Authentication Protocols for D2D Communications

Ana Paula Golembiouski Lopes

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

Authentication Protocols for D2D Communications

Ana Paula Golembiouski Lopes

Orientador: Paulo Roberto de Lira Gondim

Banca Examinadora

Prof. Paulo Roberto de Lira Gondim, UnB/ENE, Orientador ______________________________

Prof. Joel José Puga Coelho Rodrigues, UFPI, Membro Externo ______________________________

Prof. Paulo Henrique Portela de Carvalho, ENE/UnB, Membro Interno ______________________________

Prof. Marcelo Menezes de Carvalho, ENE/UnB, Suplente. ______________________________

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AUTHENTICATION PROTOCOLS FOR D2D COMMUNICATIONS

ANA PAULA GOLEMBIOUSKI LOPES

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

____

PAULO ROBERTO DE LIRA GONDIM, Dr., ENE/UNB

(ORIENTADOR)

____

PAULO HENRIQUE PORTELA DE CARVALHO, Dr., ENE/UNB

(EXAMINADOR INTERNO)

____

JOEL JOSÉ PUGA COELHO RODRIGUES, Dr., UFPI

(EXAMINADOR EXTERNO)

Brasília, 17 de dezembro de 2019.

FICHA CATALOGRÁFICA

LOPES, ANA PAULA GOLEMBIOUSKI

Authentication Protocols for D2D Communications. [Distrito Federal] 2019.

xvii, 155p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

1.Autenticação e Acordo de Chaves (AKA)

2. Comunicações Dispositivo-a-Dispositivo (D2D)

3. Internet das Coisas (IoT)

4. Segurança da Informação

I.ENE/FT/UnB

II. Mestre

REFERÊNCIA BIBLIOGRAFICA

LOPES, A. P. G. (2019). Authentication Protocols for D2D Communications. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEE.DM-737/19, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 155p.

CESSÃO DE DIREITOS

AUTORA: Ana Paula Golembiouski Lopes

TITULO: Authentication Protocols for D2D Communications.

GRAU: Mestre ANO: 2019

Ana Paula Golembiouski Lopes
UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia.
Departamento de Engenharia Elétrica.
70910-900 – Brasília – DF – Brasil.

Agradecimentos

Agradeço a Deus por mais esta conquista e por sempre ter me amparado em orações. Aos meus pais e a minha irmã, que me incentivam desde muito pequena a ser dedicada aos meus estudos e que mais uma vez foram o principal ponto de apoio para que eu pudesse alcançar meu objetivo acadêmico. Às minhas amigas Aline Santos e Luciana Brito, por todos os anos de amizade e por terem sido meus ombros amigos durante os anos de mestrado. Por fim, a todos os meus professores, em especial ao meu orientador Paulo Gondim, pelos ensinamentos que possibilitaram a execução deste trabalho.

Resumo

A comunicação Dispositivo-a-Dispositivo (D2D) é uma das tecnologias promissoras para ser usada na conexão de grandes quantidades de dispositivos, como previsto para a Internet das Coisas (IoT, do inglês *Internet of Things*), ao proporcionar a oportunidade de conexão direta entre dispositivos, sem a obrigatoriedade de emprego da infraestrutura de redes tradicionais.

A segurança é um item crucial para o sucesso da IoT e das comunicações D2D e pode ser proporcionada por protocolos de autenticação e acordo de chaves (AKA, do inglês *Authentication and Key Agreement*). Entretanto, os protocolos de autenticação utilizados nas redes tradicionais (como os protocolos EPS-AKA e EAP-AKA) não estão adaptados para D2D, e seu emprego em situação de grande aumento no número de dispositivos conectados imporia um elevado consumo de recursos, especialmente de banda e de processamento computacional. Adicionalmente, no início do trabalho foram identificados poucos protocolos dessa categoria, especificamente voltados para D2D.

Este trabalho apresenta o projeto e a avaliação de 3 (três) protocolos de autenticação e acordo de chaves para comunicações D2D, desenvolvidos para 3 (três) cenários:1) dispositivos integrantes de *Telecare Medical Information Systems (TMIS)* baseados em sistema de nuvem computacional; 2) grupos de dispositivos em cenário genérico de emprego de comunicações D2D, onde sejam esperadas grandes quantidades de dispositivos; 3) grupos de dispositivos em comunicações D2D em cenário *m-health*.

A metodologia para obtenção de novos protocolos seguros considerou, como passo inicial, uma revisão da literatura, buscando identificar protocolos que tenham sido empregados, de forma específica, em cada cenário considerado. Em seguida, foi definida uma arquitetura específica de cada cenário considerado, bem como propriedades de segurança a serem alcançadas e possíveis ataques contra os quais caberia oferecer proteção. Foram então criados novos protocolos de autenticação para os cenários e arquiteturas citados, considerando o emprego de comunicações D2D.

Em todos os três cenários, dentre as propriedades de segurança tidas como requisitos para o correto funcionamento da comunicação D2D, incluem-se a preservação da confidencialidade, a integridade e a disponibilidade do sistema; em termos de possíveis ataques, ataques tais como os dos tipos *man-in-the-middle*, repetição e personificação foram tratados, visando proteção pelo protocolo contra os mesmos.

Após a descrição de cada protocolo, esta dissertação apresenta comparações em relação a propriedades de segurança entre cada um dos protocolos propostos e alguns de seus respectivos trabalhos relacionados. Uma comparação envolvendo custos de computação, de comunicação e de energia é então realizada. Os resultados obtidos mostram bom desempenho e robustez em segurança para os três esquemas propostos. As propostas mostram-se adequadas para uso futuro, na autenticação de dispositivos IoT que utilizarem comunicação D2D, dentro dos cenários adotados e sob as condições em que foram avaliadas.

Uma validação semiformal dos protocolos é também apresentada. A ferramenta AVISPA é utilizada para verificar a robustez da segurança dos protocolos desenvolvidos.

Palavras-chave – Autenticação e Acordo de Chaves (AKA), Comunicação Dispositivo-a-Dispositivo (D2D), Internet das Coisas (IoT), segurança, mobile health (m-health).

Abstract

Device-to-Device (D2D) communication is one of the promising technologies to be used to connect the large quantity of devices, as forecasted for the Internet of Things (IoT), by providing to devices the opportunity of connecting each other without mandatory use of traditional networks infrastructure.

Security is a crucial item for the success of IoT and D2D communication and can be provided by robust authentication and key agreement protocols (AKA). However, the authentication protocols used for traditional networks (such as EPS-AKA and EAP-AKA) are not adapted for D2D and their use in the situation of large number of devices connected would impose high consume of resources, specially bandwidth and computational processing. Additionally, in the beginning of the work, it was identified a small quantity of protocols of the described category, specifically for D2D.

This work provides the project and evaluation of 3 (three) authentication protocols designed to meet the demand on Device-to-Device (D2D) communications authentication and key agreement protocols, developed for 3 (three) scenarios: 1) devices that are members of Telecare Medical Information Systems (TMIS) based on cloud system; 2) groups of devices in generic scenario for the use of D2D communications, which there are expected large quantities of devices; 3) groups of devices for D2D communication in m-health scenario.

The methodology for obtaining of new secure protocols considered, as initial step, a literature review, searching for protocols that might be specifically used in each of the scenarios considered. Next, a specific architecture for each scenario considered was developed, as well as security properties to be accomplished and possible attacks that might be suitable for the protocol to have protection. Therefore, authentication protocols were created for the scenarios and architecture cited, considering the use of D2D.

In all three cases, among the security objectives required for the proper functioning of D2D communication, there are included the preservation of confidentiality, integrity, and availability of the system; in terms of attacks, such as man-in-the-middle, replay and impersonation were treated, aiming the protection of the protocols against the cited attacks.

After the description of each protocol, this dissertation presents comparisons regarding security properties among each of the proposed protocols and some of their respective related works. A comparison involving computational, communication and energy costs is executed. The results obtained show good performance and robust security to the three proposed schemes. The proposals show up suitable future use, in the authentication of IoT devices using D2D communication, in the scenarios adopted and under the conditions evaluated.

A semi-formal validation of the protocols is also presented. The tool AVISPA is used to verify the security robustness of the protocols developed.

Keywords – Authentication and Key Agreement (AKA), Device-to-Device (D2D) communication, Internet of Things (IoT), security, mobile health (m-health).

Table of Contents

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
5G	5th generation of mobile networks
AKA	Authentication and Key Agreement
AuC	Authentication Center
AVISPA	Automated Validation of Internet Security Protocols and Applications
BAN	Body Area Network
BF-IBE	Boneh-Franklin Identity-based Encryption
BS	Backward Secrecy
CL-AtSe	Constraint Logic-based Attack Searcher
CLGSC	Certificateless Signcryption Scheme
CN-GD2C	Group Anonymity for D2D Communication with Core Network Assistance
CP-ABE	Ciphertext-Policy Attribute-based Encryption
CTN	Communicating Things Network
D2D	Device-to-Device
DHKE	Diffie Hellman Key Exchange
DoS	Denial of Service
ECDLP	Elliptic Curves Discrete Logarithm Problem
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
eNB	Evolved NodeB
eNodeB	Evolved NodeB
FS	Forward Secrecy
GRAAD	Group Anonymous and Accountable D2D Communication in Mobile Networks.
GW	Gateway

HLPSL	High Level Protocol Specification Language
HMAC	Hash-based Message Authentication Code
HSS	Home Subscribe Server
IBE	Identity Based Encryption
IBS	Identity Based Signature
IMSI	International Mobile Subscriber Identity
IND-CCA	Indistinguishability Under Adaptively Chosen Ciphertext Attack
IoT	Internet of Things
LAI	Location Area Identity
M2M	Machine-to-Machine
MAC	Message Authentication Code
MitM	Man-in-the-Middle
OFMC	On-the-Fly Model Checker
PAS	ProSE App Server User
PFS	ProSE Function Server
PPAKA	Privacy-Preserving Authentication and Key Agreement
ProSe	3GPP Proximity Services
SeDS	Secure Data Sharing Strategy
TMIS	Telecare Medical Information System
UE	User Equipment

Chapter 1

INTRODUCTION

1.1.Initial Considerations

The Device-to-Device communications (D2D) is an access technology that provides to devices the opportunity of direct connection among each other, without the necessity of traditional networks assistance. We are currently experiencing the emergence of the Internet of Things (IoT) that forecasts to increase the number of connected devices to the order of billions in the next few years, as highlighted by Gartner [1]. D2D communication is expected to be key part of IoT, because the direct connection among devices might be crucial for the success of IoT communication.

Those devices have several characteristics in common and, consequently, requirements in common. Among those characteristic and requirements is the fact that they are mostly resource constrained devices. Consequently, the technology developed to operate with them must consider that they have limited amounts of energy and reduced computational capacity, for example.

Some security challenges emerge because the traditional networks are not suitable to IoT and D2D communication scenario and there is a lack of suitable mutual authentication schemes available. The consequence is the vulnerability to several attacks and unauthorized access that results in confidentiality, integrity and privacy problems. Security is crucial for the success of IoT and, consequently, to D2D communication.

The authentication and key agreement (AKA) of devices is one of the ways that permit devices and network to verify the authenticity of each other prior to sending valuable data to each other, helping in the avoidance of the described security vulnerabilities. The AKA standardized protocols available for traditional networks still do not comprises the scenario of IoT and D2D communication and must be adapted or reformulated to attend the new security requirements.

There are some researches engaged in developing security surveys and authentication protocols suitable for IoT and D2D. Considering several situations that might include electronic health (e-health), mobile health (m-health), vehicular communication, sensors networks and many others. All of the researches believing that efficient authentication protocols can provide the attributes necessary to accomplish the security needs of these new and diverse scenarios.

1.2.Motivation

The motivation for the research is the lack of authentication protocols designed and adapted for D2D communication. The traditional schemes that are proposed by 3GPP (as EPS-AKA [2]) are not adapted to support the new requirements of IoT and new technologies as D2D. The current standard EPS-AKA designed for 3GPP LTE does not comprises a scenario with large amounts of devices willing to be connected.

The 3GPP already have some standardization material that can be used to guide the development of new protocols for D2D communication. It is the case of the technical specifications TS 23.303 and TS 36.843, which regards the Proximity Services (ProSe) in discovery of nearby devices using direct radio signals and of the TS 33.303 that comprises ProSe security requirements.

The Internet of Things (IoT) forecasts several new applications in diverse scenarios, including the e-health/m-health systems that aims at providing health services through information and communication technologies. The integration with IoT can include the monitoring of patients' health, made with sensors coupled to their body and connected by Body Area Network (BAN). It may also include the diagnosis and remote provisioning of health services to patients over public channels. D2D communication is a promising technology that might connect those devices and provide the proper message exchange for those systems.

The security of the mentioned devices must be assured to guarantee the success of IoT, thus governments and private institutions must address the discussion of cyber security of IoT. Brazilian`s government, for example, has a national plan of IoT established in a decree [3], which considers security and privacy as themes to be treated as an important part of the plan. Other themes include international insertion, regulation, professional education and infrastructure of connectivity and interoperability. Moreover, it must be reinforced the need for the adoption of international standards for information security by private and governmental initiatives and the encouragement of cooperation and interaction of private initiative, academy and civilian society to promote the awareness and funding to cyber security importance.

A large number of devices has been emerging and the extra number of control messages of the authentication and key agreement (AKA) might worsen problems of congestion already present in current communication channels. Therefore, a good solution is to group devices to be authenticated towards simplifying the process and reducing the consumption of resources.

The trust among devices is an issue that occur if D2D communication is used to perform relay among devices without direct access to network infrastructure. Not all devices are trustworthy, which can occasion in loss of data and security problems. Therefore, is it necessary to assure trust for D2D applications.

Consequently, it is necessary to develop new authentication protocols that can deal with the modifications in architecture, security and performance required for IoT and its promising access technologies as D2D communication. New schemes must be created based on the combination of new models and models already proved to be efficient for traditional communication systems.

1.3.Objectives

The general objective of this work is to propose new authentication protocols for D2D communication, considering different scenarios as m-health, e-health and situations where devices can be organized in groups to better perform their authentication. Fulfilling the D2D security requirements and having good performance when compared to other protocols published in the literature.

The specific objectives include:

1. Generation of three new authentication protocols for D2D communication, which might be used in different scenarios;

2. The application of security concepts to accomplish objectives as confidentiality, integrity, privacy and protection to several attacks as man-in-the-middle, impersonation and replay;

3. The evaluation and comparison of D2D and non-D2D protocols available in the literature, regarding general characteristics, security and computational, communication and energy costs;

4. The validation of the proposed protocols using AVISPA [5], an application that provides semiformal verification of authentication protocols.

1.4.Methodology

A methodology divided into phases was adopted in this work, as follows:

- Phase 1: bibliographic review of relevant themes for the work in development, which comprises the study of the D2D security requirements, a study of the existent authentication protocols for D2D communications;
- Phase 2: a study of a TMIS scenario, and some authentication protocols and development of a D2D TMIS-based authentication protocol;
- Phase 3: a study of the existent group authentication protocols for D2D communication and the development of a new protocol for the mentioned scenario;
- Phase 4: a study of trust among devices and the development of a D2D group authentication protocol that consider the trust problem among the devices involved;
- Phase 5: comparison of security and performance of the three proposals and their respective related work, considering bandwidth consumption for communication cost, the processing time of each operation performed, and the energy used during the authentication procedure;
- Phase 6: formal validation of the proposed protocols using AVISPA, a tool developed for the verification of authentication protocols;
- Phase 7: development of papers describing and evaluating the proposed protocols, in a comparative manner with other proposals;
- Phase 8: dissertation text writing and defense.

1.5.Contributions

The main contributions of the work are:

1. The discussion of authentication protocols for D2D communications;
2. The proposal of an authentication protocol for TMIS with cloud-based networks that is adapted to D2D communication of the patients' devices;

3. Proposal of a group authentication protocol for D2D communication, which might be used in situations with large quantities of devices, such as m-health and agriculture;

4. Proposal of a group authentication protocol for D2D communication in m-health scenario, which considers the necessity of trust among devices;

5. The evaluation of the proposed schemes regarding security properties and computational, communication and energy costs;

6. The semi-formal validation of the proposed schemes.

1.6. Publications

During the research work, a scientific paper has been published and three papers have been submitted to international journals.

Publication on a scientific event (as appears in the Appendix 1):

Ana Paula G. Lopes, Paulo R. L. Gondim, Jaime Lloret: "Mutual Authentication Protocol for Cloud-based E-health Systems", Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg'18), Natal-RN, 2018.

Publications submitted to International Journals:

Ana Paula G. Lopes, Paulo R. L. Gondim, "Mutual Authentication Protocol for D2D Communications in a Cloud-Based E-Health System" – Appendix 2;

Ana Paula G. Lopes, Paulo R. L. Gondim, "A Lightweight Authentication Scheme for D2D Communication in M-Health with Trust Evaluation " – Appendix 2;

Ana Paula G. Lopes, Paulo R. L. Gondim, "Group Authentication Protocol Based on Aggregated Signatures for D2D Communication" - submitted to Computer Networks Journal (manuscript nr. COMNET_2019_1135) – Appendix 3.

1.7. Organization

The remainder of this dissertation is organized as follows: Chapter 2 presents some relevant concepts regarding authentication and security considered during the development of the three proposed protocols.

Chapter 3 presents a new cloud-based mutual authentication and key agreement protocol for e-health/TMIS systems adapted for D2D communication focused on reduction of computational and communication resources consumption, if compared with other protocols proposed in the literature. It considers both situations: devices with direct access to the 3GPP network and devices that need to perform D2D relay to reach the network.

In chapter 4, it is proposed a new authentication and key agreement protocol for groups of D2D devices that are assisted by the 3GPP infrastructure. It uses aggregated signatures to authenticate devices among each other and with the network as a group. It is provided a security and a performance analysis and a comparison among other D2D authentication protocols and the semi-automated formal validation of the proposed protocol.

In chapter 5, it is presented a mutual authentication and key agreement scheme for D2D devices in m-health for permitting patients to securely send their medical information to a health center and doctors. It is designed forecasting the relay of data in cases where devices are outside the 3GPP coverage area or inside of the coverage area but without access to the network, considering the necessity of computational trust for this described scenario.

Finally, in chapter 6 the conclusions of the dissertation are presented with some future work.

1.8 References

[1] Hung, M., "Leading the iot, gartner insights on how to lead in a connected world", *Gartner Research*, pp.1-29, 2017.

[2] 3GPP TS 33.401, 3GPP System Architecture Evolution (SAE). Security Architecture, V8.2.1, https://www.3gpp.org/ftp/Specs/archive/23_series/23.303/ , Visited on June 2019.

[3] BRASIL. Decreto nº 9854, de 25 de junho de 2019. Institui o Plano Nacional de Internet das Coisas e dispõe sobre a Câmara de Gestão e Acompanhamento do Desenvolvimento de Sistema de Comunicação Máquina a Máquina e Internet das Coisas.

