm{TID}_{\mathrm{HSS}})$ and finds integers $w_{i,j}$, $d_{i,j}, j = 1, 2 \epsilon Zp*$, such that $S = \sum_{j=0}^{k} d_{i,j} f_j(w_{i,j})$, where $w_{i,1} \neq w_{i,2}$ for every pair i and j. It then generates a secret S, as in [17]:

$$S = f_l(0) = a_0, \tag{7}$$

$$S = g^{\sum_{j=1}^{2} d_{i,j} f_j(w_{i,j}) \mod q} \mod p. \tag{8}$$

Table 1: Parameters used in the protocol.

Symbol	Description
D_i	Patient i or device i, where $i = 1, 2, 3, \cdots, n$
p, q	Large public prime numbers
$Zp*/Zq*$	A finite field of order p/prime field of order q
$E(Zp)$	Elliptic curve over Zp
Gp	Group of order p
Gq	A subgroup of Gp of order q
g	Point generator of G; $i = 1, 2, 3 \cdots$
$f_l(x)$	Random polynomial; $l = 1, 2, \cdots$
z	Master private key
$M_{k\text{pub}}$	Master public key
SK_{D_i}	Private key of device D_i
PK_{D_i}	Public key of device D_i
ID_x/TID_x	Real identity of entity x/temporary identity of entity x
R_j	j random number generated
T_{x_i}	Timestamp generated by entity $x = D_i$, HSS; $i = 1, 2, 3 \cdots$
h_1	Temporary identity generation hash function; $H_1 : \{0,1\}* Z_p*$
h_2	Device's partial private key generation hash function; $H_2 : \{0,1\}* G_q$
h_3	Symmetric key generation hash function; $H_4 : \{0,1\}* \times G_q G_p$
H	Shamir's secret hash function
LT_i	Local trust level of device i
$\text{LTK}_{D_u - D_u'}$	Local trust secret key
	Secure channel
	Insecure channel

Finally, it chooses an integer $v_{\text{HSS}-D_i} \epsilon Zq*$ and sends it to the respective devices of the relay group:

$$\text{AUTH}_{D_i} = \left[\text{EPK}_{D_i}\left(\text{TID}_{\text{HSS}}, H(S), w_{i,j}, d_{i,j}, f_l\left(\text{TID}_{D_i}\right), v_{\text{HSS}-D_i}\right), \text{TID}_{D_i}, T_{\text{HSS}^-1}\right]. \tag{9}$$

The devices decrypt the message and store the parameters.

Figure 3 shows a summary of the registration phase.

4.3. Mutual Authentication. Since the devices still must authenticate each other and HSS, each device selects a pair of nonused TID_{D_i} and respective tokens $f_l(\text{TID}_{D_i})$, $l = 1, 2, \cdots$ and computes its Lagrange component (an adaptation of what is proposed in [15]) LC_{D_i} through the Lagrange interpolating formula:

$$\text{LC}_{D_i} = \sum_{l=1}^{2} d_{i,l} f_l\left(\text{TID}_{D_i}\right) \prod_{q=1;q\neq i}^{n} \frac{w_{i,l} - \text{TID}_{D_i - q}}{\text{TID}_{D_i} - \text{TID}_{D_i - q}} \bmod q. \tag{10}$$

Next, they calculate $e_{D_i} = g^{\text{LC}_{D_i}} \bmod p$.

HSS also calculates its Lagrange component LC_{HSS} through the Lagrange interpolating formula:

$$\text{LC}_{\text{HSS}} = \sum_{l=1}^{2} d_{i,l} f_l(\text{TID}_{\text{HSS}}) \prod_{q=1;q\neq i}^{n} \frac{w_{i,l} - \text{TID}_{D_i - q}}{\text{TID}_{\text{HSS}} - \text{TID}_{D_i - q}} \bmod q. \tag{11}$$

Its own e_{HSS} was also calculated:

$$e_{\text{HSS}} = g_i^{\text{LC}_{\text{HSS}}} \bmod p. \tag{12}$$

It generates a random value $r_{D_i} \epsilon Zq*$. The devices send TID_{D_i}, e_{D_i}, r_{HSS}, and a timestamp $T_{D_i^-2}$ to the other devices in the relay group and to HSS, which also sends TID_{HSS}, e_{HSS}, T_{HSS^-2}, and r_{HSS} to the other devices in the relay group. After receiving such parameters, the entities verify the validity of the timestamp to avoid denial of service (DoS) attack. They proceed with the authentication

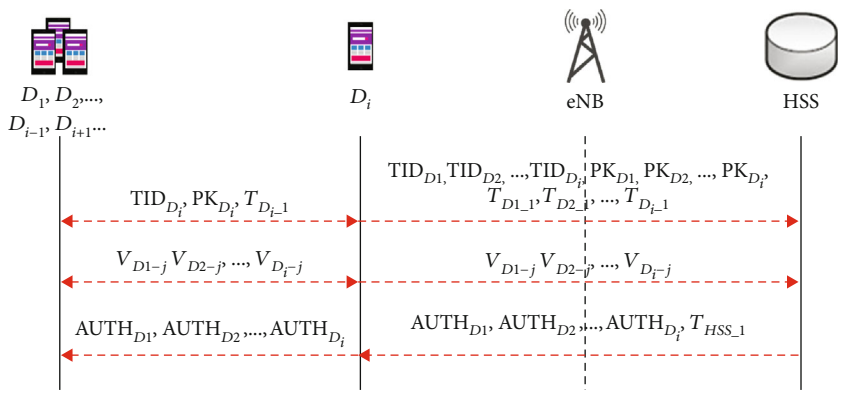

Figure 3: Messages exchanged in the registration phase.