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

Chapter 2

THEORETICAL BACKGROUND

Abstract. *Some important concepts to understand the schemes proposed in this work are presented in the theoretical background chapter. Concepts as authentication, encryption, security objectives and attacks, D2D communication, device discovery and trust are treated. In addition, there is a brief explanation of the verification tool used to validate the proposed protocols.*

2.1.Preliminary Definitions

System's security has as foundation three main objectives. They are confidentiality, integrity and availability and together are referred as the CIA triad. Stallings [1].

Confidentiality: Is the guarantee that information is only accessible and available to authorized entities. Information must be controlled and protected from attackers and eavesdroppers to avoid the leakage of secret data and data manipulation.

Integrity: Is the guarantee that information has not been manipulated, modified or destructed by unauthorized entities. Ensures that information source is authentic and supports nonrepudiation of the origin of data.

Availability: Is the guarantee that the system is operating properly to authorized and authentic entities whenever they need to access its components. Attackers might compromise availability by infection with malicious code or exploring computational vulnerabilities.

Some other relevant concepts are:

Authentication: The authentication of entities provides the chance of guaranteeing their authenticity among each other's, using parameters as identities and information that profs their unicity. It is executed prior to the occurrence of data exchange to guarantee CIA triad.

Authorization: It to grant access to determined information only to entities or individuals that have proven their authenticity through an authentication process.

Nonrepudiation: It is the guarantee that an entity cannot deny the origin of determined message or information. Therefore, it is part of the procedure of assuring the authenticity of information.

Privacy: It is the assurance that an entity has its information secure from unauthorized individuals.

Anonymity: Is the guarantee that individuals involved in a system have not their real identities disclosed. Avoiding the chance of being impersonated by bad intentioned individuals.

Trust: It is the assurance that an entity or individual can fulfill the commitments made regarding security, delivery of messages, computational capacity and many others.

Backward and Forward Secrecy (BS/FS): It is the assurance that information is secure in previous and subsequent sessions, by the utilization of secret keys at each authentication session. Even if the current key is disclosed, information exchanged in previous or future sessions cannot be accessed, because each key has its validity expired by the end of each session executed.

2.2.Security Attacks

There are some security attacks that are relevant for the protocols proposed in this work, because they are the most relevant regarding D2D communication. They are described by Stallings [1] as follows:

Replay Attack: The obtention of secret parameters by an intruder that eavesdrops the communication channel, which are used in the subsequent process executions to forge authenticity to the entities involved. It is imperceptible to the victim. It can be solutioned with the use of freshly generated parameters at each session and expiration of old parameters already used by the entities involved in the process.

Impersonation Attack: It occur when an intruder succeeds in obtaining enough information to pretend to be an authentic entity. Deceiving the other entities involved in the session execution to send to the intruder the messages destined to the genuine device. It can be avoided by the use of temporary or pseudo identities, that are valid only for a determined session. The permanent identities of the entities are never exposed over insecure channel.

Man-in-the-Middle (MitM) Attack: Performed by an intruder that can eavesdrop the communication channel and access all the information that is passing through it. Then, it joins the channel as a third entity, without being noticed. In a scenario with two entities, named A and B, the intruder, C, tricks A by intercepting the messages sent by A to B and making A to believe that it is B. The reverse also is executed: C tricks B by intercepting the messages sent from B to A and makes B believe that it is A. The intruder forwards the messages from entity A to entity B, which makes this attack hard to be detected. It can be avoided by the use of parameters shared offline and not transmitted in plaintext through insecure channel. Therefore, the attacker cannot forge valid parameters and impersonate the genuine entities.

Denial of Service Attack (DoS): The DoS attack is executed by attackers decided to make the services unavailable to authentic entities. It can be performed by overloading the network, a server or some entity with a large quantity of messages, which demands much time to be processed and interrupt the entire service. Entities with limited resources can be affected by just one attacker, while more complex system requires a group of attackers to occasion the unavailability of the system. It can be avoided by the inclusion of simple verification parameters to be verified before verification that require more complex calculations are performed. In this way, if an invalid timestamp or nonce is detected, the procedure is interrupted before the execution of the complex authentication calculations and the DoS attack is avoided.

2.3. Some other relevant concepts

Hash-based Message Authentication Code (HMAC) – The HMAC is a Message Authentication Code (MAC) generated through a one-way hash function. MAC values are a bloc of bits used in the authentication of entities.

Identity Based Signcryption (IBS) – It is a scheme that combines encryption and signatures (signcryption) and uses identities (or an arbitrary string) to produce system parameters and keys. Then, a plaintext can be signcrypted to obtain a ciphertext [2]. It is used in secure data exchange.

Diffie-Hellman Key Exchange (DHKE) - It has the purpose of enabling two or more users two securely exchange a key for symmetric encryption [1]. Its security depends on the attacker ability of solving the discrete logarithm problem.

2.4. Device-to-Device (D2D) Communication

Device-to-device (D2D) communication is a technology that enables the direct communication among devices, without the intervention of traditional network infrastructure such as 3GPP’s. This technology was not well developed in the past cellular network, but it is supposed to be a vital part of 3GPP 5G networks [3]. The current cellular network infrastructure is not adapted for D2D communication and the studies are just beginning. Therefore, D2D communication must be subject of many studies in order to fulfill the expectations for 5G networks.

According to Shen [4] “5G cellular networks are envisioned to attain 1,000 times higher mobile data volume per unit area, 10-100 times higher number of connecting devices and user data rate, 10 times longer battery life, and five times reduced latency.” , which might be accomplished by the aggregation of technologies such as spatial modulation, millimeter waves and massive MIMO [4].

Some advantages of devices’ direct connection are the offload of data, enlargement of coverage area, improvement of communication capability and reduction of communication delay and power consumption [4].

Gandotra and Jha [3] categorized D2D communication in four types:

1. **Device relaying with controlled link establishment from the operator**

Devices that are located outside the coverage area or with poor connection can communicate with traditional network infrastructure through direct connection with other D2D devices that can relay their information. The communication establishment and control is made by the base station (BS).

2. **Direct communication between devices with controlled link establishment by the operator.**

Devices can communicate directly with each other’s. The communication establishment and control is made by the base station (BS).

3. **Device relaying with controlled link establishment from the device.**

Similarly to item 1, devices that are located outside the coverage area or with poor connection can communicate with traditional network infrastructure through direct connection with other D2D devices that can relay their information. However, the establishment and control are made entirely by the devices involved in the communication.

4. **Direct communication between devices (Direct D2D) with controlled link establishment by the device.**

Similarly to item 2, Devices can communicate directly with each other's. However, the establishment and control are made entirely by the devices involved in the communication.

2.5. Elliptic Curves Diffie-Hellman (ECDH)

The Elliptic Curve Diffie-Helman protocol (ECDH) is described by Stallings [1] as a secret shared among two or more entities that is based on their information. It can be used as a shared key of the respective entities.

It uses elliptic curves cryptography associated to the Diffie-Hellman problem. Therefore, the security of the scheme is founded in the difficulty of resolving the discrete logarithm problem.

Here we have an example of ECDH key exchange among two entities, Alice and Bob:

- Step 1: Some system parameters are set, such as a large finite prime number *p*, an elliptic curve E over a large finite field $F_{p.}$ and a point P on that curve, which is a public value.
- Step 2: Alice and Bob chose a random number, Ra to Alice and Rb to Bob, and execute a multiplication over the elliptic curve RaP and RbP.
- Step 3: Alice sends to Bob RaP and Bob sends to Alice RbP.
- Step 4: Both calculate RaRbP and set it as the secret shared among them. Now they can use RaRbP as an encryption key to be used in data exchange.

In the case described above, the security of the system relies on the difficulty of an intruder to obtain Ra or Rb if it knows RaP, RbP and P. The discrete logarithm problem proves that it is computationally infeasible to recover these values. However, the original ECDH is vulnerable to Man-in-the-Middle attack, because an intruder can infiltrate the channel and intercept the messages from the entities and impersonate them. Figure 2.1 illustrates the message exchange.

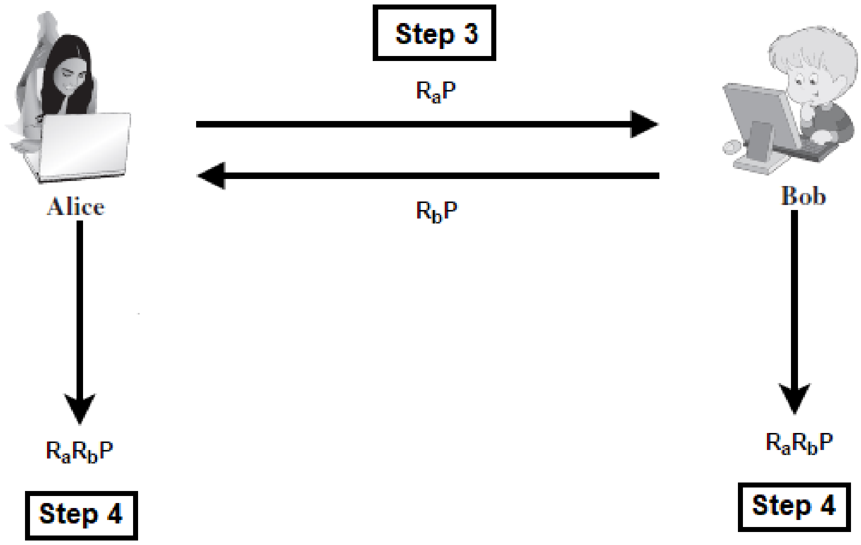


Figure 2.1 – ECDH example.

The protocols proposed in this work uses a modified ECDH, which uses some other secret parameters that are not exchanged over insecure channel and consequently, not vulnerable to attacks.

2.6. Bilinear Pairing

The bilinear pairing operation is used in this work in chapter 4, for group authentication. It provides the verification of entities among each other based on manipulation of critical parameters.

It was described by Menezes [5] as follows:

- Step 1: A prime number *p*, G_1 an additive group and G_T a multiplicative group of order *p* are generated.
- Step 2: A bilinear pairing on (G_1, G_T) is generated considering the map:

$$\hat{e}: G_1 \times G_1 \rightarrow G_T \tag{2.1}$$

The bilinear pairing satisfies the following conditions and properties

1. **Bilinearity:** For all R, S, T ∈ G1, $\hat{e}(R+S,T) = \hat{e}(R, T)\hat{e}(S, T)$ and $\hat{e}(R,S+T) = \hat{e}(R,S)\hat{e}(R,T)$.
2. **Non-degeneracy:** $\hat{e}(P,P) \neq 1$.
3. **Computability:** ê can be efficiently computed.”

4. $\hat{e}(S,\infty) = 1$ and $\hat{e}(\infty, S) = 1$
5. $\hat{e}(S,-T) = \hat{e}(-S, T) = \hat{e}(S, T)^{-1}$
6. $\hat{e}(aS, bT) = \hat{e}(S, T)^{ab}$ for all a, b ∈ Z
7. $\hat{e}(S, T) = \hat{e}(T, S)$
8. If $\hat{e}(S,R) = 1$ for all R ∈ G1, then S = ∞

The bilinear pairing operation can be used combined to ECDH and Aggregated Signatures schemes. In the DHKE example, Alice and Bob could exchange RaP and RbP and validate each other using bilinear pairing:

$$\hat{e}(S,T)^{RaRbP} \tag{2.2}$$

2.7.Shamir Secret

Adi Shamir proposed a scheme named Shamir's secret [6] that permit entities to obtain group authentication and to authenticate each other. Each entity sends to the other entities in the group its own share of the secret. In the *(k,n) threshold scheme* [6], a secret *D* is divided into *n* pieces D1, D2, …, Dn, and only with at least *k* pieces the secret *D* can be rebuild. In addition, the secret only can be restored if the pieces are legit.

The advantage of using Shamir's Secret in authentication protocols is that it is fast. Just one verification is necessary to authenticate the whole group of devices. The devices only are authenticated if all devices have proven to have a legit share of the secret. Consequently, a disadvantage is the impossibility of discovering which device is the intruder. However, Shamir’s Secret is used in many areas nowadays, such image compression, cryptography algorithms and authentication protocols.

2.8. Aggregated Signatures

Boneh et al. [7] says that “An aggregated signature scheme is a digital signature that supports aggregation.” Considering a scenario with n signatures from n distinct users, all signatures can be aggregated into a single short signature, which is enough to prove to a verifier that the n users signed the n original message.

Aggregated signatures can be used in authentication protocols to provide fast authentication, because all devices can be authenticated at one unique authentication procedure. Therefore, it is suitable to group authentication, which is in the scope of this work.

In the work of [7], it is also described the bilinear aggregation, which uses bilinear map on two groups G1 and G2 and a bilinear map *e* on G1 ×G2 → GT. Each group has its own generators g_1 and g_2.

First, each entity calculates its public key $v_i \leftarrow g_1^x$, where x is a random number chosen by the entity. Then, each device calculate the hash of a message M and its signature σ_i , as follows:

$$h_i = H(M) \tag{2.3}$$

$$\sigma_i = h_i^x \tag{2.4}$$

The signatures are aggregated as follows:

$$\sigma = \prod_{i=1}^{k} \sigma_i \tag{2.5}$$

Then, the verification of the aggregated signature is made using bilinear pairing, as follows:

$$e(g_i, \sigma) = \prod_{i=1}^{k} e(v_i, h_i) \tag{2.6}$$

The verification is accepted if the equality above holds.

2.9. AVISPA Tools

Automated Validation of Internet Security Protocols and Applications (AVISPA) is a verification tool that provides the validation of security-sensitive protocols, which was created by the AVISPA project [8]. The objective is to formalize protocols, security goals and threat models by automatically validating them and detecting errors.

The validation if performed with the message exchange writing in High-Level Protocol Language (HLPSL), which are organized in a sender/receiver style [8]. It supports asymmetric and symmetric encryption, cryptographic hash functions, nonatomic keys and exponentiation [9]. The code is divided into roles performed by the agents (or entities) involved in the authentication procedure.

AVISPA has four back-ends and two are used in the validation of the protocols proposed in this work, the On-the-fly-Model-Checker (OFMC) and the Constraint-Logic-based Attack Searcher (CL-AtSe). The back-ends return “SAFE” if the verification judges the protocol message exchange safe and “UNSAFE” if any security properties were violated, and the protocol is vulnerable attacks.

The OFMC back-end generates a binary tree with the decisions that can be executed by the protocol and return the following results, as described in [10]: ParseTime, the time took to analyze the system; SearchTime, the time took for the system to search for attacks; VisitedNodes, the number of nodes visited in the verification; Depth, the depth reached in the visit.

The CL-AtSe back-end each step is modeled by constraints on the adversary ‘s knowledge and the analysis are designed for a bounded number of protocol steps (loops). It translates the HLPSL of the protocol into constraints that can be used to find attacks [10]. It returns the following results, as described in [10]: Analyzed, number of loops analyzed; Reachable, number of steps reached by the analysis; Translation, the time took to translate the HLPSL code; Computation, time took in the analysis of the protocol.

2.10 References

[1] Stallings, William. “Cryptography and network security: principles and practice”, v.6, 2014.

[2] Malone-Lee, John. "Identity-Based Signcryption", *IACR Cryptology ePrint Archive,* v. 2002. n.98, 2002.

[3] Gandotra, P., and Jha, R. K., “Device-to-device communication in cellular networks: A survey”, *Journal of Network and Computer Applications,* n.*71,* pp.99-117, 2016.

[4] Shen, X., “Device-to-device communication in 5G cellular networks.” *IEEE Network,* v.29, n.2, pp.2-3, 2015.

[5] Menezes, Alfred. “An introduction to pairing-based cryptography”, *Recent trends in cryptography,* v.477, pp.47-65, 2005.

[6] Shamir, A. “How to share a secret”, *Communications,* pp. 612-613, 1979.

[7] Boneh, D., Gentry, C., Lynn, B., and Shacham, H. “Aggregate and verifiably encrypted signatures from bilinear maps.” *International Conference on the Theory and Applications of Cryptographic Techniques,* pp. 416-432, 2003.

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

[9] Basin, D., Moedersheim, S., and Vigano, L., “OFMC: A symbolic model checker for security protocols”, *International Journal of Information Security,* v.4, n.3, p.181-208, 2005.

[10]AVISPA Team, “AVISPA v1.0 User Manual”, Project funded by the European Community under the Information Society Technologies Programme, v. 1, 2006, from http://www.avispa-project.org , Visited on November 2019.

Chapter 3

MUTUAL AUTHENTICATION PROTOCOL FOR D2D COMMUNICATIONS IN A CLOUD-BASED E-HEALTH SYSTEM

Abstract. *The development of the Internet of Things has predicted several new applications, of which some will be incorporated into e-health systems, and some technologies, such as cloud computing and device-to-device communication (D2D), are promising to be used as a support to resource-constrained devices employed in m-health and Telecare Medicine Information Systems (TMIS) for the avoidance of performance problems and lack of spectrum in a scenario with billions of devices predicted for establishment of IoT, performance and security problems, among other issues, must be avoided. Security is fundamental for the achievement of optimal performance, regarding sensibility of e-health shared data and, especially, anonymity of patients and other entities, and scarcity of bandwidth in wireless networks must also be considered. This research proposes a new mutual authentication protocol for m-health systems, which supports D2D communication, ensuring security and surpassing the performance and security of other authentication procedures reported in the literature.*

Keywords: *authentication, device-to-device, m-health, security, IoT*

3.1. Introduction

Among the several applications for the development of Internet of Things (IoT), e-health/m-health aims at providing health services through information and communication technologies. Such applications include, for example, monitoring by sensors coupled to the body of patients and connected by Body Area Network (BAN), diagnosis and remote provisioning of health services to patients over public channels.

The assistance of cloud servers is an alternative for supplying the large demands of storage and processing generated by multiple medical service providers and increasing operational efficiency. According to Mohit et al. [1], in Telecare Medical Information Systems (TMIS), doctors and patients would work together through the cloud server. Patients send a report containing the sensors' measures to the cloud server and a doctor collects the data, provides a diagnosis and finally sends a diagnosis report to the cloud server. Both data exchanges are performed through public channels.

Additionally, the use of cloud servers as auxiliaries to the storage and processing in e-health/m-health/TMIS requires special attention, due to the high sensitivity of the information exchanged among the cloud server and the entities involved. Information of the sensor measurements report and patient diagnosis can be crucial for saving lives and must not be accessed or modified by possible attackers.

A good example is the anonymity of entities, since the user of those systems may not be interested in having their identity disclosed. In certain cases, the disclosure of a patient's identity can leave it vulnerable to the action of attackers against their life, or to the access to personal information. One of the requirements for a proper functioning of e-health/m-health/TMIS and other systems for IoT is reduction in both the consumption of computational and communication resources for energy-savings and the congestion on communication channels, given the large number of new emerging devices. Most devices destined to e-health/m-health and IoT are small, e.g., sensors, and do not show high processing capacity and long battery life. Therefore, computational costs must be reduced for the optimization of power resources.

Device-to-Device (D2D) communication provides a direct connection of devices with or without the intervention of a traditional network infrastructure (e.g., 3GPP standards). Therefore, the ability of connecting devices can provide data offload through nearby devices, thus reducing problems, such as congestion and scarcity of spectrum, and expanding network coverage by enabling devices to relay their data. D2D communication is promising for 5G technology and IoT due to its adaptation to support small and resource-constrained devices predicted by those two technologies. However, security schemes for D2D communication are still in initial development steps, which require more research and studies for their improvements and consolidation, and authentication and key agreement protocols adapted to them.

D2D is suitable for e-health/m-health/TMIS, since it can accelerate the transmission of data and provide a connection to devices located outside the coverage of 3GPP networks. This might be the key for the success of e-health/m-health/TMIS applications, because most data exchanged provide information of patients’ health, e.g., heartbeats, blood sugar and pressure, which is sensible to delays for saving lives. Moreover, since e-health/m-health/TMIS devices are mostly resource-constrained, they require adapted traditional authentication protocols that consider their limitations and avoid costly data exchanges and computations. Therefore, new authentication and key agreement protocols can be designed towards fulfilling such requirements, when used for e-health/m-health/TMIS, while being secure and light to not overload them.

3.1.1 Main Contributions

The main contributions of the protocol proposed in this chapter involve:

a) a new symmetric cryptography-based mutual authentication and key agreement protocol for e-health/m-health/TMIS that is adapted to support D2D communications;

b) the guarantee offered by the proposed protocol of several security properties (e.g., confidentiality and anonymity) and resistance to attacks, such as replay, denial-of-service and man-in-the-middle;

c) computational, communication and energy costs evaluation and comparison with other authentication protocols, which demonstrated the scheme proposed provided the best results.

d) semi-formal validation of the proposed protocol, using Automated Validation of Internet Security Protocols and Applications (AVISPA) [18].

The new cloud-based mutual authentication and key agreement protocol for e-health/m-health/TMIS systems has been adapted for D2D communication towards reducing computational and communication resources consumption, in comparison with other protocols from the literature.

3.1.2 Structure of the chapter

Section 3.2 describes some related works; Section 3.3 introduces the protocol; Sections 3.4 and 3.5 address security and performance analyses, respectively; Section, 3.6

describes a semi-formal validation of the protocol; finally, Sections 3.7 and 3.8 provide the conclusions and future work and the references, respectively.

3.2. Related Work

Chiou et al. [2] and Mohit et al. [1] considered a cloud server an auxiliary entity that stores patient data, such as measures collected from sensors coupled to their bodies. Such data are encrypted and transmitted over public channels, from the entities involved to the cloud server and vice versa, after the execution of mutual authentication and generation of a session key.

The authors designed protocols based on asymmetric and symmetric cryptography and composed of four phases, namely health center upload (HUP), patient upload (PUP), treatment (TP) and checkup (CP). A security analysis conducted revealed some issues in the protocol of Chiou et al. [2]. According to Mohit et al. [1], it fails to preserve the system anonymity and security if the patient's device is lost or stolen. On the other hand, the protocol of Mohit et al. [1] fails to avoid the Denial of Service (DoS) attack.

Jiang et al. [3] and Li et al. [4] also developed interesting approaches. Although the protocols considered no an auxiliary cloud server, (the entities authenticate themselves directly with the health center server through the Internet), they were developed for e-health/m-health/TMIS, similarly to the protocol proposed in this chapter. The proposal of Li et al. [4] is based on asymmetric cryptography, whereas the one designed by Jiang et al. [3] is based on symmetric cryptography. Both are composed of three phases in common, namely Initialization, Registration and Authentication. Li et al. [4] accomplished all the security objectives considered in the security analysis section of this manuscript. However, the proposal of Jiang et al. [3] is vulnerable to the loss/stealing of a patient's device and shows some lack of confidentiality.

The protocols of Jiang et al. [5], Amin et al. [6] and Shen et al. [7] differ from those of Chiou et al. [2] and Mohit et al. [1] because they consider only the communication channel between the user (patient) and the cloud. They also employ asymmetric cryptography based on Elliptic Curves Discreet Logarithm Problem (ECDLP) and comprise three phases, namely initialization, registration and login/authentication. Jiang et al. [5] and Amin et al. [6] accomplished all the security objectives analyzed in this study, however, the protocol of Shen et al. [7] shows some security issues, such as lack of confidentiality and vulnerability to patient trackability due to loss/stealing of the patient's mobile device.

Gunes et al. [8] proposed a hybrid model for LTE network assisted D2D discovery and communication towards the integration of D2D into the current 3GPP LTE architecture through the development of a device's direct discovery model and optimization of the establishment of communications. It is based on the Proximity Services (ProSe) standard developed by 3GPP and its security requirements for D2D communication.

Zhang et al. [9] developed an m-health authentication scheme for D2D communication. Based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), it is a certificateless signcryption scheme (CLGSC) that considers the necessity of protecting data from eavesdropping on the relays involved in D2D communication. However, it differs from ours because it does not consider a cloud server as an auxiliary in the scheme.

The protocol proposed in this chapter uses D2D communication for e-health/m-health/TMIS for enabling the transmission of large amounts of data, such as health reports with images, sound and video, between devices in a short range. It can accomplish high data rate and lower energy consumption in comparison with traditional access technologies (e.g., 3GPP LTE, according to Kar and Sanyal [10].

D2D communication enables patients' devices to connect directly to a medical entity to send health data collected by sensors and receive diagnosis faster than in the traditional way. The constant monitoring of patients and analyses of health reports are crucial for the avoidance of medical conditions, such as strokes and heart attacks, because the chances of a person being sick can be detected much faster. 3GPP has started to standardize D2D communication for its network architecture and developed several technical reports (TR) and technical specifications (TS) (e.g., TS 33.303 [11], TS 23.303 [12] and TR 36.843 [13], which describe security aspects, device discovery and configuration for D2D communication.

The literature reports several authentications and key agreements for D2D communication, they are not designed for m-health environments. It is the case in the works of Wang and Yan [14] and Hsu et al. [15]. However, they are not designed for m-health environments. Wang and Yan [14] developed two authentication protocols for D2D, one based on hash Message Authentication Code (HMAC) and the other based on Identity-Based Signatures (IBS). Hsu et al. [15] proposed a group authentication protocols for D2D based on Identity-Based Encryption (IBE) and Diffie Hellman Key Exchange (DHKE). Table 3.1 shows a comparison among some studies relevant for the design of the protocol proposed in this chapter.

Table 3.1. Comparison among some protocols.

	D2D Communication	m-health/e-health/TMIS	Type of Cryptography	Cloud Server
Chiou et al. [2]	No	Yes	Asymmetric	Yes
Mohit et al. [1]	No	Yes	Asymmetric	Yes
Jiang and Lian [3]	No	Yes	Symmetric	No
Li et al. [4]	No	Yes	Asymmetric	No
Wang and Yan [14]	Yes	No	Asymmetric	No
Hsu et al. [15]	Yes	No	Asymmetric	No
Zhang et al. [9]	Yes	Yes	Asymmetric	No
Proposed Protocol	Yes	Yes	Symmetric	Yes

3.3 Proposed Protocol

The protocol proposed in this chapter is based on challenge-response and was developed as a secure and efficient mutual authentication scheme alternative, without incurring high computational and communication costs. The use of symmetric cryptography may generate security issues due to key exchanges over public channels. However, the protocol does not exchange keys or real identities over insecure channels, as explained in sections 3.4.4 and 3.4.7, and consequently, it is not affected by such problems. Symmetric cryptography and challenge response are adopted in this chapter because they can provide secure authentication with lower costs when compared to asymmetric cryptography. We also propose a D2D communication environment that enables devices inside the 3GPP network to perform data offloading and those outside the coverage area to be connected and send their owner’s health reports.

Figure 3.1 shows the system architecture, composed of a health center, a cloud server, patients with and without sensors, patients’ devices, doctors, the 3GPP access technology, Evolved Node B (eNB) and 3GPP Evolved Packet Core (EPC), represented by the Home Subscriber Server (HSS). It is also comprised of two coverage domains: a device’s coverage domain, comprehending devices located both inside the 3GPP coverage area and outside the coverage area (patients located outside the coverage area can access the 3GPP network relaying their data through devices located inside the coverage area), and the 3GPP domain, where the doctor is located.

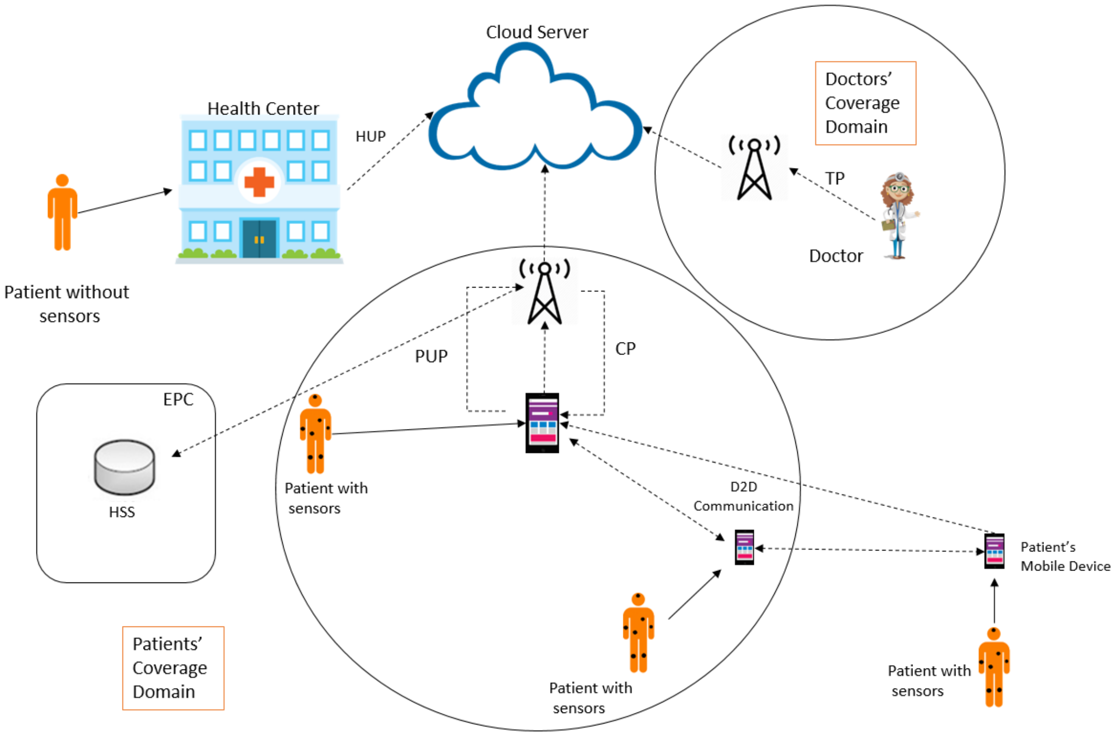


Figure 3.1. Architecture of the proposed scheme.

Patients without sensors visit the health center to collect identity information to be used in future mutual authentication sessions. The health center must perform mutual authentication with the cloud server prior to sending its patients' data. Patients' devices perform mutual authentication with the cloud server prior to sending the data collected from the respective sensors. Devices with direct access to the 3GPP infrastructure might use it to reach the cloud server. Those devices located outside the coverage area can perform mutual authentication using D2D communication, prior to sending health reports. In the second case, the other D2D devices in the path to the cloud server are used as relays. A device with direct connection to the 3GPP infrastructure might choose to send its information through D2D communication to perform data offload, which is not addressed in this study.

Finally, each doctor also performs mutual authentication to obtain patients' reports, evaluate their health conditions and guide them to the most suitable treatment.

Table 3.2. Notations used in the protocol.

Symbol	Description
x, y	Entities: patient (P), health center (H), doctor (D), cloud server (C).
ID_x /TID_x	Real identity of entity x/ Temporary identity of entity x.
k	Random numbers generated in the registration phase.
R_k	k random number generated.
MACxy	Message Authentication Code generated from entity x to entity y.
R_x	Random number generated by entity x.
R_{Cy}	Random number generated by the cloud and sent to entity y.
T_x	Timestamp generated by entity x.
K_{xy}	Session key generated by entities x and y.
C_{xy}	Validator of the session key generated by x and y.
E_{Kxy} /D_{Kxy}	Encryption/Decryption operation that used the session key generated by x and
$IMSI_x$	International Mobile Subscriber Identity of device x
h_1	Temporary identity generation hash function.
h_2	MAC generation hash function.
h_3	Session key generation hash function.
h_4	Session key verifier hash function.
⟶	Secure channel.
⇢	Insecure channel.

The following sections detail each of the phases required for the mutual authentication of the entities considered and the cloud server, named registration, health center upload, patient upload, treatment and checkup. Table 3.2 shows the notations used.

3.3.1 Device Discovery Scheme

The devices must perform a device discovery to detect and identify devices in proximity [8] for establishing D2D communication. 3GPP technical specification TS 23303 [12] describes two models of devices discovered with no permission necessary from the UE to be discovered or with authorization required. The first is model Model A “I am here”, in which devices broadcast some information to announce their existence and monitor if interested have devices also shared their information. In the second, i.e., Model B “Who is there?”, devices work as discoverers by broadcasting the characteristics they expect to find in the nearby devices and wait for the response of those eligible to fulfill their expectations.

We have adopted Model A and the device discovery follows the solution presented in Gunes et al. [8] and the technical specification TS 23303 [12], which is described as below:

Each device must prove authentic to the HSS, which checks if its International Mobile Subscriber Identity (*IMSI*) has matched the identity of the device registered in the database and if the device is eligible to perform D2D. If the verification succeeds, the device performs D2D communication. The authorization is stored in the eNB and refreshed at the expiration of a validity timer.

Next, it adopts a model with direct discovery among devices through a dedicated ProSe server, one of the solutions presented by [8] and based on the specifications of TS 23303 [12]. The devices detect and identify each other using E-UTRAN or WLAN direct radio signals to share their identities.

3.3.2 Registration Phase

This phase enables the exchange of important authentication-related parameters used in the subsequent phases. The International Mobile Subscriber Identity (*IMSI*) of each device must be registered occurs offline in the Home Subscriber Server (HSS) by the manufacturer.

The health center, patients and doctors are then registered in the cloud server through a secure channel. Each entity generates *k* different random numbers R_k and calculates a set of temporary identities, $TID_x = h_1(ID_x \,||\, R_k)$, which are individually used in each authentication session initiated by the entities. The use of real identities associated with a random number in the calculation of temporary identities guarantees their uniqueness. They send their real identity ID_x, and temporary identities TID_x to the cloud server, which stores the data to be used in the following phases. If all temporary identities of a certain entity are used, a new registration phase is performed. If a real identity is revoked, a special registration phase is performed to indicate the identity revoked and the new equivalent identity. Only registered entities can perform the following phases.

3.3.3 Health Center Upload Phase (HUP)

An insecure channel is considered for this phase. The aim is the mutual authentication among entities for a secure transmission of the patient's collected data, from the health center to the cloud server. The complete procedure is shown in Figure 3.2. The phase starts when the user goes to the health center for a health inspection and receives a login and a password to access the patient's system in his/her mobile device. Patients can access his/her health information whenever wanted by inserting the login/password pair on their device.

Step 1. The health center selects a TID_H and generates a random number R_H. Then, it calculates $MAC_{HC} = h_2(ID_H \parallel R_H)$ and sends *Message 1* = (TID_H, R_H, MAC_{HC}) to the cloud server with a timestamp T_H.

Step 2. After receiving *Message 1* and T_H from the health center, the cloud server verifies if T_H is valid. If the verification fails, the procedure is terminated. Otherwise, the cloud server calculates $MAC_{HC}' = h_2(ID_H \parallel R_H)$ using the real identity of the health center received in the registration phase and the random number received in *Message 1*. It then verifies if $MAC_{HC}' = MAC_{HC}$. If the verification fails, the procedure ends because an intruder has been detected. Otherwise, the cloud server authenticates the health center, selects a random number R_{CH}, calculates $MAC_{CH} = h_2(ID_H \parallel R_{CH})$ and sends *Message 2* = (MAC_{CH}, R_{CH}) with a timestamp T_C to the health center.

Step 3. The health center receives *Message 2* and T_C from the cloud server and checks if timestamp T_C is valid. If the validation fails, the procedure ends. Otherwise, the health center calculates $MAC_{CH}' = h_2(ID_H \parallel R_{CH})$ and verifies if $MAC_{CH}' = MAC_{CH}$. If the verification fails, the procedure is terminated because an intruder has been detected. Otherwise, the health center authenticates the cloud server and generates the session key, $K_{HC} = h_3(ID_H \parallel R_H \parallel R_{CH})$ and the session key validator, $C_{HC} = h_4(K_{HC})$. It then uses the session key to encrypt the patient's report, $M_{RP} = E_{KHC}\ (Patient\ Report,\ TID_P,\ C_{HC})$ and finally sends *Message 3* = M_{RP} and a new timestamp T_H to the cloud server.

Step 4. The cloud server receives {Message 3, T_H} and verifies T_H. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{HC} = h_3(ID_H \parallel R_H \parallel R_{CH})$ and decrypts the patient's report, $(Patient\ Report,\ TID_P,\ C_{HC}) = D_{KHC}(M_{RP})$. It then calculates $C_{HC} = h_4(K_{HC})$ and verifies if $C_{HC}' = C_{HC}$. If the verification fails, it ends the procedure. Finally, the cloud server stores the patient´s report with the respective identities.