only if the timestamp is valid. Otherwise, they discard the respective entity. When each entity has a complete set of e_{D_i} and e_{HSS}, a secret S' is calculated:

$$S' = \left(e_{\mathrm{HSS}} * \prod_{i=1}^{n} e_{D_i} \right) \bmod p = g_i^{\left(\mathrm{LC}_{\mathrm{HSS}} + \sum \mathrm{LC}_{D_i}\right) \bmod q} \bmod p. \tag{13}$$

Again, an attacker must solve the DLP problem to obtain $e_{D_i}, e_{\mathrm{HSS}}$, and S', as in [15].

Next, each device checks if the $H(S')$ calculated is equal to the $H(S)$ received from HSS in the registration phase. If $H(S) = H(S')$, the devices and HSS are legit and mutually authenticated. If the verification fails, one or more intruders are in the path.

Finally, a session key is generated for each possible connection between devices D_i and HSS.

$$SKey_{Di-k} = h_3[S \| r_{Di} \| r_u \| v_{Di-u} \oplus v_{u-Di}] \tag{14}$$

In this stage, if the source device has direct access to the network infrastructure, it can encrypt its health information with the session key and send it to the core network. Otherwise, it must execute phase 3.4 prior to phase 3.5.

Figure 4 shows a summary of the mutual authentication phase.

4.4. Trust Evaluation. This phase is executed whenever a patient that must send his/her health information to the doctor/physician is not in the coverage area of a 3GPP cell. Therefore, data must be relayed through other D2D devices available, until a device with a direct connection to the network infrastructure has been reached.

Due to the delicacy of the data exchanged, the trust level of each node authenticated in the mutual authentication phase must be measured before the data are sent. The trust evaluation enables the origin device to choose the path with the most reliable devices available for the relay of data. The trust system adopted follows the same idea of local trust presented by Yan et al. [12]. However, we have created our own calculations, which differ from those of [12], due to the use of symmetric cryptography for reducing costs.

This phase is performed over an insecure channel. An architecture involving the use of relay devices, as shown in Figure 2, is employed. After the measurement of local trust, all devices considered trusted are candidates to be relay devices.

4.4.1. Local Trust Evaluation. The local trust evaluation is based on the experiences of nearby devices. Each device defines a trust threshold for deciding whether the devices are trustworthy or not.

When a device D_u wants to know if a device $D_{u'}$ is trustworthy, it compares the $\mathrm{LT}_{D_u}{}'$ level with the desired threshold LT. If it is higher than the threshold, device $D_{u'}$ is considered trustworthy, and device D_u can relay data through it. Otherwise, the communication is refused.

Whenever a device D_u wants to obtain $\mathrm{LT}_{D_u}{}'$ of a device $D_{u'}$, it sends $\mathrm{TID}_{D_u}{}'$ to another device D_k, which once communicated with device $D_{u'}$, to request its local trust evaluation $\mathrm{LT}_{D_{u'}}$.D_k generates local trust level $\mathrm{LT}_{D_{u'}}$ of device $D_{u'}$, encrypts the result with the session key generated between D_k and D_u, and sends it to D_u and $D_{u'}$:

$$B_{D_k-D_u} = ESKey_{D_k-D_u}\left(\mathrm{LT}_{D_{u'}}\right). \tag{15}$$

D_u decrypts the message and obtains the local trust level of $D_{u'}$. It then checks if $\mathrm{LT}_{D_u}{}'$ is acceptable by comparing it with the local trust threshold. If it is acceptable, D_u calculates a local trust secret key:

$$\mathrm{LTK}_{D_u-u}{}' = h_3\left(S \| r_{D_i} \| r_u \| v_{D_i-u} \oplus v_{u-D_i} \| \mathrm{LT}_{D_{u'}}\right). \tag{16}$$

Otherwise, it must choose another available device suitable to relay the message.

Figure 5 shows a summary of the local trust evaluation phase.

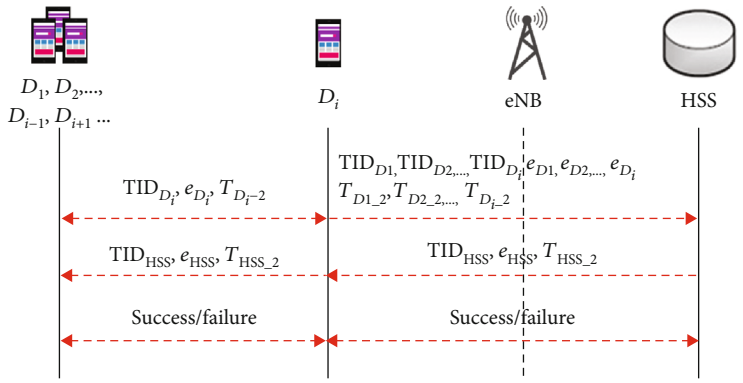

Figure 4: A message exchanged in the mutual authentication phase.