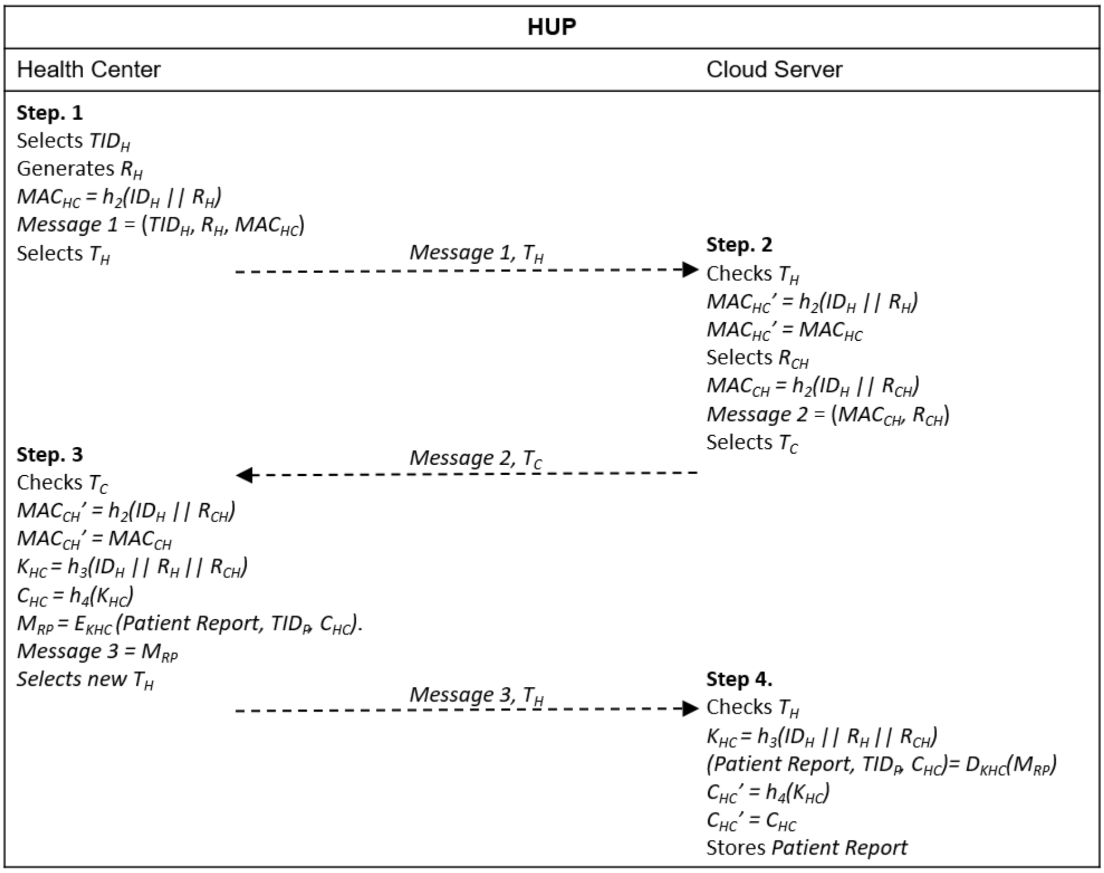

HUP
Health Center
Cloud Server
Step. 1
Selects TID_H
Generates R_H
$MAC_{HC} = h_2(ID_H \mid\mid R_H)$
$Message\ 1 = (TID_H, R_H, MAC_{HC})$
Selects T_H
$Message\ 1, T_H$
Step. 2
Checks T_H
$MAC_{HC}' = h_2(ID_H \mid\mid R_H)$
$MAC_{HC}' = MAC_{HC}$
Selects R_{CH}
$MAC_{CH} = h_2(ID_H \mid\mid R_{CH})$
$Message\ 2 = (MAC_{CH}, R_{CH})$
Selects T_C
$Message\ 2, T_C$
Step. 3
Checks T_C
$MAC_{CH}' = h_2(ID_H \mid\mid R_{CH})$
$MAC_{CH}' = MAC_{CH}$
$K_{HC} = h_3(ID_H \mid\mid R_H \mid\mid R_{CH})$
$C_{HC} = h_4(K_{HC})$
$M_{RP} = E_{KHC}(Patient\ Report, TID_P, C_{HC})$.
$Message\ 3 = M_{RP}$
Selects new T_H
$Message\ 3, T_H$
Step 4.
Checks T_H
$K_{HC} = h_3(ID_H \mid\mid R_H \mid\mid R_{CH})$
$(Patient\ Report, TID_P, C_{HC}) = D_{KHC}(M_{RP})$
$C_{HC}' = h_4(K_{HC})$
$C_{HC}' = C_{HC}$
Stores *Patient Report*

Figure 3.2. Message exchange in HUP.

3.3.4 Patient Upload Phase (PUP)

The PUP phase is performed over an insecure channel, and its focus is on the mutual authentication between patients and the cloud server and generation of a session key to encrypt health information measured by the sensors attached to the user's body, prior to sending it to the cloud server. The complete procedure is shown in Figures 3.3 and 3.4. The phase starts when the patient's device requests the health information measures collected to the sensors attached to user's body and stores them.

If necessary, the device discovery is performed for the finding of other nearby devices, based on proximity. However, first they must be authenticated by a 3GPP network to prove their reliability. All devices interested in performing D2D communication generate a random number R_P, calculates and sends the hash of its IMSI to the HSS to be authenticated: $Auth_p = h_1(IMSI_p \mid\mid R_P)$.

HSS receives each $Auth_p$, calculates $Auth'_p = h_1(IMSI_p \mid\mid R_P)$ and verifies if $Auth_p = Auth'_p$. If the verification succeeds, it authenticates the device. All devices authenticated by HSS can perform D2D.

Devices interested in D2D broadcast their $\mathrm{TID}_{\mathrm{Di\text{-}j}}$ to reach other devices nearby, thus, showing intention to establish connection with them. Next, they send their own temporary identities to signalize their existence and position. A device located outside the coverage area, or inside it but with no access to the 3GPP network can perform their authentication with the cloud server by relaying their messages through the nearby devices, until the 3GPP network has been reached.

Step 1. The device calculates $MAC_{PC} = h_2(ID_P \,||\, R_P)$ and sends *Message 1* = (TID_P, R_P, MAC_{PC}) with a timestamp T_P to the cloud server. A device with direct access to the 3GPP network can choose between sending data directly or to performing offload through D2D communication until the cloud server has reached. Devices with no 3GPP coverage send their data through D2D communication.

Step 2. The cloud server receives *Message 1* and T_P and verifies if T_P is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{PC}' = h_2(ID_P \,||\, R_P\ )$ and verifies if $MAC_{PC}' = MAC_{PC}$. If the verification fails, the procedure is interrupted. Otherwise, the cloud server authenticates the device, selects a random number R_{CP}, calculates $MAC_{CP} = h_2(ID_P \,||\, R_{CP})$ and sends *Message 2* = (MAC_{CP}, R_{CP}) with a timestamp T_C to the patient.

Step 3. After receiving *Message 2* and T_C from the cloud server, the patient checks if T_C is valid. If the validation fails, the procedure ends. Otherwise, it calculates $MAC_{CP}' = h_2(ID_P \,||\, R_{CP})$ and verifies if $MAC_{CP}' = MAC_{CP}$. If the verification fails, the procedure is terminated. Otherwise, the patient authenticates the cloud server, generates the session key $K_{PC} = h_3(ID_P \,||\, R_P \,||\, R_{CP})$ and calculates $C_{PC} = h_4(K_{PC})$. He/she then encrypts the sensors measures using the session key, $M_{MS} = E_{KPC}$ *(Sensors Measures*, TID_P, $C_{PC})$ and sends *Message 3* = M_{MS} with a new timestamp T_P to the cloud server.

Step 4. The cloud server receives {*Message 3*, T_P} and verifies if T_P is valid. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{PC} = h_3(ID_P \,||\, R_P \,||\, R_{CP})$, decrypts the sensors measures, *(Sensors Measures*, TID_P, $C_{PC})$ = $D_{KPC}(M_{MS})$, calculates $C_{PC} = h_4(K_{CP})$ and verifies if $C_{PC}' = C_{PC}$. If the verification fails, it terminates the procedure. Otherwise, the cloud server stores the sensors' measures with the respective identities.

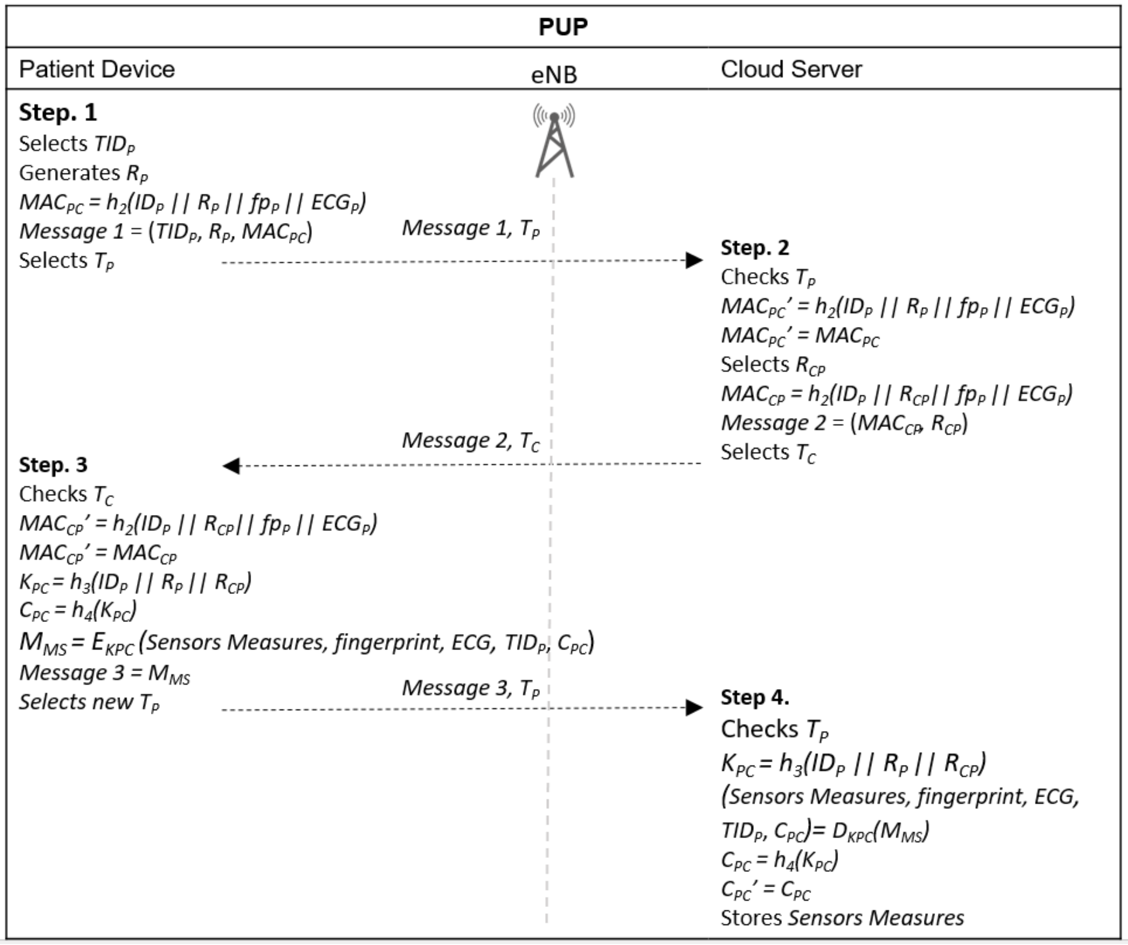

PUP
Patient Device
eNB
Cloud Server
Step. 1
Selects TID_P
Generates R_P
$MAC_{PC} = h_2(ID_P \,||\, R_P \,||\, fp_P \,||\, ECG_P)$
$Message\ 1 = (TID_P, R_P, MAC_{PC})$
Selects T_P
$Message\ 1, T_P$
Step. 2
Checks T_P
$MAC_{PC}' = h_2(ID_P \,||\, R_P \,||\, fp_P \,||\, ECG_P)$
$MAC_{PC}' = MAC_{PC}$
Selects R_{CP}
$MAC_{CP} = h_2(ID_P \,||\, R_{CP} \,||\, fp_P \,||\, ECG_P)$
$Message\ 2 = (MAC_{CP}, R_{CP})$
Selects T_C
$Message\ 2, T_C$
Step. 3
Checks T_C
$MAC_{CP}' = h_2(ID_P \,||\, R_{CP} \,||\, fp_P \,||\, ECG_P)$
$MAC_{CP}' = MAC_{CP}$
$K_{PC} = h_3(ID_P \,||\, R_P \,||\, R_{CP})$
$C_{PC} = h_4(K_{PC})$
$M_{MS} = E_{KPC}$ (Sensors Measures, fingerprint, ECG, TID_P, C_{PC})
$Message\ 3 = M_{MS}$
Selects new T_P
$Message\ 3, T_P$
Step 4.
Checks T_P
$K_{PC} = h_3(ID_P \,||\, R_P \,||\, R_{CP})$
(Sensors Measures, fingerprint, ECG, TID_P, C_{PC})= $D_{KPC}(M_{MS})$
$C_{PC} = h_4(K_{PC})$
$C_{PC}' = C_{PC}$
Stores Sensors Measures

Figure 3.3. Message exchange in PUP for direct access to 3GPP infrastructure.

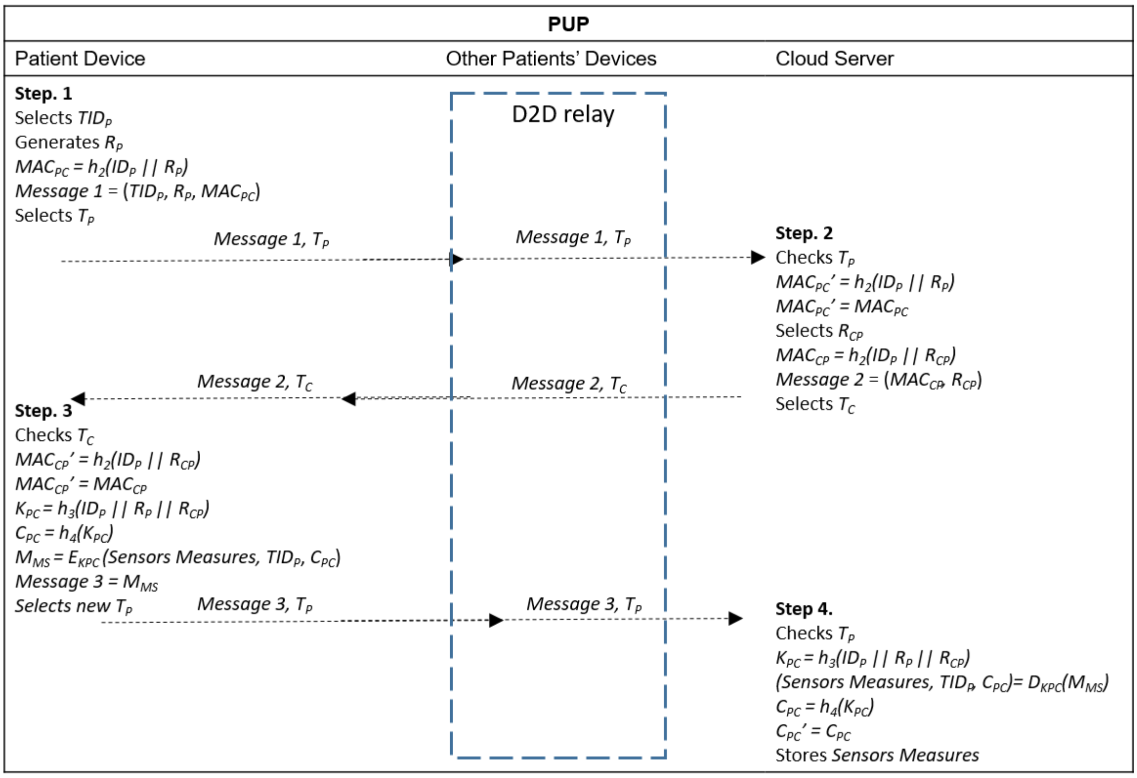


Figure 3.4 Message exchange in PUP when D2D communication is adopted to reach the 3GPP infrastructure and the cloud server.

3.3.5 Treatment Phase (TP)

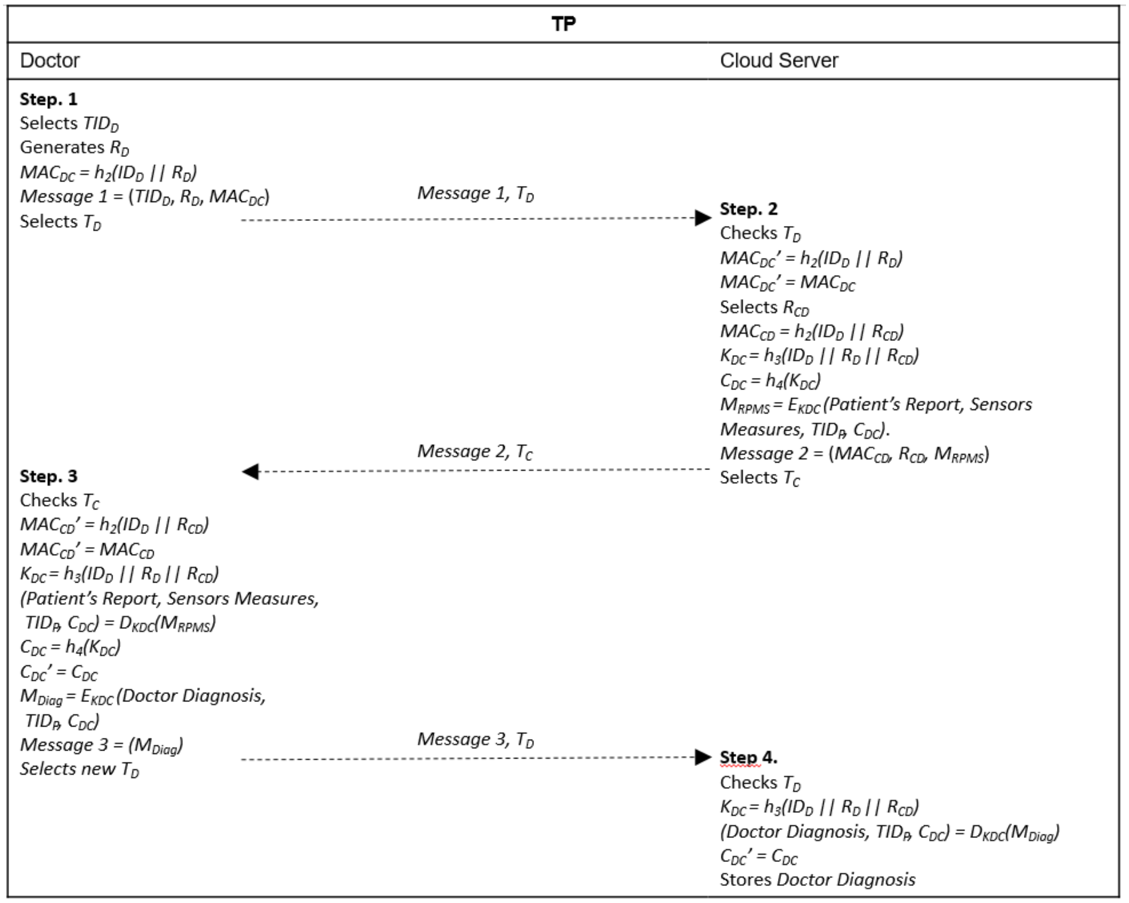
This phase is performed over an insecure channel. It aims at mutual authentication between the doctor and the cloud server and generation of a session key for encrypting the patient's health report and sensors' measures before they are sent to the doctor, and encrypting the doctor's diagnosis before it is sent to the cloud server. The complete procedure is shown in Figure 3.5.

Step 1. The doctor selects one of his/her temporary identities $TID_{D,}$ generates a random number R_D, calculates $MAC_{DC} = h_2(ID_D \mid\mid R_D)$ and sends *Message 1* = (TID_D, R_D, MAC_{DC}) with a timestamp T_D to the cloud server.