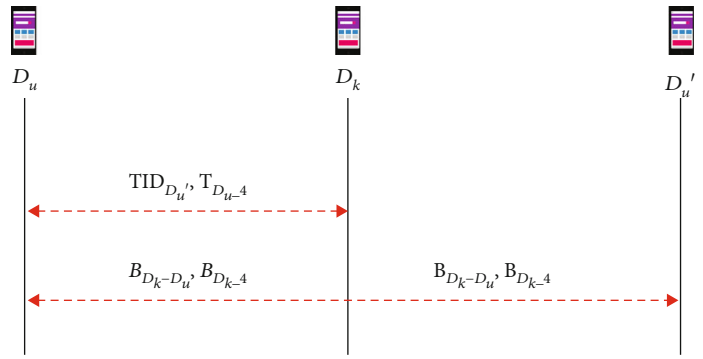

Figure 5: Messages exchanged when LT is used.

4.5. Encryption/Decryption. Finally, after the tests, the original device encrypts the data with session key $\mathrm{SKey}_{D_i-\mathrm{HSS}}$:

$$M = E\mathrm{SKey}_{D_i-\mathrm{HSS}}(\mathrm{data}). \quad (17)$$

The result (M) is encrypted with the proper key:

$$C = \mathrm{ELTK}_{D_u-D_u'}(M). \quad (18)$$

The message is sent to the most adequate device in the relay group with $T_{D_i{}^-5}$ and $ESKey_{D_u-D_u'}(LT_{D_u'})$. Then, $D_{u'}$ calculates the secret key:

$$\mathrm{LTK}_{D_u-u}{}' = h_3\left[(S \| r_{D_i} \| r_u \| v_{D_i-u} \oplus v_{u-D_i} \| \mathrm{LT}_{D_u'}\right], \quad (19)$$

decrypts the message, and obtains M:

$$M = \mathrm{DLTK}_{D_u-u}{}'(C). \quad (20)$$

$D_{u'}$ encrypts M with its own trust information through Equation (17) and sends the resulting message to the most adequate device in the relay group with a timestamp $T_{D_i{}^-5}$. The process is repeated until the device nearest the 3GPP infrastructure has been reached. This device sends M with a timestamp $T_{D_m{}^-5}$ to HSS.

HSS decrypts M using session key $\mathrm{SKey}_{D_i-\mathrm{HSS}}$ generated at the end of the mutual authentication phase, thus guaranteeing the legitimacy of the sender and the integrity of the data. It then forwards the patient's information to the health center server, which sends it to the doctor on a secure channel. Finally, the doctor receives the data and evaluates them.

Figure 6 shows a summary of the encryption/decryption phase, with a focus on encryption.

5. Security Analysis

This section reports on a security analysis of all D2D communication security devices and discusses the way they are approached by the proposed scheme.

5.1. Mutual Authentication. Devices perform mutual authentication to authenticate the other devices in the relay group. Each device calculates its Lagrange component (LC_{D_i}) and e_{D_i}, and they share e_{D_i} with the other devices in the relay group. Next, they calculate secret S' and $H(S')$ and compare the value obtained with the $H(S)$ received from HSS in the registration phase. If $H(S') = H(S)$, all devices involved are mutually authenticated. Otherwise, the operation is terminated.

After mutual authentication, the devices start the mutual authentication procedure with HSS. Each device generates MAC_{D_i} and sends it with the respective TID_{D_i} to HSS, which calculates MAC'_{D_i} and checks if $\mathrm{MAC}'_{D_i} = \mathrm{MAC}_{D_i}$. If the

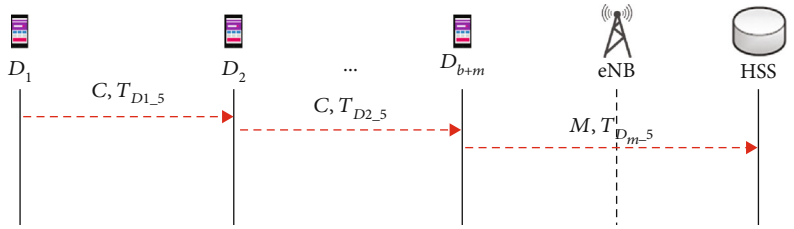

Figure 6: Encrypted data sent to HSS.

Table 2: Comparison of security objectives among protocols.

Security objectives	Zhang et al. [13]	Mustafa [18]	Zhou [16]	Proposed protocol
Mutual authentic	Yes	Yes	Yes	Yes
Trust evaluation	No	No	No	Yes
Confidentiality	No	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes
Privacy	No	Yes	Yes	Yes
Anonymity	Yes	No	Yes	Yes
Forward/backward secrecy	Yes	Yes	Yes	Yes
Nonrepudiation	Yes	Yes	Yes	Yes
Session key security	Yes	Yes	Yes	Yes
Resistance to replay attack	No	Yes	Yes	Yes
Resistance to insider attack	No	Yes	Yes	Yes
Resistance to DoS attack	No	Yes	No	Yes
Resistance to man-in-the-middle attack	No	Yes	Yes	Yes

Table 3: Cost of each unitary operation.

Notation	Devices (ms)	Network (ms)	Description
T_{hash}	0.201	0.067	Cost of a one-way hash operation
T_{mul}	1.84	0.612	Cost of a multiplication operation over an elliptical curve, represented as $*$
T_{add}	0.375	0.125	Cost of an addition operation over an elliptical curve
T_{mod}	0.372	0.124	Cost of a modular operation
T_{exp}	0.37	0.123	Cost of an exponential operation
T_{pair}	13.53	4.51	Cost of a bilinear pairing operation
T_{PK}	1.1	0.367	Cost of public key encryption
T_{AES}	0.483	0.161	Cost of an AES encryption/decryption operation

values are equal, HSS authenticates the devices and proceeds. Otherwise, the operation is terminated. Then, HSS generates its own Lagrange component LC_{HSS} and e_{HSS} and sends e_{HSS} to the group of relay devices. Each device recalculates its own Lagrange component LCnew_{D_i}, $e_{D_i-\text{new}}$, and a new secret S'' and compares S'' with secret S' previously calculated. If $S'' = S'$, HSS is authenticated by the devices. In the proposed scheme, an attacker finds a Lagrange component by solving the DLP problem, which has proven computationally infeasible.

5.2. Forward/Backward Secrecy of Session Key. Forward secrecy guarantees an intruder with access to an old key does not use it in the future for forging its authenticity. On the other hand, backward secrecy provides security against the use of newer keys for access to information originated in older sections. In the proposed scheme, forward and backward secrecies of the session key are guaranteed through the use of freshly generated random values r_{D_i}, timestamps T_{D_i}, and session keys $\text{SK}_{D_i-\text{HSS}}/\text{SK}_{D_i-Dk}$ in each authentication procedure.

5.3. Confidentiality. The scheme provides confidentiality of patients' data by generating a different session key $\text{SK}_{D_i-\text{HSS}}/\text{SK}_{D_i-Dk}$ in each session established between any device and HSS. All data exchanged over an insecure channel are encrypted with the respective session key, whose security is ensured.

5.4. Integrity. Data integrity is guaranteed by the encryption of the data sent by each patient through a securely established session key $\text{SK}_{D_i-\text{HSS}}/\text{SK}_{D_i-D_k}$ before it is sent over an insecure channel. When HSS decrypts the messages with the

Table 4: Comparison of computational costs.

Protocol	Devices (ms)	Server network (ms)	Total (ms)
Zhang et al. [13]	$(3n+3m+9)T_{mul}+2nT_{mod}+(4m+9)T_{hash}+2nT_{exp}+(n+4m+2)T_{add}+3T_{ECC}=7.38n+7.83m+22.42$	$(n+6)T_{mul}+2nT_{mod}+(n+7)T_{hash}+2nT_{exp}+(n+2)T_{add}+2T_{ECC}=1.3n+5.13$	$(4n+3m+15)T_{mul}+4nT_{exp}+(n+4m+16)T_{hash}+4nT_{exp}+(2n+4m+4)T_{add}+5T_{ECC}=8.68n+7.83m+27.53$
Zhou [16]	$(3n+3m+11)T_{mul}+(2n+2)T_{mod}+(4m+12)T_{hash}+2nT_{exp}+(n+4m+2)T_{add}+T_{ECC}=7.38n+7.83m+25.25$	$(n+10)T_{mul}+2nT_{mod}+(n+6)T_{hash}+2nT_{exp}+(n+5)T_{add}+T_{ECC}=1.3n+7.52$	$(4n+3m+21)T_{mul}+(4n+2)T_{mod}+(n+4m+18)T_{hash}+4nT_{exp}+(2n+4m+7)T_{add}+2T_{ECC}=8.68n+7.83m+32.77$
Proposed protocol	$nT_{mul}+3nT_{mod}+(n+2m)T_{hash}+nT_{exp}+3nT_{ECC}+(3m+1)T_{AES}=6.83n+1.85m+0.48$	$(n+1)T_{mul}+(2n+7)T_{mod}+3nT_{hash}+(2n+4)T_{exp}+nT_{ECC}+T_{AES}=1.67n+2.13$	$(n+1)T_{mul}+(5n+7)T_{mod}+(4n+3m)T_{hash}+(3n+4)T_{exp}+4nT_{ECC}+(4m+2)T_{AES}=8.5n+1.85m+2.61$

appropriate session key, it knows the information was generated by the genuine source and not modified on the way to the destination.

5.5. Anonymity. The anonymity of entities, devices, and HSS is safeguarded through the exchange of only temporary identities (TID_{D_i}, TID_{HSS}) over an insecure channel. Therefore, the permanent identities are not disclosed over an insecure channel. HSS knows the permanent identity of all devices; however, this information is obtained offline.

5.6. Nonrepudiation. Nonrepudiation certifies an entity cannot deny its actions. In the proposed scheme, it is guaranteed through the use of permanent (ID_{D_i}) and temporary identities (TID_{D_i},TID_{HSS}) and private and public keys.

5.7. Session Key Security. The security of the session key is ensured by confidential information Y_{D_i} and ID_{D_i} in its generation process, as shown in Equation (13).

5.8. Resistance to Impersonation Attack. Impersonation attack is avoided by different temporary identities in each session established. A TID is never used twice and HSS can recognize whether a certain TID has already been used.

5.9. Resistance to Replay Attack. A replay attack is avoided by freshly generated parameters, such as random values r_{D_i} and timestamps T_{D_i} in the mutual authentication phase, generation of session keys, and use of different TID_{D_i} and TID_{HSS} in each session.

5.10. Resistance to Denial of Service (DoS) Attack. The use of timestamps in each message exchanged over an insecure channel avoids denial of service (DoS) attacks. Each timestamp is synchronized with its respective entity's clock, which is also synchronized with the whole system.

5.11. Resistance to Man-in-the-Middle Attack. Session keys and local trust keys do not depend only on values exchanged on an insecure channel, but on secret values securely exchanged in the registration phase encrypted with devices' public key.