Step 2. The cloud server receives {*Message* 1, T_D} and verifies if T_D is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{DC}' = h_2(ID_D \mid\mid R_D)$ and verifies if $MAC_{DC}' = MAC_{DC}$. If the verification fails, the procedure is interrupted. Otherwise, the cloud server authenticates the doctor, selects a random number R_{CD} and calculates $MAC_{CD} = h_2(ID_D \mid\mid R_{CD})$, a session key $K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$ and $C_{DC} = h_4(K_{DC})$. It then uses the doctor's real identity to obtain the patient´s report and sensors' health information measures previously stored in the cloud and prepares the information to be sent to the doctor, encrypting the data with the session key calculated, $M_{RPMS} = E_{KHC}$ *(Patient Report, Sensors Measures,* TID_P, C_{DC}*)*. Finally, it sends *Message 2* = (MAC_{CD}, R_{CD}, M_{RPMS}) with a timestamp T_C to the doctor.

Step 3. The doctor receives {*Message* 2, T_C} and checks if T_C is valid. If the validation fails, the procedure ends. Otherwise, the health center calculates $MAC_{CD}' = h_2(ID_D \mid\mid R_{CD})$ and verifies if $MAC_{CD}' = MAC_{CD}$. If the verification fails, the procedure is terminated. Otherwise, the doctor authenticates the cloud server, generates the session key $K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$, decrypts M_{RPMS} to obtain the patient's report and the health information measured by the sensors, *(Patient's Report, Sensors Measures,* $TID_{P,}$ C_{DC}*)* $= D_{KDC}(M_{RPMS})$, calculates $C_{DC}' = h_4(K_{DC})$ and verifies if $C_{DC}' = C_{DC}$. Then, he/she analyzes the data received, generates the patient's diagnosis, encrypts it, $M_{Diag} = E_{KDC}$ *(Doctor Diagnosis,* TID_P*)* and finally sends *Message 3* = M_{Diag} and a new timestamp T_D to the cloud server.

Step 4. After receiving *Message 3* and $T_{D,}$ the cloud server verifies if T_D is valid. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$, $C_{DC}' = h_4(K_{DC})$ and verifies if $C_{DC}' = C_{DC}$. If the verification fails, it interrupts the procedure because the message was not originated from the authenticated doctor and might have been forged by an intruder. If the verification succeeds, the cloud server uses the session key to decrypt the doctor's diagnosis and its respective temporary identity, *(Doctor Diagnosis,* TID_D*)* = $D_{KDC}(M_{Diag})$. Finally, it stores the doctor's diagnosis with its respective identities.



TP

Doctor

Cloud Server

Step. 1
Selects TID_D
Generates R_D
$MAC_{DC} = h_2(ID_D \mid\mid R_D)$
Message 1 = (TID_D, R_D, MAC_{DC})
Selects T_D

Step. 2
Checks T_D
$MAC_{DC}' = h_2(ID_D \mid\mid R_D)$
$MAC_{DC}' = MAC_{DC}$
Selects R_{CD}
$MAC_{CD} = h_2(ID_D \mid\mid R_{CD})$
$K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$
$C_{DC} = h_4(K_{DC})$
$M_{RPMS} = E_{KDC}$*(Patient's Report, Sensors Measures,* TID_P, C_{DC}*).*
Message 2 = $(MAC_{CD}, R_{CD}, M_{RPMS})$
Selects T_C

Step. 3
Checks T_C
$MAC_{CD}' = h_2(ID_D \mid\mid R_{CD})$
$MAC_{CD}' = MAC_{CD}$
$K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$
(Patient's Report, Sensors Measures, TID_P, C_{DC}*)* = $D_{KDC}(M_{RPMS})$
$C_{DC} = h_4(K_{DC})$
$C_{DC}' = C_{DC}$
$M_{Diag} = E_{KDC}$*(Doctor Diagnosis,* TID_P, C_{DC}*)*
Message 3 = (M_{Diag})
Selects new T_D

Step 4.
Checks T_D
$K_{DC} = h_3(ID_D \mid\mid R_D \mid\mid R_{CD})$
(Doctor Diagnosis, TID_P, C_{DC}*)* = $D_{KDC}(M_{Diag})$
$C_{DC}' = C_{DC}$
Stores *Doctor Diagnosis*

Figure 3.5. Message exchange in TP.

3.3.6 Checkup Phase (CP)

This phase is performed over an insecure channel and aims at a new mutual authentication between the patient and the cloud server and generation of a new session key for encrypting the doctor's diagnosis, before the cloud sends it to the patient. The complete procedure is shown in Figure 3.6.

Step 1. The patient generates a new random number R_{PCP}, calculates $MAC_{PCP} = h_2(ID_P \mid\mid R_{PCP})$ and sends *Message 1* = (TID_P, R_{PCP}, MAC_{PCP}, *Request*) with a timestamp T_P to the cloud server. Devices with direct access to the 3GPP network can send their data directly or use D2D communication to reach the cloud server. Devices with no 3GPP coverage must send their data through D2D communication.

Step 2. After receiving *Message* 1 and T_P, the cloud server verifies if T_P is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{PCP}' = h_2(ID_P \mid\mid R_{PCP})$ and verifies if $MAC_{PCP}' = MAC_{PCP}$. If the verification fails, the procedure ends.

Otherwise, it authenticates the patient, selects a random number R_{CCP}, calculates $MAC_{CCP} = h_2(ID_P \,||\, R_{CCP})$, generates the session key $K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$ and computes $C_{PCP} = h_4(K_{PCP})$. It then uses the session key to encrypt the doctor's diagnosis, $M_{DiagP} = E_{KPCP}$ *(Doctor's Diagnosis, TID*$_P$, C_{PCP}*)* and sends to the patient *Message 2* = (MAC_{CCP}, R_{CCP}, M_{DiagP}) with a timestamp T_C.

Step 3. The patient receives {*Message* 2, T_C} and checks if T_C is valid. If the validation fails, the procedure is terminated. Otherwise, he/she calculates $MAC_{CCP}' = h_2(ID_P \,||\, R_{CCP})$ and verifies if $MAC_{CCP}' = MAC_{CCP}$. If the verification fails, the procedure is interrupted. Otherwise, he/she authenticates the cloud server, generates the session key $K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$, decrypts the doctor's diagnosis, *(Doctor's Diagnosis, TID*$_P$, C_{PCP}*)* $= D_{KPCP}(M_{DiagP})$, calculates $C_{PCP} = h_4(K_{PCP})$ and verifies if $C_{PCP}' = C_{PCP}$. If the verification fails, it ends the procedure is ended. Otherwise, the patient stores the doctor's diagnosis and looks for a convenient treatment.

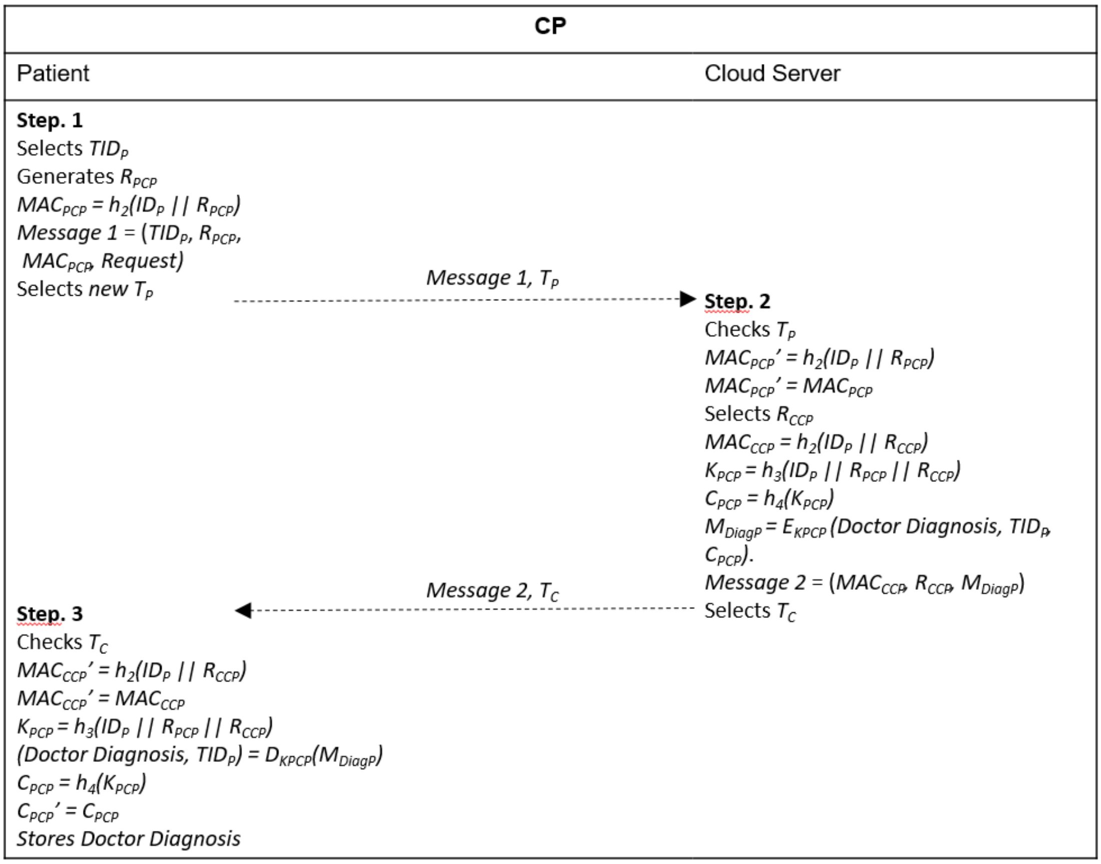

CP
Patient
Cloud Server
Step. 1
Selects TID_P
Generates R_{PCP}
$MAC_{PCP} = h_2(ID_P \,||\, R_{PCP})$
$Message\ 1 = (TID_P, R_{PCP}, MAC_{PCP}, Request)$
Selects $new\ T_P$
$Message\ 1, T_P$
Step. 2
Checks T_P
$MAC_{PCP}' = h_2(ID_P \,||\, R_{PCP})$
$MAC_{PCP}' = MAC_{PCP}$
Selects R_{CCP}
$MAC_{CCP} = h_2(ID_P \,||\, R_{CCP})$
$K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$
$C_{PCP} = h_4(K_{PCP})$
$M_{DiagP} = E_{KPCP}(Doctor\ Diagnosis, TID_P, C_{PCP}).$
$Message\ 2 = (MAC_{CCP}, R_{CCP}, M_{DiagP})$
Selects T_C
$Message\ 2, T_C$
Step. 3
Checks T_C
$MAC_{CCP}' = h_2(ID_P \,||\, R_{CCP})$
$MAC_{CCP}' = MAC_{CCP}$
$K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$
$(Doctor\ Diagnosis, TID_P) = D_{KPCP}(M_{DiagP})$
$C_{PCP} = h_4(K_{PCP})$
$C_{PCP}' = C_{PCP}$
Stores Doctor Diagnosis

Figure 3.6. Message exchange in CP.

3.4. Security Analysis

This section presents the security objectives accomplished by the protocol. Table 3.3 shows a security comparison among the protocol and those designed by Chiou et al. [2] and Mohit et al. [1].

3.4.1. Mutual Authentication

In the proposed protocol, each entity calculates a MAC to perform mutual authentication with the cloud server and vice versa. For example, in the HUP phase, the health center calculates $MAC_{HC} = h_2(ID_H \| R_{H)}$ and sends it to the server cloud, which calculates $MAC_{HC}' = h_2(ID_H \| R_H)$ and verifies if $MAC_{HC}' = MAC_{HC}$. If the verification is successful, the server cloud authenticates the health center, calculates its own $MAC_{CH} = h_2(ID_H \| R_{CH})$ and sends it to the health center, which calculates $MAC_{CH}' = h_2(ID_H \| R_{CH})$ and verifies if $MAC_{CH}' = MAC_{CH}$. If the verification succeeds, the health center authenticates the server cloud and the mutual authentication procedure is complete. A similar procedure is performed in the PUP, TP and CP phases.

3.4.2 Forward/Backward Secrecy

The forward and backward secrecies are guaranteed by the use of random values (R_H, R_{CH}, R_P, R_{CP}, R_D, R_{CD}, R_{PC}, R_{CPC}) newly generated in each authentication session, during the calculation of the system keys, as the one generated in the PUP phase $K_{CP} = h_3(ID_P \| R_P \| R_C)$. Therefore, if an intruder discovers old system keys, it cannot use them in future authentication sessions (backward secrecy). On the other hand, if an intruder discovers future system keys, it cannot use them in past authentication sessions (forward secrecy).

3.4.3 Confidentiality

The system’s confidentiality is guaranteed by the access control of the patient’s mobile device. A possible user must insert login and password to access his/her information in the system. Consequently, sensitive information is available only to authorized users. An authentication procedure is performed between the cloud and an entity in each phase for the generation of a session key that will encrypt the patient’s data before they are exchanged on a public channel.

3.4.4 Non-Repudiation

At the beginning of each phase in the protocol, the entities send the cloud their temporary identities (TID_H, TID_P, TID_D) and a MAC calculated with their real identities (ID_H,

ID_P, ID_D). The cloud also sends the entities a MAC containing their real identities. Since real identities are known only by the cloud and each respective entity, a valid MAC can be generated only by them. The session keys established among the cloud and the entities also depend on their real identity, therefore, neither the cloud, nor the entities can deny the message they originated.

3.4.5 Anonymity

Anonymity is assured only by entities' temporary identities (TID_H, TID_P, TID_D), while messages are exchanged on an insecure channel during the authentication procedure, which protects their real identities. The identity of the cloud server is protected because it is not used in the authentication procedure, hence, not exchanged on an insecure channel.

3.4.6 Non-Traceability

The use of different temporary identities and newly generated random numbers in each new authentication session generates different parameters exchanged. Therefore, outsiders cannot track patients by the parameters exchanged on a public channel.

3.4.7 Session Key Security

Session keys are not exchanged on a public channel, but securely calculated on each side involved in the authentication. Moreover, the security of the session keys established in each phase of the protocol is guaranteed through the use of entities' real identities in the calculation, some secret information known only by the cloud server and the respective entities. For example, in HUP, the session key calculated is $KHC = h_3(IDH \parallel RH \parallel RCH)$, consequently, an intruder cannot obtain or calculate a valid session key.

3.4.8 Patient's mobile device loss/stealing

The security objective is accomplished through the access control of the patient's mobile device using login and password. The system is accessible only if a valid login and password pair is inserted. If the mobile device is stolen or lost, no unauthorized person can access the patient's system, because it would not have a valid login and password pair.

3.4.9 Impersonation Attack

The impersonation attack is avoided because neither the cloud server's real identity, nor the entities' real identities are disclosed. Therefore, an attacker cannot impersonate them and generate a valid MAC, because its calculation depends on the entities' real identities.

3.4.10 Replay Attack

The replay attack is avoided because all entities involved in the proposed protocol use different random values freshly calculated in each authentication process. Therefore, an attacker cannot forge messages using old random values.

3.4.11 Denial of Service (DoS)

The prevention of this attack involves the inclusion of a verification parameter in each message exchanged in the authentication phases (HUP, PUP, TP, CP). The parameter used in the protocol proposed in this chapter was a timestamp and its validity has been verified before the recipient processed each message. Therefore, if an attacker uses an invalid timestamp, the entire procedure is interrupted in time to prevent the DoS attack.

3.4.12 Man-in-the-Middle Attack

No intruder can perform a man-in-the-middle attack, because the session key cannot be forged with the use of only the parameters exchanged on the insecure communication channel. The session key calculation uses the entities' real identities, which is a secret value not disclosed in the insecure channel.

According to Table 3.3, the protocol designed by Chiou et al. [2] does not guarantee anonymity, non-traceability and resistance to patient's mobile device loss/stealing, which are three critical failures. First, as detected by Mohit et al. [1], in the protocol of Chiou et al. [2], the patient's real identity is sent in plain text through a public channel, which compromises its anonymity. We observed it also affects the patient's non-traceability. Second, as detected by Mohit et al. [1], the proposal of Chiou et al. [2] fails to be resistant to patient's mobile device loss/stealing, because it does not perform access control and requests login and password to the user, which makes it vulnerable to the access of non-authorized people and hampers its confidentiality.

Table 3.3. Comparison of security objectives among protocols

Security Objectives	Chiou et al. [2]	Mohit et al. [1]	Proposed Protocol
Mutual Authentication	Yes	Yes	Yes
Forward/Backward Secrecy	Yes	Yes	Yes
Confidentiality	No	Yes	Yes
Non-Repudiation	Yes	Yes	Yes
Anonymity	No	Yes	Yes
Patient's Non-Traceability	No	Yes	Yes
Session Key Security	Yes	Yes	Yes
Resistance to patient's mobile device loss/stealing	No	Yes	Yes
Resistance to Impersonation attack	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes
Resistance to Denial of Service (DoS)	Yes	No	Yes
Resistance to man-in-the-middle attack	Yes	Yes	Yes

The protocol of Mohit et al. [1] fails to prevent DoS attack. No initial verification parameter is generated (timestamp, nonce, sequence number) is generated to be sent with the parameters exchanged. The validity of a simpler parameter is not verified before the recipient processes more complex calculations. Therefore, the protocol is vulnerable to DoS attacks, because the system of D2D devices is not robust enough to deal with message flooding. The protocol proposed in this chapter accomplished all security objectives analyzed and can, therefore, be considered safer than those designed by Chiou et al. [2] and Mohit et al. [1].

3.5. Performance Analysis

This section addresses a performance analysis of the protocol proposed in this chapter and a comparison with those developed by Chiou et al. [2] and Mohit et al. [1] regarding computational abd communication cost. The analysis evaluated and compared the computational and communication costs. The registration phase of the protocol was not included in the analysis because it is performed over a secure channel, and the comparisons focused on operations executed and parameters exchanged over an insecure channel. “n” is the number of devices executing mutual authentication with the cloud server by a traditional 3GPP, and “m” is the number of devices using D2D communication to perform mutual authentication with the cloud server.

3.5.1 Computational Cost

The execution time in seconds (s) of the operations considered is shown in Table 3.4. Chiou et al. [2] and Mohit et al. [1] adopted those values and performed tests with the following operational characteristics: CPU: Intel (R) Core (TM) 2 Quad Q8300, 2.50Hz; memory: 2GB; operational system: Windows 7 Professional.

Table 3.4. Execution time of each operation considered.

Symbol	Description	Cost (seconds)
T_S	Execute/Verify a Signature	0.3317s
T_P	Bilinear Pairing	0,0621s
T_E	Encrypt/Decrypt (Symmetric)	0.0087s
T_H	One Way Hash Function	0.0005s

All four phases were analyzed, and all operations executed were considered. Table 3.5 shows a comparison of the computational costs among the protocol proposed in this chapter and those of Chiou et al. [2], Mohit et al. [1], details of the operations performed in each phase, and the total time in seconds.