Such security objectives were not accomplished by [13, 16 and [18]. First, any of the compared protocol performs the trust evaluation of relay devices. Secondly, as shown by [14], the scheme proposed in [13] is vulnerable to an insider attack, which compromises its confidentiality and might also affect patients' privacy and the protocol's resistance to replay and man-in-the-middle attacks.

The schemes proposed by [13, 16] are vulnerable to DoS attacks, since they do not use verification values as nonces or timestamps prior to the execution of more complex calculations. The scheme designed by [18] does not protect the anonymity of devices because it does not mention the use of temporary or pseudoidentity instead of their permanent identities.

Table 2 shows a comparison among our scheme and those of [13, 18] and [16].

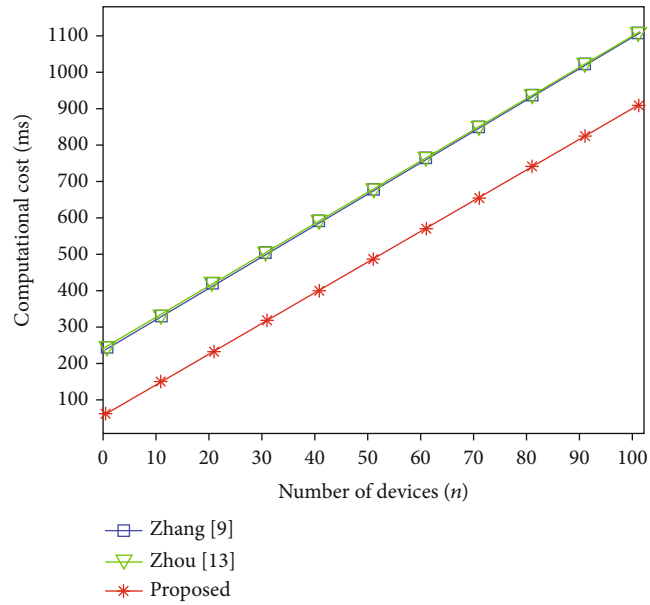

Figure 7: Comparison of computational costs.

6. Performance Analysis

This section reports on a performance analysis of our protocol regarding computational and energy costs in each authentication session executed, and a performance comparison among our scheme and those of [13, 16].

6.1. Computational Costs. The evaluation of computational costs is based on an estimate of the time necessary for the execution of unitary operations, considered part of the messages previously described in the different phases of the protocol. Table 3 shows such unitary operations (hash, multiplication, and addition over elliptical curve, exponentiation,…) with the corresponding computational costs (running times, measured in milliseconds) for both devices and the fixed part of the cellular wireless network.

The cost values are based on common and realistic values obtained by experimentation and adopted for performance comparisons of authentication protocols, as in Choi et al. 2014 [8] and Hsu et al. 2018 [21].

The computational platform was configured as follows:

(i) Intel Core Duo 1.86 GHz and 2 gigabyte RAM under an Ubuntu 11.10 operating system [8]

(ii) HTC One X smartphone with Android 4.1.1, 1.5 GHz Quad-core ARM Cortex-A9 CPU, 1 GB RAM [21]

The methodology adopted for the performance evaluation considers the cost of each unitary operation multiplied by the number of times each operation is executed, comprehending the several messages that include one or more of such unitary operations, as required for the different authentication protocols.

Table 4 shows a comparison of the computational costs (in milliseconds) among our protocol and those designed by [13, 16]. An environment with "n" devices

Table 5: Comparison of energy costs.

Protocol	Devices	Network	Total (mJ)
Zhang et al. [13]	$80.3n + 85.2m + 243.93$	$14.14n + 55.81$	$94.44n + 85.19m + 299.53$
Zhou [16]	$80.3n + 85.2m + 274.72$	$14.14n + 81.82$	$94.44n + 85.19m + 356.54$
Proposed protocol	$74.31n + 20.13m + 5.22$	$18.17n + 23.17$	$92.48\,n + 20.13m + 28.4$

registered in the 3GPP network and "m" devices involved in the relay of the messages sent from the source device and the HSS was considered. Only the devices involved in the relay of data performed the calculations of the trust evaluation phase.

The devices take $nT_{\text{mul}} + 3nT_{\text{ECC}}$ in the registration phase to calculate their partial public key and encrypt/decrypt secret values $v_{D_i\text{-}D_u}$. Then, they take $3nT_{\text{mod}} + nT_{\text{hash}} + nT_{\text{exp}}$ in the mutual authentication and key agreement phase to calculate their Lagrange component, secret S', and session key $SKey_{D_i\text{-}k}$. Next, $mT_{\text{hash}} + mT_{\text{AES}}$ is required for the encryption of local trust result $LT_{D_u}{}'$ and local trust secret key $LTK_{D_u\text{-}u}{}'$ is calculated. Finally, the devices expend $mT_{\text{hash}} + (2m+1)T_{\text{AES}}$ to encrypt the patients' information generating M and encrypting/decrypting M with local trust secret key $LTK_{D_u\text{-}u}{}'$. Consequently, the total computational cost for the devices is $nT_{\text{mul}} + 3nT_{\text{mod}} + (n+2m)T_{\text{hash}} + nT_{\text{exp}} + 3nT_{\text{ECC}} + (3m+1)T_{\text{AES}}$. According to Table 4, the computational cost for the devices is $6.827n + 1.851m + 0.483$ ms.

3GPP network takes $(n+1)T_{\text{mul}} + 2nT_{\text{hash}}$ to calculate temporary identities and partial private keys for each device and its master public key in the initialization phase. It takes $(2n+4)T_{\text{mod}} + (2n+3)T_{\text{exp}} + nT_{\text{ECC}}$ to generate tokens for each device and for itself in the registration phase. Next, it requires $3T_{\text{mod}} + nT_{\text{hash}} + T_{\text{exp}}$ to calculate the Lagrange component of HSS, secret S', and session keys $SKey_{D_i\text{-HSS}}$. Finally, it takes T_{AES} to decrypt message M and obtain the source patient's information. Therefore, the computational cost for the core network is $(n+1)T_{\text{mul}} + (2n+7)T_{\text{mod}} + 3nT_{\text{hash}} + (2n+4)T_{\text{exp}} + nT_{\text{ECC}} + T_{\text{AES}}$. According to the operation costs in Table 4, the computational cost for the network is $1.674n + 2.133$ ms.

The lines in Figure 7 show satisfactory results of our protocol regarding computational costs. A situation in which 25% of devices are involved in the relay of data was considered. The protocol clearly showed better costs than [13, 16]; [16] yielded slightly different results from [13].

Regarding the similarity involving the results of [13, 16] is an improvement of [13], and, consequently, most calculations are similar.

The main difference between [13, 16] is the correction of security vulnerabilities, made in [16] with the use of more variables in the authentication procedure. In terms of operations, [16] requires the calculation of only an extra elliptic curve cryptography-based (ECC-based) scalar multiplication on G_1 when compared to [13]; such operation requires small processing time.

Our scheme has shown excellent computational performance regarding all subjects addressed. The use of Shamir's secret sharing in the authentication phase reduces the computational resources consumption, since all devices and HSS are authenticated with a single calculation and comparison of secrets S' and S, respectively.

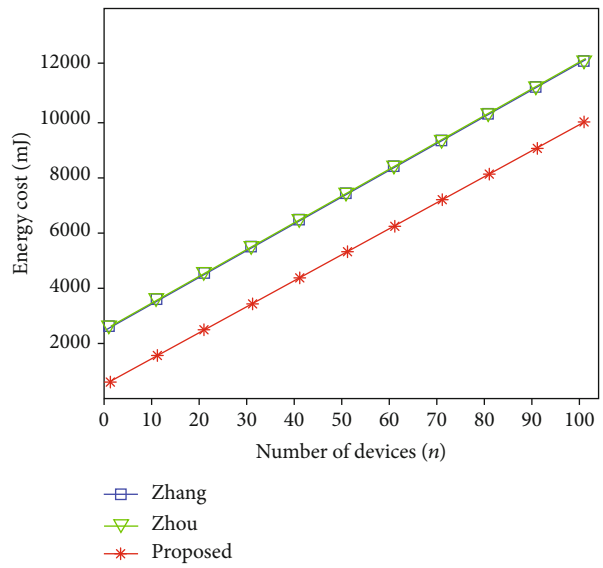

Figure 8: Comparison of energy costs.

6.2. Energy Costs. This section reports on an analysis of the energy cost of our protocol and a comparison with [13, 16]. The evaluation is based on the proposals of Kumar et al. [22] and He et al. [23], which consider 10.88 Watts the maximum CPU power of devices (W). The following operation was performed for the calculation of the energy overhead: ETotal = CCTotal × W, where CCTotal is the computational cost of each operation. Table 5 shows the results.

Figure 8 shows a comparison based on the energy costs provided in Table 5. The red line representing our protocol proves its lower energy consumption in comparison to the protocols of [13, 16]. As in the evaluation of computational costs, the energy costs of [13, 16] were similar.

The main differences between [13, 16] that lead to consequences in the energy costs are the same of the computational costs. Our scheme also shows higher energy efficiency due to reduced processing efforts and computational cost.