Table 3.5. Computational Cost of the Protocols.

	Chiou et al. [2]	Mohit et al. [1]	Proposed Protocol
HUP	$nT_S + 3nT_P + 2nT_E + 7nT_H$	$nT_S + 3nT_E + 11nT_H$	$2nT_E + 8nT_H$
PUP	$nT_S + 3nT_P + 2nT_E + 9nT_H$	$2nT_S + 2nT_E + 10nT_H$	$4nT_E + 9nT_H$
TP	$2nT_S + 3nT_P + 2nT_E + 8nT_H$	$2nT_S + 2nT_E + 9nT_H$	$4nT_E + 8nT_H$
CP	$nT_S + 2nT_P + 2nT_E + 8nT_H$	$nT_S + 2nT_E + 5nT_H$	$2nT_E + 8nT_H$
TOTAL (s)	$5nT_S + 11nT_P + 8nT_E + 32nT_H$	$4nT_S + 9nT_E + 35nT_H = 1.42n$	$12nT_E + 33nT_H = 0.121n$

The protocol proposed in this chapter required the lowest computational cost, due to the exclusive use of symmetric cryptography (low communication cost) for the authentication procedure, therefore it can performs the necessary operations. Chiou et al. [2] and Mohit et al. [1] conducted some signature operations and bilinear pairing, which incurred higher computational costs.

Figure 3.7 shows a graphic representation of costs that confirms the best performance of the protocol proposed in this chapter regarding computational costs, in compared to [1] and [2].

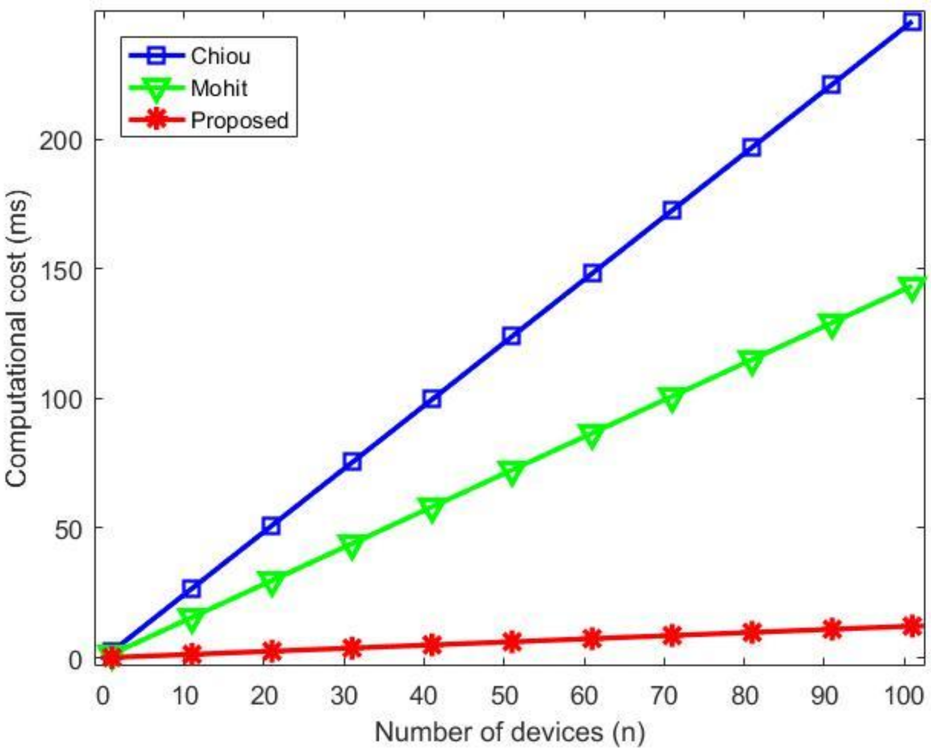


Figure 3.7. Computational cost comparison

3.5.2 Communication Cost

The evaluation of the communication costs considered messages exchanged over an insecure channel and parameters and their respective costs in bits (see Table 3.6).

Table 3.6 – Parameters and costs in bits

Parameter	Cost
Random Number/Identity/Timestamp	48 bits
Bilinear Pairing/Hash	160bits
Symmetric Key	128 bits
Signature (symmetric algorithm)	512 bits

The message exchange over an insecure channel was analyzed in each of the four common phases performed by the protocol proposed in this chapter and those of Chiou et al.

[2] and Mohit et al. [1]. Table 3.7 shows comparisons of each phase and a comparison of the total communication cost of each protocol.

Table 3.7. Comparison of communication costs in bits.

	Chiou et al. [2]	Mohit et al. [1]	Proposed Protocol
HUP	704n	592n	736n
PUP	1600n	1744n	736n + 736m + 208(m-1)
TP	2112n	1792n	864n
CP	1504n	1184n	736n
TOTAL	6920n bits	4832n bits	3072n + 736m + 208(m-1) bits

The protocol proposed in this chapter required the lowest communication cost, hence, the best, due to the reduced number of parameters exchanged and choice of small parameters to be exchanged (identities, random numbers, timestamps) and the adaptation to D2D communication, which offloads part of the traffic outside the 3GPP spectrum. The proposals of Chiou et al. [2] and Mohit et al. [1] required higher communication costs, because of the exchange of some costly signature parameters. The protocol proposed in this chapter achieved the best performance, as revealed by security and performance analyses.

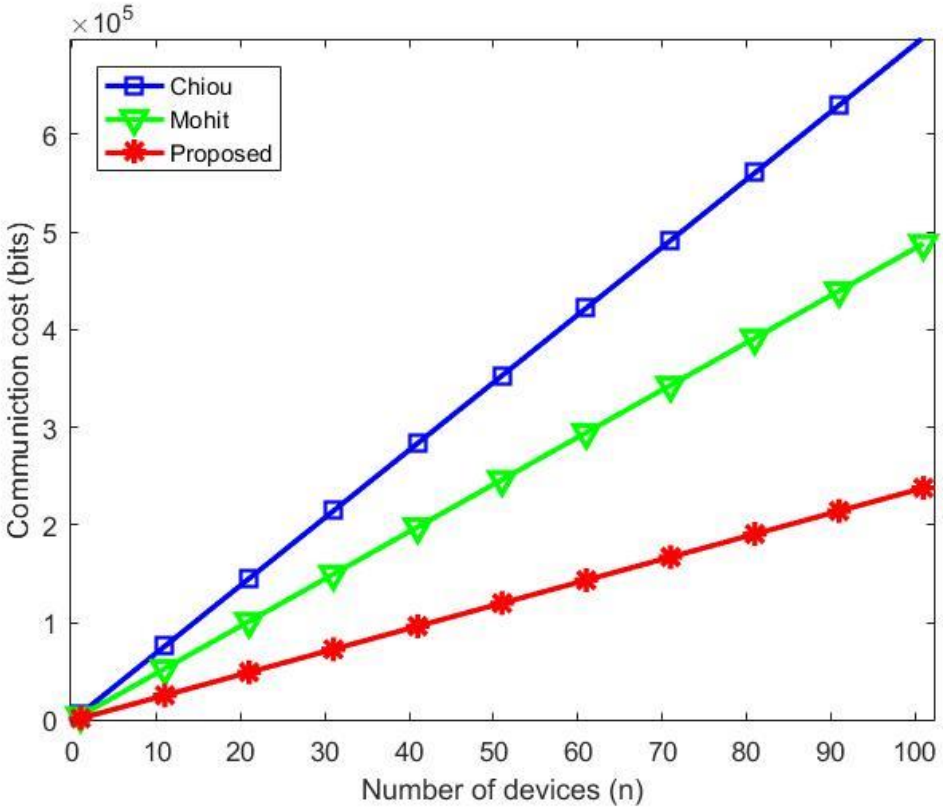


Figure 3.8. Communication cost comparison

Figure 3.8 shows the performance of the protocol proposed in this chapter regarding communication costs. 30% of devices performed offload and used D2D communication for their mutual authentication, which reduced in the traditional 3GPP network.

3.5.3 Energy Cost

Kumar et al. [16] and He et al. [17] proposed an energy cost evaluation that considers the maximum CPU power of devices (W) is approximately 10.88 Watts. The energy overhead was calculated as ETotal = CCTotal × W, where CCTotal is the computational cost calculated and presented in Section 3.5.1. Table 3.8 and Figure 3.9 show the comparison of energy costs among the protocol proposed in this chapter and other protocols from the literature.

Table 3.8. Energy cost of protocols

	Chiou et al. [2]	Mohit et al. [1]	Proposed Protocol
TOTAL	$(5nT_S + 11nT_P + 8nT_E + 32nT_H)$ *10.88 = 26.43n mJ	$(4nT_S + 9nT_E + 35nT_H)$ *10.88 = 15.45n mJ	$(12nT_E + 33nT_H)$ *10.88 = 1.32n mJ

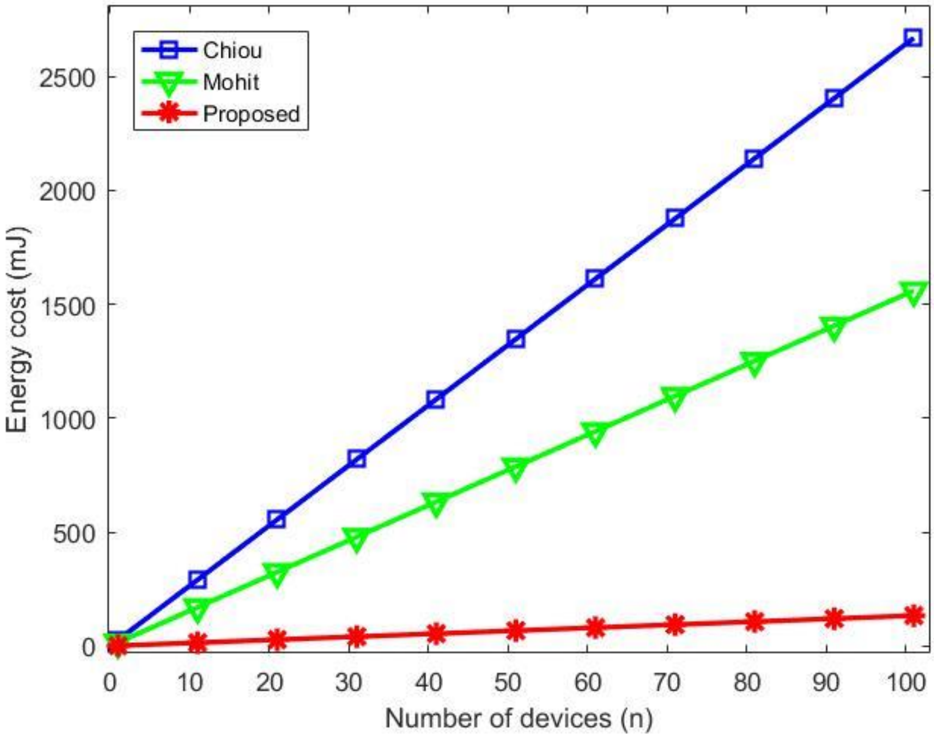


Figure 3.9. Energy cost comparison

According to Figure 3.9, our scheme showed the best performance regarding energy. The energy cost directly related to the computational cost, consequently, the graphic results refer to both costs, and are very similar.

Finally, the good results from the security and performance evaluations have proven the protocol proposed in this chapter can perform better than those of [1] and [2]. Below are some aspects compared:

a) the protocols of Chiou et al. [2] and Mohit et al. [1] are based on asymmetric cryptography, while our approach is based on symmetric cryptography, which produces lower computational and communication costs;

b) the security flaws of Chiou et al. [2] and Mohit et al. [1] are avoided in the protocol proposed in this chapter through the use of access control to the patient's device, timestamps, temporary identities and freshly generated parameters in each authentication session;

c) differently from our scheme, the protocols of Chiou et al. [2] and Mohit et al. [1] do not support D2D communication, which is a promising technology for the development of e-health systems due to its agility in data transmission. The protocol proposed in this chapter takes into consideration the criticality of health systems, which, in some cases, may depend on the agility of data transmission to save lives.

3.6. Validation

The protocol proposed in this chapter was validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) [18]. It is a semi-automated validation tool that verifies the security robustness of authentication protocols by checking the secrecy of key parameters and vulnerability to intruders.

AVISPA employs is made through codes written in High-level Protocol Specification Language (HLPSL) language. The message exchange of the protocol is translated to HLPSL code, and each entity is defined as a communication agent that performs roles, which contains all the parameters exchanged in the messages (States). Those that must remain secret are signalized and observed during the code execution. If no secret value is vulnerable to be discovered by intruders, the protocol is considered safe.

Each of the four phases performed over an insecure channel (HUP, PUP, TP and CP) Was validated, Figure 3.10 presents the role of an ordinary device in the PUP phase, called Dpi in the code. Each State symbolizes the messages sent (SND) and received (RCV), and each parameter that must remain secret is signalized with a flag (e.g., sec_3 and sec_4 in Figure 3.10). The flag SecureChannel flag accompanies and encrypted parameters sent encrypted are signalized as secret(parameter). Figure 3.11 shows the role of the cloud server in PUP phase.

```
role role_Dpi
(Dpi:agent,CS:agent,Dpk:agent,TIDpk:text,Rpi:text,MACpic:text,Tpi:text,MACcpi:text,
Rcpi:text,Tc:text,Smeasure:text,TIDpi:text,Cpic:text,SND,RCV:channel(dy))
played_by Dpi
def=
        local
                State:nat,SecureChannel:symmetric_key
        init
                State := 0
        transition
        1.  State=0 /\ RCV(start) =|>
          State':=1 /\ SND(TIDpi)
        2.  State=1 /\ RCV(TIDpk) =|>
          State':=2 /\ SND(TIDpi.Rpi.MACpic.Tpi)
                4.State=2 /\ RCV(MACcpi.Rcpi.Tc) =|>
          State':=3 /\ SecureChannel':=new() /\ secret(Cpic',sec_4,{}) /\ secret(Smeasure',sec_3,{}) /\
ND(Tpi.{Smeasure.TIDpi.Cpic}_SecureChannel')
end role
```

Figure 3.10. Role of D2D device Dpi in PUP phase.

```
role role_CS(CS:agent,Dpi:agent,Dpk:agent,TIDpk:text,Rpi:text,MACpic:text,Tpi:text,MACcpi:text,
Rcpi:text,Tc:text,Smeasure:text,TIDpi:text,Cpic:text,SND,RCV:channel(dy))
played_by CS
def=
        local
                State:nat,SecureChannel:symmetric_key
        init
                State := 0
        transition
        3.  State=0 /\ RCV(TIDpi.Rpi.MACpic.Tpi) =|>
          State':=1 /\ SND(MACcpi.Rcpi.Tc)
        4.  State=1 /\ RCV(Tpi.{Smeasure.TIDpi.Cpic}_SecureChannel') =|>
          State':=2 /\ secret(Cpic',sec_4,{}) /\ secret(Smeasure',sec_3,{})
end role
```

Figure 3.11. Role of the cloud server in PUP phase.

Two of the AVISPA’s four security evaluation backends, namely On-the-Fly-Model-checker (OFMC) [19] and Constraint Logic-Based Attack Searcher (CL-AtSe) [20] were used in the validation of out protocol. Figure 3.12 shows the results of the OFMC analysis in the PUP phase that prove the safety of the protocol proposed in this chapter.

```
% 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: 0.00s
  searchTime: 0.04s
  visitedNodes: 7 nodes
  depth: 6 plies
```

Figure 3.12. OFMC analysis result.