7. AVISPA Verification

The protocol was validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) [24], which simulates messages exchanged among entities involved in an

```
role
role_Dkj(Dkj:agent,Dij:agent,HSS:agent,TIDdij:text,TIDdkj:text,TIDhss:text,PKdkj:text,Vdkdj
:text,
      AUTHdk:text,Rdk:text,Edk:text,Tdkdu:text,M:text,Ydk:text,Tdkhss:text,C:text,
      Key_set_Dkj_HSS:(symmetric_key) set,Key_set_HSS_Dkj:(symmetric_key)
set,SND,RCV:channel(dy))
played_by Dkj
def=
        local
        State:nat,PKdij:text,Vdidj:text,Fkj:text,Wkj:text,Ddkj:text,Vhssdk:text,Rdi:text,Edi:text,
Tdkj:text,

Rhss:text,Thss:text,Ehss:text,LTdkdu:text,Tdij:text,Data:text,Tdku:text,SecureChannel:symmet
ric_key,
  Key_4:symmetric_key,Key_3:symmetric_key,Key_2:symmetric_key,Key_1:symmetric_key
        init
                State := 0

    1.  State=0 /\RCV(Tdij'.TIDdij.PKdij') =|>
                State':=1 /\Tdkj':=new() /\SND(Tdkj'.TIDdkj.PKdkj)
    2.  State=1 /\RCV({Vdidj'}_SecureChannel') =|>
                State':=2  /\  SND({Vdkdj}_SecureChannel')  /\  Key_1':=new()  /\
              Key_set_Dkj_HSS':=cons(Key_1',Key_set_Dkj_HSS) /\SND({Vdkdj}_Key_1')
    1.  State=2 /\ in(Key_2',Key_set_HSS_Dkj) /\
        RCV({{Wkj'.Ddkj'.Fkj'.Vhssdk'}_SecureChannel}_Key_2') =|>
                State':=3 /\Key_set_HSS_Dkj':=delete(Key_2',Key_set_HSS_Dkj)
    2.  State=3 /\RCV(Tdij.TIDdij.Edi'.Rdi') =|>
                State':=4  /\  SND(Tdkj.TIDdkj.Edk.Rdk)  /\  Key_3':=new()  /\
              Key_set_Dkj_HSS':=cons(Key_3',Key_set_Dkj_HSS)                    /\
              SND({Tdkj.TIDdkj.Edk.Rdk}_Key_3')
        14. State=4 /\in(Key_4',Key_set_HSS_Dkj) /\
RCV({Thss'.TIDhss.Ehss'.Rhss'}_Key_4') =|>

                State':=5 /\Key_set_HSS_Dkj':=delete(Key_4',Key_set_HSS_Dkj)
        15. State=5 /  \ RCV(Tdij.TIDdij) =|>
                State':=6 /\LTdkdu':=new() /\Tdku':=new() /\SND(Tdku'.LTdkdu') /\
SND(Tdku'.LTdkdu')
        18. State=6 /\  RCV(Tdij.{Data'}_SecureChannel) =|>
            State':=7 /\  secret(Data',sec_8,{}) / \ SND(Tdku.{Data'}_SecureChannel) /\
secret(C',sec_7,{}) /\   SND(Tdkhss.{C} _SecureChannel)

end role
```

Figure 9: Role of a device in HLPSL language for AVISPA software.

authentication scheme. AVISPA simulation is written in High-level Protocol Specification Language (HLPSL), which divides the message exchanged into roles representing each entity involved in the authentication procedure.

Figure 9 shows an example of the role of an ordinary D2D communication device. The objectives verified were ability of the protocol to perform D2D mutual authentication and key agreement and secrecy of session keys.

AVISPA has four backends to verify security. We used two of them, namely, On-the-Fly-Model-checker (OFMC) [25] and Constraint Logic-Based Attack Searcher (CL-AtSe) [26], which return "SAFE" if the protocol analyzed is considered safe, and "UNSAFE" if it has found an issue that might compromise security. According to Figures 10 and 11, the protocol was considered safe.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/hlpslGenFile.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00 s
  searchTime: 0.72 s
  visitedNodes: 63 nodes
  depth: 8 plies
```

Figure 10: OFMC analysis.

8. Conclusions and Future Work

Internet of Things (IoT) devices have been designed for several new applications and creation of a framework of benefits that improves services and people's life quality, assures safety and security, and reduces expenses [27]. Some of such applications include solutions for m-health, which enable patients to share information on their health to be monitored or receive fast aid in emergencies, thus improving the quality of care [28]. D2D communication is suitable for m-health IoT applications, since it provides direct communication among devices with no intermediation of infrastructures, such as the one available by 3GPP.

The traditional authentication and key agreement standardized by 3GPP is not suitable for D2D authentication

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/hlpslGenFile.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analyzed : 77 states
  Reachable : 21 states
  Translation: 1.50 seconds
  Computation: 0.00 seconds
```

Figure 11: CL-AtSe analysis.

and therefore cannot deal with the lack of access to the network infrastructure faced by some devices. New applications that exchange critical data (e.g., m-health applications) require novel AKA schemes to fulfill such a demand. A good alternative is the relay of data through close devices until the network infrastructure has been reached, as proposed by [12].

Our protocol has been designed to provide a new AKA scheme; it aims at fulfilling the security properties detailed by 3GPP specifications TS 23.303 [19] and TS 33.303 [11] and reducing resource consumption. Such a reduction has been achieved by the scheme adopted, as proposed by Harn [15], based on Shamir's secret sharing [17]. A trust evaluation indicated the close devices suitable for the relay of data; it was based on the scheme developed by [12], to guarantee the delivery of data from the source device to the health center

Several security properties and resistance to attacks, as addressed in Section 5.4, demonstrated the robustness of our protocol, which has proven safer than those of [13, 16]. The protocol designed by [13] showed confidentiality issues and, consequently, is not resistant to attacks (e.g., insider and man-in-the-middle). The scheme of [16] is not resistant to DoS attack, and the one developed by [18] shows anonymity problems, since it offers no protection to devices' real identities.

Our protocol has proven the safest, because it has fulfilled all security objectives required by [11, 19], as shown in Table 2, and achieved better performance, in comparison to [13, 16], which have similar costs due to their similarity. The validation made by AVISPA with the use of two of its backends also confirmed the safety of the protocol regarding message exchange of secret parameters. Therefore, no intruder can discover confidential and critical parameters and information.

Our protocol is part of a project that involves the development of software applications, which benefit from the junction of D2D communications and edge computing. Among the application areas considered are smart cities, e-health/m-health, and smart grid networks.

Future work will include the proposal of authentication and authorization protocols based on cyber-physical systems ([28, 29]), as well as the formal validation of our protocol by tools, such as ProVerif and Tamarin. The simulation of our protocol by NS-3 or OMNET++ tools has been considered for the evaluation of energy efficiency and influence of device mobility.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

On behalf of all authors, the corresponding author states there is no conflict of interest.

Acknowledgments

This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brasil (Finance Code 001) (a scholarship was awarded to Ana Paula G. Lopes). The authors acknowledge the University of Brasilia, especially the Post-Graduation Program in Electrical Engineering (PPGEE), for the research support.

References

[1] C. Huang, K. Yan, S. Wei, and D. H. Lee, "A privacy-preserving data sharing solution for mobile healthcare," in *2017 International Conference on Progress in Informatics and Computing (PIC)*, pp. 260–265, Nanjing, China, December 2017.

[2] M. Wang and Z. Yan, "A Survey on Security in D2D Communications," *Mobile Networks and Applications*, vol. 22, no. 2, pp. 195–208, 2017.

[3] N. Hassan, K.-L. A. Yau, and C. Wu, "Edge Computing in 5G: A Review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019.

[4] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D Communications Meet Mobile Edge Computing for Enhanced Computation Capacity in Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1750–1763, 2019.

[5] D. Wu, F. Wang, X. Cao, and J. Xu, "Joint communication and computation optimization for wireless powered mobile edge computing with D2D offloading," *Journal of Communications and Information Networks*, vol. 4, no. 4, pp. 72–86, 2019.

[6] S. Kekki, W. Featherstone, Y. Fang et al., *ETSI White Paper No. 28 MEC in 5G networks First edition*, European Telecommunications Standards Institute, 2018.

[7] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, "Security and Privacy in Device-to-Device (D2D) Communication: A Review," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1054–1079, 2017.

[8] D. Choi, H.-K. Choi, and S.-Y. Lee, "A group-based security protocol for machine-type communications in LTE-advanced," *Wireless Networks*, vol. 21, no. 2, pp. 405–419, 2015.

[9] J. Cao, M. Ma, and H. Li, "GBAAM: group-based access authentication for MTC in LTE networks," *Security and Communication Networks*, vol. 8, no. 17, pp. 3282–3299, 2015.

[10] Y. Sun, J. Cao, M. Ma, H. Li, B. Niu, and F. Li, "Privacy-Preserving Device Discovery and Authentication Scheme for D2D Communication in 3GPP 5G HetNet," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 425–431, Honolulu, HI, USA, Febuary 2019.

[11] 3GPP TS 33.303, "Proximity-based services (ProSe); security aspects," 2018, https://www.3gpp.org/ftp/Specs/archive/33_series/33.303/. Visited on November 2019.

[12] Z. Yan, H. Xie, P. Zhang, and B. B. Gupta, "Flexible data access control in D2D communications," *Future Generation Computer Systems*, vol. 82, pp. 738–751, 2018.

[13] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 662–675, 2017.

[14] C. Zhou, "Comments on "Light-Weight and Robust Security-Aware D2D-Assist Data Transmission Protocol for Mobile-Health Systems"," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1869-1870, 2018.

[15] L. Harn, "Group authentication," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1893–1898, 2013.

[16] C. Zhou, "An improved lightweight certificateless generalized signcryption scheme for mobile-health system," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, Article ID 155014771882446, 2019.

[17] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 2001.

[18] U. Mustafa and N. Philip, "Group-Based Key Exchange for Medical IoT Device-to-Device Communication (D2D) Combining Secret Sharing and Physical Layer Key Exchange," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, UK, January 2019.

[19] 3GPP TS 23.303, "Proximity-based services (ProSe); stage 2," Jun. 2018. https://www.3gpp.org/ftp/Specs/archive/23_series/23.303/. Visited on November 2019.

[20] 3GPP TS 36.843, "Technical specification group radio access network; study on LTE device to device proximity services; radio aspects," 2014, https://www.3gpp.org/ftp/Specs/archive/36_series/36.843/, Visited on November 2019.

[21] R. H. Hsu, J. Lee, T. Q. S. Quek, and J. C. Chen, "GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 449–464, 2018.

[22] A. Kumar and H. Om, "Handover authentication scheme for device-to-device outband communication in 5G-WLAN next generation heterogeneous networks," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7961–7977, 2018.

[23] D. He, S. Chan, and M. Guizani, "Handover authentication for mobile networks: security and efficiency aspects," *IEEE Network*, vol. 29, no. 3, pp. 96–103, 2015.

[24] The AVISPA Project, *European Union in the Future and Emerging Technologies (FET Open)*http://www.avispa-project.org/ Visited on November 2019.

[25] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: a symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, no. 3, pp. 181–208, 2005.

[26] M. Turuani, "The CL-Atse protocol analyser," in *Term Rewriting and Applications. RTA 2006. Lecture Notes in Computer Science, vol 4098*, F. Pfenning, Ed., pp. 277–286, Springer, Berlin, Heidelberg, 2006.

[27] C. Free, G. Phillips, L. Felix, L. Galli, V. Patel, and P. Edwards, "The effectiveness of M-health technologies for improving health and health services: a systematic review protocol," *BMC Research Notes*, vol. 3, no. 1, p. 250, 2010.

[28] A. Essa, T. Al-Shoura, A. Al Nabulsi, A. R. Al-Ali, and F. Aloul, "Cyber physical sensors system security: threats, vulnerabilities, and solutions," in *2nd International Conference on Smart Grid and Smart Cities (ICSGSC)*, Kuala Lumpur, Malaysia, August 2018.

[29] L. Vegh, "Cyber-physical systems security through multi-factor authentication and data analytics," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, Lyon, France, Febuary 2018.