Figure 3.13 shows the analysis of the PUP phase in CL-AtSe backend and its respective results. The protocol proposed in this chapter was considered safe.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 12 states
  Reachable  : 8 states
  Translation: 0.05 seconds
  Computation: 0.00 seconds
```

Figure 3.13. CL-AtSe analysis result for PUP phase.

3.7. Conclusions

The application of e-health/m-health to the monitoring, diagnosis and treatment of patients speeds up the provision of medical services. In many cases, the patient does not need to leave his/her home for a doctor’s appointment, which facilitates the access to medical advice for patients with limited mobility, the elderly or patients located in difficult access areas.

The protocols analyzed showed interest in the development of efficient and safe e-health/m-health/TMIS systems for protecting patient's data and their respective identities. The protocol proposed in this chapter has proven suitable showed itself to be suitable to e-health/m-health/TMIS and overperformed those of Chiou et al. [2] and Mohit et al. [1]. The protocol of Chiou et al. [2] does not control the access to patients' mobile devices for avoiding their system's exposure to intruders, if the device is lost or stolen, which is a problem with a simple solution. The protocol designed by Mohit et al. [1] fails to avoid the Denial of Service (DoS) attack. Neither of the protocols supports D2D communication.

Furthermore, reductions in computational and communication costs are reinforced by the use of symmetric cryptography. Asymmetric cryptography demands more resource consumption due to the execution of more complex operations such as elliptic curves [22], and some common misconceptions (e.g., asymmetric cryptography is safer than symmetric cryptography) have been reported. Regarding cryptoanalysis, the length of the key and the computational work for the breakage of a cipher are essential for security evaluation. Symmetric cryptography is suitable to situations that require costs reduction (e.g., resource-constrained devices used for m-health). Performance and security analyses confirmed the protocol proposed in this chapter can be reduce resource consumption in comparison with other solutions that use asymmetric cryptography, with no impact on the system's security through the use of symmetric cryptography.

Future studies will include, storage cost analysis and comparisons with related work and development of other mutual authentication protocols based on asymmetric cryptography for cloud-based e-health systems that accomplish more security objectives, (e.g., the objectives presented by Liu et al. [23], with reduced resource consumption).

The development of authentication and authorization protocols that consider CPS (Cyber Physical Systems) ([24], [25], [26]), as well as security evaluation based on integrated systems of ambient-assisted living (AAL) and e-health (as in Rghioui et al. [27]) will also be considered, and the influence of the mobility on the authentication of D2D communications ([28] [29] [30]) will be explored.

3.8. References

[1] Mohit, P., Amin, R., Karati, A., Biswas, G. P., Khan, M. K., "A standard mutual authentication protocol for cloud computing based health care system.", *Journal of medical systems*, v.41, n. 4, pp. 50, 2017.

[2] Chiou, S., Ying, Z. and Liu, J., "Improvement of a privacy authentication scheme based on cloud for medical environment", *Journal of medical systems*, v. 40, n. 4, pp.101, 2016.

[3] Jiang, Q., Lian, X., Yang, Chao., Ma, J., Tian, Y. and Yang, Y., "A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth", *Journal of medical systems*, v. 40, n. 11, pp. 4650–4666, 2016.

[4] Li, X., Niu, J., Karuppiah, M., Kumari, S. and Wu, F., (2016) "Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications.", *Journal of medical systems*, v.40, n.12, pp.268, 2016.

[5] Jiang, Q., Khan, K. M., Lu, X., Ma, J. and He, D., "A privacy preserving three-factor authentication protocol for e-health clouds.", *The Journal of Supercomputing*, v. 72, n. 10, pp. 3826–3849, 2016.

[6] Amin, R., Islam, S. K., Biswas, G. P., Giri, D., Khan, K. M. and Kumar, N., "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments.", *Security and Communication Networks*, v. 9, n. 17, pp. 4650–4666, 2016.

[7] Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H. and Tang, Y., "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks.", *Journal of Network and Computer Applications*, v. 106, pp. 117-123, 2018.

[8] Gunes, T. T., and Afifi, H., "Hybrid model for lte network-assisted d2d communications" In *International Conference on Ad-Hoc Networks and Wireless,* pp.100-113, 2014.

[9] Zhang, A., Wang, L., Ye, X., & Lin, X. (2017). "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems", *IEEE Transactions on Information Forensics and Security*, v.12, n.3, pp.662-675.

[10] Kar, Udit Narayana, and Debarshi Kumar Sanyal. "An overview of device-to-device communication in cellular networks." *ICT express,* 2017.

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

[12] 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 June /2019.

[13] 3GPP TS 36.843 “Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects.” Mar. 2014. https://www.3gpp.org/ftp/Specs/archive/36_series/36.843/, Visited on June /2019.

[14] M., Wang, and Z., Yan, “Privacy-preserving authentication and key agreement protocols for D2D group communications”, *IEEE Transactions on Industrial Informatics*, vol.14, no.8, pp.3637-3647, 2018.

[15] R. H., Hsu, J., Lee, T. Q., 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.

[16] Kumar, A., and Om, H., “Handover Authentication Scheme for Device-to-Device Outband Communication in 5G-WLAN Next Generation Heterogeneous Networks”, *Arabian Journal for Science and Engineering,* n.*43, v.*12, pp.7961-7977, 2018.

[17] He, D., Chan, S., and Guizani, M., “Handover authentication for mobile networks: security and efficiency aspects”, *IEEE Network,* n.*29,* v.3, pp.96-103, 2015.

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

[19] D., Basin, S., Moedersheim, and L., Vigano, “OFMC: A symbolic model checker for security protocols”, *International Journal of Information Security,* v.4, n.3, pp. 181-208, 2005.

[20] M., Turuani, “The CL-Atse protocol analyser”, *International Conference on Rewriting Techniques and Applications,* pp. 277-286, Springer, Berlin, Heidelberg, 2006.

[21] Kang, M., Park, E., Cho, B. H., and Lee, K. S. “Recent patient health monitoring platforms incorporating internet of things-enabled smart devices.” *International neurourology journal,* n.22, suppl 2, p.S76, 2018.

[22] Stallings, William. Cryptography and network security: principles and practice. sixth ed. Boston: Pearson, 2014.

[23] Liu, K., Shen, W., Cheng, Y., Cai, L. X., Li, Q., Zhou, S., and Niu, Z. “Security Analysis of Mobile Device-to-Device Network Applications.” *IEEE Internet of Things Journal,* v.*6,* pp.2922-2932, 2018.

[24] Kassa, T., Nurrie, T., Feleke, E., Biadgligne, Y. Applications and Security Challenge of Cyber-Physical Systems: Survey. https://www.researchgate.net/publication/331223221_Cyber_Physical_System_Applications_and_Security_issues-_Survey , Visited on November 2019.

[25] Gawanmeh, A., and Alomari, A. “Taxonomy analysis of security aspects in cyber physical systems applications.” *IEEE International Conference on Communications Workshops (ICC Workshops),* pp.1-6, 2018.

[26] Yassine, M., Shojafar, M., Haqiq, A., and Darwish, A., “Cybersecurity and Privacy in Cyber Physical Systems”, *CRC Press Taylor & Francis*, 2019.

[27] Rghioui, A., Sendra, S., Lloret, J., Oumnad, A., “Internet of things for measuring human activities in ambient assisted living and e-health”, In *Network Protocols and Algorithms*, v.8, n. 3, pp.15-28, 2016.

[28]Omri, A., and Mazen O. H. "A distance-based mode selection scheme for D2D-enabled networks with mobility." *IEEE Transactions on Wireless Communications*, v. 17, pp.4326-4340.

[29] Waqas, M., Niu, Y., Li, Y., Ahmed, M., Jin, D., Chen, S., and Han, Z. “Mobility-Aware Device-to-Device Communications: Principles, Practice and Challenges.” *IEEE Communications Surveys & Tutorials*, 2019.

[30] Wang, R., Zhang, J., Song, S. H., and Letaief, K. B. “Exploiting mobility in cache-assisted D2D networks: Performance analysis and optimization.” *IEEE Transactions on Wireless Communications*, n.17, issue 8, pp.5592-5605, 2018.

Chapter 4

GROUP AUTHENTICATION PROTOCOL BASED ON AGGREGATED SIGNATURES FOR D2D COMMUNICATION

Abstract. Device-to-device (D2D) communication is one of the most promising technologies of new mobile communication networks (5G), and essential for the Communicating Things Networks (CTNs) paradigm. Its application scope has been widened and billions of devices are expected to be communicating in the next years. Moreover, goals, such as end-to-end security and reductions in computational costs required by resource-constrained devices, must be accomplished for its full implementation. The Third Generation Partnership Project (3GPP) standardized authentication and key agreement (AKA) protocols are not suitable for D2D due to differences in the architecture and communication scenario. They may cause several security issues and excessive computational and transmission overhead, since they require individual executions for each pair of devices. The development of AKA protocols adapted to D2D group communication is still in its initial steps. This article introduces a new authentication and key agreement protocol for D2D groups of devices based on asymmetric cryptography and aggregated signatures. The scheme focuses on robust security and reductions in computational and communication costs. It was compared to other group AKA protocols and yielded better results regarding security and overhead reduction. A formal validation conducted by Automated Validation of Internet Security Protocols and Applications tool (AVISPA) proved its robustness.

Keywords: Device-to-device communication, group authentication, proximity service, security, CTN.

4.1 Introduction

The 5th generation of mobile networks (5G), accompanied by the Internet of Things (IoT), is expected to emerge in the next few years and, therefore, must be adapted to meet the requirements of the IoT regarding capacity, transmission rates, and security. The exponential growth in the number of mobile devices has caused network issues, as congestion, use of spectrum and user's security, which will become more serious, since billions of devices will be connected.

According to Zhang et al. [1], Device-to-device communication (D2D) might be a data offloading solution to 5G, enhancing the spectrum efficiency through the use of resources, user's throughput and extension of the battery lifetime. It aims to provide direct communication among physically close devices without the intermediation of fixed network infrastructures, as base stations in the 3GPP network model, as in traditional cellular systems.

However, security is an important item to be considered in D2D communication. Wang et al. [2] recalled security solutions currently used in 3GPP Long Term Evolution (LTE) [3] provide only mutual authentication and key agreement among devices and core network, which exposes the new application scenario to several vulnerabilities. The proper functioning of some applications can be sensitive to security objectives, as privacy, anonymity, and confidentiality.

Due to the large number of devices that has emerged and the extra number of negotiation messages necessary for the authentication and key agreement (AKA) of D2D communication devices, groups of devices should be authenticated towards simplifying the process and reducing the consumption of resources. The new authentication protocols designed to groups of devices must guarantee the accomplishment of such security objectives and resist attacks, as replay, man-in-the-middle, and personification.

An important need related to D2D communications involves the discovery of devices. The mechanism provided by 3GPP Proximity Services (ProSe) [4] enables devices to discover each other based on proximity. Then, close devices with common interests can form a group to facilitate their mutual authentication procedures, after a discovery process. Therefore, applications that comprise large quantities of devices trying to authenticate simultaneously with each other or with the core network can have more efficient authentication when compared to single authentication. Such applications are m-health, which forecasts large amounts of people being monitored and sending their health information to doctors and health centers and agriculture, which estimates sensors spread in crops to monitor humidity, temperature and sunlight incidence.

The advancement of Communicating Things Networks (CTNs) predicts billions of devices connected, executing thousands of new applications, many of which will be classified as a fundamental part of society's daily life, since they would provide solutions for health, smart cities, vehicles and smart metering of resources consumption, such as electricity and water, for example. The widespread adoption of CTNs will probably augment the complexity of network management and require security solutions that are more robust and sophisticated in order to contain the growth of threats. The security of the CTNs systems has not been well studied yet and needs special attention. Even a single attack might harm the integrity of the system if considered the magnificence and complexity of CTNs.

D2D communication is suitable to be used in the development of the CTNs since it can provide direct connection among devices, without the intervention of traditional network infrastructure. Consequently, due to the number of devices, D2D can help to reduce the usage of the spectrum, avoiding congestion and collision. D2D communication devices are commonly resource-constrained and have low computation power, small storage and short battery lifetime. Consequently, AKA protocols designed for D2D must not overload them, be computationally light-weighted and avoid excessive transmission overhead for overcoming such restrictions**.**

4.1.1 Main Contributions

New solutions to current and future problems to be faced by D2D communication must be designed. The main contributions of this research involve:

1. the design of a new mutual authentication and key agreement protocol for D2D group of devices, based on asymmetric cryptography and assisted by the 3GPP infrastructure that can be used in situations with large amounts of devices such as applications triggered by device proximity, as sensoring in agriculture, advertisements and smart communication between vehicles;
2. the use of aggregated signatures for the authentication of groups of D2D devices, since such a mechanism provides mutual authentication to all devices in a group;
3. an authentication protocol for D2D communications which provides security properties as confidentiality, privacy, anonymity and protection against several attacks, including DoS, man-in-the-middle and impersonation;
4. evaluation of computational, communication and energy costs, in a comparative manner with other authentication protocols;
5. semi-automated formal validation of the proposed protocol.

4.1.2 Organization of the Chapter

The remainder of this chapter is structured as follows: Section 4.2 addresses some related works; Section 4.3 introduces the protocol; Sections 4.4 and 4.5 report on a security analysis and a performance analysis, respectively, and comparisons among the proposed protocol and other protocols; Section 4.6 is devoted to the semi-automated formal validation of the protocol conducted by AVISPA tool; finally, Section 4.7 provides the main conclusions.

4.2. Related Work

This section provides an overview of some group authentication protocols for D2D communication designed for enhancing security and reducing resources consumption.

Among the studies on the security challenges that might be faced by D2D communication in comparison to conventional connections are those conducted by Wang and Yan [8], Zhang and Lin [9] and Haus et al. [10]. The authors reported reviews and surveys that summarize the challenges, requirements, and features related to security and privacy and identified problems that have motivated research into D2D communication.

According to Wang and Yan [8], the connection of close devices causes some vulnerabilities in security due to their direct connection, new transmission structure, mobility, handover, and roaming of devices and issues caused by loss of privacy in some social applications. Zhang and Lin [9] observed D2D communication faced threats, as eavesdropping, jamming and impersonation. Nonetheless, some preliminary protection can be obtained through the authentication among devices.

In this sense, some relevant authentication protocols for D2D communication are here described, for future comparison.

Wang and Yan [5] developed two group authentication protocols for D2D communication coined Privacy-Preserving Authentication and Key Agreement - Hash-based Message Authentication Code (PPAKA-HMAC) and Privacy-Preserving Authentication and Key Agreement - Identity-Based Signature (PPAKA-IBS). The first combines group key agreement with HMAC and uses pseudonyms, instead of permanent identities, to preserve the anonymity of devices. At the end of the authentication procedure, all devices in the group generate a common session key. Although a single session key facilitates the interactivity of devices, it compromises their confidentiality and privacy.

The other protocol is based on IBS and uses pseudonyms, instead of permanent identities. Unlike PPAKA-HMAC, it promises to be resistant against insiders' attacks. However, PPAKA-IBS shows the same session key generation calculations used in PPAKA-HMAC and generates the same session key for all devices in the group, which jeopardizes security objectives, as confidentiality and privacy.

Hsu and Lee [6] designed a group authentication protocol coined Group Anonymity for D2D Communication (GD2C) with core network (CN) assistance (CN-GD2C) that is assisted by the 3GPP infrastructure and based on indistinguishability under adaptively chosen ciphertext attack (IND-CCA), symmetric and asymmetric cryptography and Diffie-Hellman key exchange (DHKE) [11]. The indistinguishability (IND-CCA) provides the avoidance of injection attacks, because an attacker cannot distinguish the ciphertext from common messages exchanged in the insecure channel.

Hsu et al. [7] proposed a network-assisted group authentication protocol for D2D communication coined Group Anonymous and Accountable D2D Communication in Mobile Networks (GRAAD) that uses Identity-Based Encryption (IBE), Diffie-Hellman key exchange, symmetric encryption, and hash functions. However, similarly to the protocol proposed by [8], it conducts an authentication session for each pair of devices and causes high communication and computational overheads.

The literature reports other group authentication protocols for D2D. For example, Abd-Elrahman et al. [12] proposed a group authentication scheme that uses ID-based cryptography (IBE and ECC integrated), and Kwon et al. [13] designed a scheme based on Bluetooth and Wifi Direct that enables users to share secret keys by exploiting ciphertext-policy attribute-based encryption (CP-ABE). Additionally, Tayade and Vijayakumar [14] proposed a Secure Data Sharing Strategy (SeDS) for secure communication between evolved NodeB and gateway (GW) in LTE-A network. The protocol is based on digital signatures and symmetric encryption and ECC and has the objective of achieving security and availability parameter for D2D communication. Table 4.1 shows a general comparison among the protocol proposed in this chapter and those previously described.

Table 4.1. General comparison among protocols.

	System Model	Group Authentication	Group Management	Leader Election	Type of Cryptography	ECC	DHKE
PPAKA-HMAC [5]	Assisted by SN	Yes	Yes	No	Asymmetric	No	Yes
PPAKA-IBS [5]	Assisted by SN	Yes	Yes	No	Asymmetric (IBS)	Yes	Yes
CN-GD2C [6]	Assisted by eNB, MME, HSS/ AuC	Yes	No	No	Symmetric/ Asymmetric (IND-CCA)	No	Yes
GRAAD [7]	Assisted by eNB, ProSe function, HSS/AuC	Yes	No	No	Symmetric/ Asymmetric (IBE)	No	Yes
Abd-Elrahman et al. [12]	Assisted by SN	Yes	Yes	No	Asymmetric (IBE)	Yes	Yes
Kwon et al. [13]	Not assisted by SN	Yes	No	No	Symmetric	No	No
Tayade and Vijayakumar [14]	Assisted by eNB/GW	No	No	No	Symmetric/ Asymmetric (ECC)	Yes	Yes
Proposed Protocol	Assisted by HSS/AuC	Yes	Yes	Yes	Asymmetric (Aggregated Signatures)	Yes	Yes

4.3. Proposed Protocol

In this section, we propose a new D2D group authentication protocol for 3GPP 5G networks and the procedures of entering/leaving devices. The protocol is composed of two phases, namely registration/group organization and mutual authentication, and has been adapted to D2D environment and its requisites. It considers a device discovery, a group leader election, management of devices entering and leaving the group and the mutual authentication and key agreement of those devices. All the components described are obtained reducing communication, computational and energy costs, and strengthening the security of devices in comparison to other D2D group authentication protocols, as [5], [6] and [7].

4.3.1 Basic Assumption

First, some basic assumptions must be defined. The system architecture, shown in Figure 4.1, is a D2D communication environment where devices can communicate directly with each other without the intermediation of a network infrastructure. However, the devices must complete an AKA procedure prior to the establishment of a data exchange link between any two devices. In the scenario adopted, such a procedure is assisted by the 3GPP infrastructure and performed in groups of devices. The 3GPP infrastructure is composed of an Evolved NodeB (eNB) in the Evolved Universal Terrestrial Radio Access Network (E-

UTRAN) and the Home Subscriber Server/Authentication Center (HSS/AuC) inside the Evolved Packet Core (EPC).

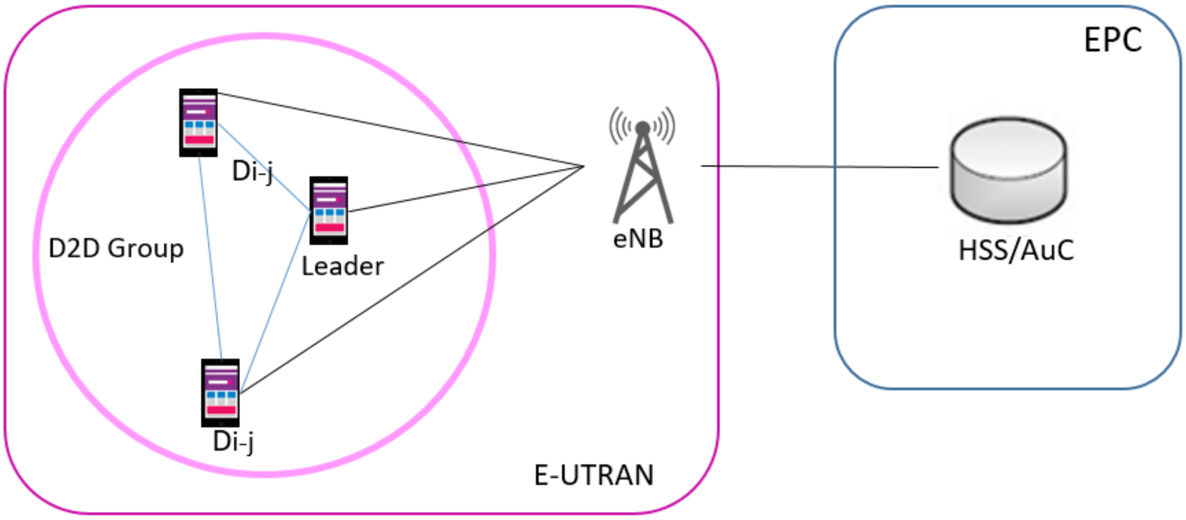


Figure 4.1. Architecture of the system.

The protocol is based on asymmetric cryptography and Diffie-Hellman key exchange (DHKE) [11], because it provides a way of sharing a mutual key among two entities. Moreover, it is partially based on the aggregation of ID-based signatures (explored in the context of M2M by [15] and [29], and here partially adapted for D2D communication), because it is a solution that has proven to reduce communications costs due to the reduction in the number of messages exchanged, which is provided by the aggregation of signatures. The system is based on TDMA, in which each base station allocates time slots to D2D-capable devices. The synchronization is executed by the eNodeB.

A group leader is elected among the devices to intermediate the messages exchanged in the authentication procedure and enable an almost simultaneous authentication. The leader is responsible of receiving authentication parameters from ordinary devices and aggregating them before the group authentication. A D2D discovery is performed according to the recommendations of ProSe [4]. The group management is based on the binary tree described by [16], because it provides an organization of entities that facilitates the management of devices joining or leaving the group. The configuration of parameters is made offline, taking into consideration the defined architecture. Table 4.2 shows standardized notations, where $D_{i\text{-}j}$ represents device *i* of group *j*, and the parameters of the system.

Table 4.2. System parameters.

Symbol	Definition
p	a k bit prime
Zp	a prime field of order p
G1, G2	two elliptical curve groups of order p
X	random number x ϵ Zp
PK	system public key
i	index of devices
j	index of groups
Di-j	device i of group j
$\text{Secret}_{\text{Node\_x}}$	the secret of node *x* in the binary tree
$\text{SEC}_{\text{i-j}}$	the secret between device i of group j and HSS/AuC
GID_{j}	group identity
GK_{j}	group key
$\text{ID}_{\text{Di-j}}$	permanent identity of device i of group j
$\text{TID}_{\text{Di-j}}$	temporary identity of device i of group j
$\text{QID}_{\text{Di-j}}$	public key of $\text{D}_{\text{i-j}}$
R_{G}	random number of the group key
$\text{S}_{\text{Di-j}}$	private key of device $\text{D}_{\text{i-j}}$
LAI_{j}	location area identity
$\sigma\ _{\text{Di-j}}$	signature of device i of group j
$\text{T}_{\text{Di-j}}$	timestamp of device i of group j
$\text{U}_{\text{Di-j}}$	first signature element of Di-j
$\text{V}_{\text{Di-j}}$	second signature element of Di-j
V_{j}	aggregation of all the signature elements of group j
e(-,-)	bilinear pairing function
$\text{SK}_{\text{Di-k}}$	session key between devices i and k
$\text{Verif}_{\text{SKDi-j}}$	verification value of the session keys of Di-j
$\text{T}_{\text{Di-j}}$	a timestamp of Di-j
$\text{T}_{\text{2Di-j}}$	a timestamp of Di-j
n	number of devices in the group
H_1	identity hash function - H_1: {0,1}* $\longrightarrow$ Z_p
H_2	private key hash function - H_2: {0,1}* $\longrightarrow$ G_1
H_3	secure hash function - H_3: {0,1}* $\longrightarrow$ Z_p
H_4	key generation hash function - H_4: {0,1}* x G_1 $\longrightarrow$ G_2
$\longrightarrow$	secure channel
$\dashrightarrow$	insecure channel
$\textcolor{green}{\longrightarrow}$	D2D message exchange
$\textcolor{blue}{\longrightarrow}$	device/Network message
$\textcolor{red}{\longrightarrow}$	broadcast message

4.3.2 ProSe D2D Discovery

Here we briefly explain 3GPP device discovery based on the technical specification TS 23303 [4], which provides standardizations for the detection and identification of nearby devices through E-UTRAN or WLAN direct radio signals with two models of operation:

- **Model A ("I am here") –** Devices broadcast information to enable other devices nearby to discover their existence; each device that aims at establishing connection with them shows its interest. They evaluate such interested devices, read and process them for establishing connections;

- **Model B ("who is there?" / "are you there") –** Devices ask other devices nearby for information on their interest, and they respond with the information requested. Connection is then established with interested devices.

Our scheme adopts Model A of device discovery, in which devices are expected to announce pre-defined information, as in Sun et al. [17], to be used in their identification, monitoring and processing. They broadcast information and temporary identities to nearby devices, which can show interest in establishing communication with them. After receiving information from other devices, the interested ones try to establish connection with the device that sent the information by performing an authentication and key agreement procedure. For a more detailed description see 3GPP TS 23.303 [4].

4.3.3 Registration/Group Organization Phase

The registration and group organization phase is performed over a secure channel. It aims at the registration of devices on the 3GPP's core network, D2D discovery, group organization and distribution of some authentication parameters.

The phase starts with each device sending to HSS/AuC its real identity $ID_{Di\text{-}j}$, provided by the manufacturer. HSS/AuC generates temporary identities by choosing a random number $R_{IDi\text{-}j} \epsilon Z_p$ and calculating:

$$TID_{Di\text{-}j} = H_1 (ID_{Di\text{-}j} \| R_{IDi\text{-}j} * P) \qquad (4.1)$$

The D2D discovery is based on proximity and follows the process described in 3GPP ProSe [4]. Devices discover each other by broadcasting their $TID_{Di\text{-}j.}$ The interested devices show their intention of establishing connection with them by sending their own temporary identity. Then, after discovery, the devices with common interests form a group.

After the group formation, a group leader is elected according to characteristics, as computational power, storage, battery life, and associativity (see Hussain et al. [18] and Gharehchopogh and Arjang [19]). The devices are then organized in the binary tree structure described by Choi et al. [16] and shown in Figure 4.2.

Each node in the binary tree has a secret ($\text{Secret}_{\text{Node\_x}}$) based on the secret values of its parents. Two hash functions (HR and HL) are defined for the calculation of their secrets. HR is used by nodes located on the right side of their parent and HL is employed for the node on the left side of their parent. For example, in Figure 4.2, Node_3's secret is determined by hash function HL of the secret of its parent, i.e., Node_1.

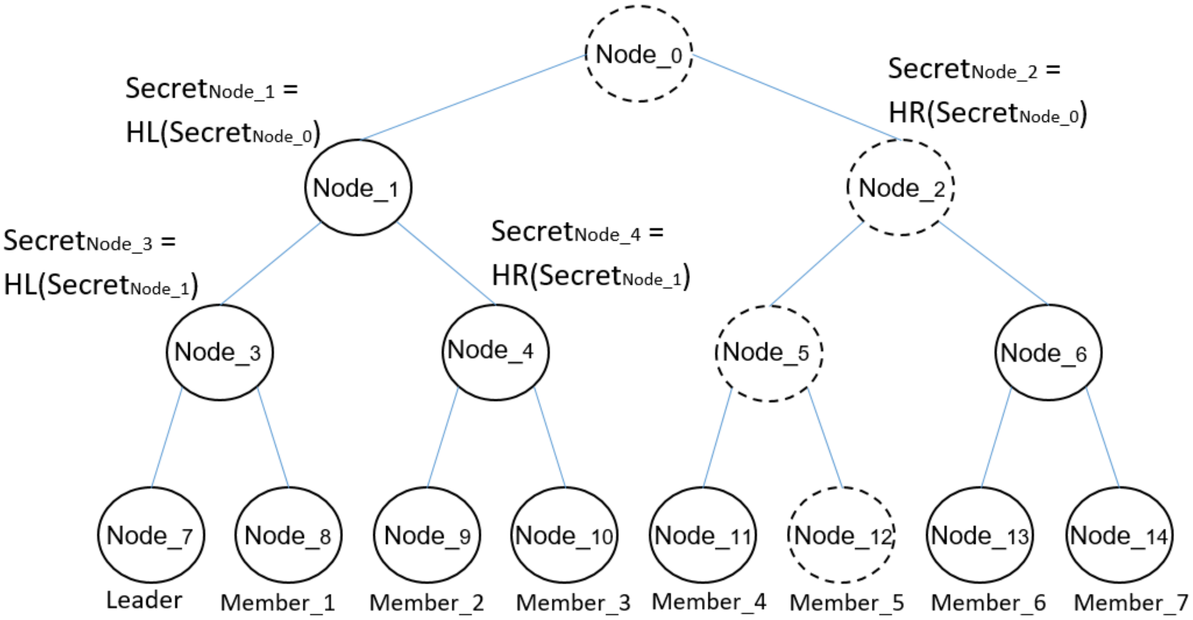


Figure 4.2. Binary tree group management.

Since no node can know its own secret and the secrets in the path to the main node Node_0, each device knows the remaining secrets on the tree, which avoids the access to secrets by unauthorized devices. The secrets known by each device will be used in the session key calculation. In Figure 4.2, Member_5 is placed on a doted node, meaning it cannot know the secret of its own node. The other nodes in the path to the main node, i.e., Node_0, are also doted, meaning Member_5 cannot know their secrets. HSS/AuC knows all node secrets.

Leader election and group organization are performed offline. After the group organization, the group leader requests the assistance of the 3GPP LTE network. HSS/AuC chooses a k bit prime p, two elliptical curve groups G_1 and G_2 of order p and a generator point P in G_1 and selects four hash functions of a finite domain (SHA-256), $H_1(.)$, $H_2(.)$, $H_3(.)$ and $H_4(.)$.The characteristics (domain,…) of hash functions are summarily presented in Table 4.2.

The group leader then sets the system master key by selecting a random number x ϵ Z_p, computes the system public key PK = x*P, generates a secret key SEC_{i-j} (where ***i*** is the number of the device in the group and ***j*** is the group number of which the device is part) to be shared with each device and calculates a group identity and a group key by choosing a random number $R_G \in Z_p$:

$$GID_j = H_1 (R_G) \tag{4.2}$$

$$G_{Kj} = H_4 (SEC_{1-j} \oplus SEC_{2-j} \oplus ... \oplus SEC_{i-i} \parallel R_G) \tag{4.3}$$

It then searches the Location Area Identity (LAI_j) of eNB that covers the devices, for preventing devices from being deceived by intruder eNBs and calculates private and public keys for each of them:

$$S_{Di-j} = x \times QID_{Di-j} \tag{4.4}$$

$$QID_{Di-j} = H_2 (TID_{Di-j} \parallel LAI_j \parallel GID_j) \tag{4.5}$$

Finally, HSS/AuC sends TID_{Di-j}, QID_{Di-j}, S_{Di-j}, GID_j, GK_j, SEC_{Di-j}, and LAI_j to the devices over a secure channel, chooses a bilinear pairing function and publishes the system's parameters ($P,PK,Z_p,G_1,G_2,e(-,-),H_1,H_2,H_3,H_4$). Figure 4.3 shows the messages exchanged in this phase.

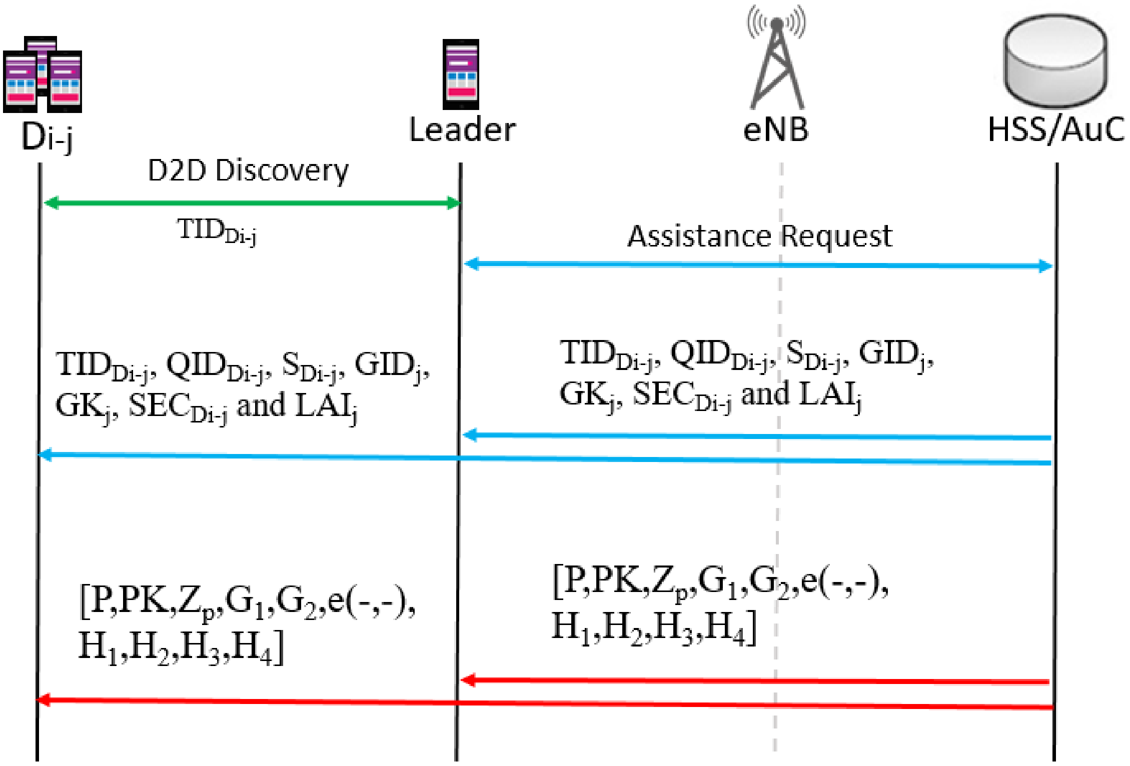

Di-j
Leader
eNB
HSS/AuC
D2D Discovery
TIDDi-j
Assistance Request
TIDDi-j, QIDDi-j, SDi-j, GIDj, GKj, SECDi-j and LAIj
TIDDi-j, QIDDi-j, SDi-j, GIDj, GKj, SECDi-j and LAIj
[P,PK,Zp,G1,G2,e(-,-), H1,H2,H3,H4]
[P,PK,Zp,G1,G2,e(-,-), H1,H2,H3,H4]

Figure 4.3. Registration/group organization phase.

4.3.4 Mutual Authentication

This phase is performed over an insecure channel. The authentication and key agreement procedure is conducted among D2D communication devices and a different session key is established for each pair of devices (each connection).

First, each device calculates its signature by choosing a random number $R_{Di\text{-}j} \in Z_p$, similarly to the signature generation proposed by [15]:

$$U_{Di\text{-}j} = R_{Di\text{-}j} * SEC_{Di\text{-}j} * P \tag{4.6}$$

$$h_{Di\text{-}j} = H_3\,(TID_{Di\text{-}j} \,\|\, GID_j \,\|\, T_{Di\text{-}j} \,\|\, U_{Di\text{-}j} \,\|\, LAI_j) \tag{4.7}$$

$$V_{Di\text{-}j} = S_{Di\text{-}j} * TID_{Di\text{-}j} + h_{Di\text{-}j} * SEC_{Di\text{-}j} * R_{Di\text{-}j} * PK \tag{4.8}$$

The signature is:

$$\sigma_{Di\text{-}j} = (V_{Di\text{-}j}, U_{Di\text{-}j}) \tag{4.9}$$

Then, each device sets a timestamp $T_{Di\text{-}j}$ and sends to the group leader the following parameters: $(TID_{Di\text{-}j},\sigma_{Di\text{-}j},T_{Di\text{-}j})$. After receiving the signatures, the leader performs the signature aggregation procedure, based on the signature aggregation proposed by [15]:

$$V_j = \sum_{i=1}^{n} V_{Di\text{-}j} \tag{4.10}$$

$$\sigma_j = (U_{D1\text{-}j}, U_{D2\text{-}j}, \ldots, U_{Di\text{-}j}, V_j) \tag{4.11}$$

The leader sets a timestamp T_j and broadcasts the subsequent parameters to the devices and HSS/AuC: ($TID_{D1\text{-}j}$, $TID_{D2\text{-}j}$…, $T_{D1\text{-}j}$,$T_{D2\text{-}j}$,…, σ_j, T_j). After receiving the aggregated signature from the leader, each device performs the mutual authentication by first checking if the timestamps are valid. If so, the devices calculate the public key $QID_{Di\text{-}j}$ and $h_{Di\text{-}j}$ of all the other devices in the group for the bilinear pairing verification:

$$h_{\text{Di-j}} = H_3\,(TID_{\text{Di-j}} \parallel GID_j \parallel T_{\text{Di-j}} \parallel U_{\text{Di-j}} \parallel LAI_j) \tag{4.12}$$

$$QID_{\text{Di-j}} = H_2\,(TID_{\text{Di-j}} \parallel LAI_j \parallel GID_j) \tag{4.13}$$

$$U_{\text{Di-j}} = R_{\text{Di-j}} * SEC_{\text{Di-j}} * P \tag{4.14}$$

$$h_{\text{Di-j}} = H_3\,(TID_{\text{Di-j}} \parallel GID_j \parallel T_{\text{Di-j}} \parallel U_{\text{Di-j}} \parallel LAI_j) \tag{4.15}$$

$$V_{\text{Di-j}} = S_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * PK \tag{4.16}$$

Below is the bilinear pairing operation:

$$e(V_j, P) \equiv e[\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * U_{\text{Di-j}}),PK] \tag{4.17}$$

$$\begin{aligned} & e(\sum_{i=1}^{n}(S_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * PK), P) \equiv \\ & e[\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * R_{\text{Di-j}} * SEC_{\text{Di-j}} * P), PK] \end{aligned} \tag{4.18}$$

$$\begin{aligned} & e(\sum_{i=1}^{n}(x * QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * x * P), P) \equiv \\ & e[\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * P), PK] \end{aligned} \tag{4.19}$$

$$\begin{aligned} & e(\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * P), x * P) \equiv \\ & e[\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * P), PK] \end{aligned} \tag{4.20}$$

$$\begin{aligned} & e(\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * P), PK) \equiv \\ & e[\sum_{i=1}^{n}(QID_{\text{Di-j}} * TID_{\text{Di-j}} + h_{\text{Di-j}} * SEC_{\text{Di-j}} * R_{\text{Di-j}} * P), PK] \end{aligned} \tag{4.21}$$

If the verification succeeds, all devices are authenticated, and a session key can be generated. A different session key is required for each pair of devices towards avoiding insider's attacks and violation of confidentiality and privacy. Below is the description of the session key calculation.

$$SK_{\text{Di-k}} = H_4\,[(Secret_{\text{Node1-j}} \oplus Secret_{\text{Node2-j}} \oplus ... \oplus Secret_{\text{Nodei-j}}) * U_{\text{Di-j}} * U_{\text{Dk-j}}) \tag{4.22}$$

The session key is based on both the secrets of the nodes known by the two devices involved in DHKE method. Only the common secrets of the pair of devices are considered, and those not known by the devices are discarded. The use of nodes' secrets in the calculation of session keys prevents possible man-in-the-middle attacks that might occur to the DHKE.

Finally, we have developed a verification scheme to validate the authentication procedure. Each device calculates a verification value $Verif_{SKDi\text{-}j}$, composed of all its session keys, and sends it with a timestamp $T_{2Di\text{-}j}$ to HSS/AuC for the authentication approval by HSS/AuC:

$$Verif_{SKDi\text{-}j} = H_3\,(SK_{Di\text{-}1} \oplus SK_{Di\text{-}2} \oplus ... \oplus SK_{Di\text{-}k}) \tag{4.23}$$

After receiving $Verif_{SKDi\text{-}j,}$ HSS/AuC calculates all session keys, as previously described, using all $U_{Di\text{-}j}$ received and the nodes' secrets. It then generates parameter $Verif_{SKDi\text{-}j}$' of each device in the group and compares each $Verif_{SKDi\text{-}j}$ received with the $Verif_{SKDi\text{-}j}$' calculated. If $Verif_{SKDi\text{-}j}$= $Verif_{SKDi\text{-}j}$', HSS/AuC sends a message to all devices informing on the success of the authentication. If the message indicates a failure, the authentication procedure fails and is disregarded by the devices. Figure 4.4 illustrates the messages exchanged among the entities in the mutual authentication phase.

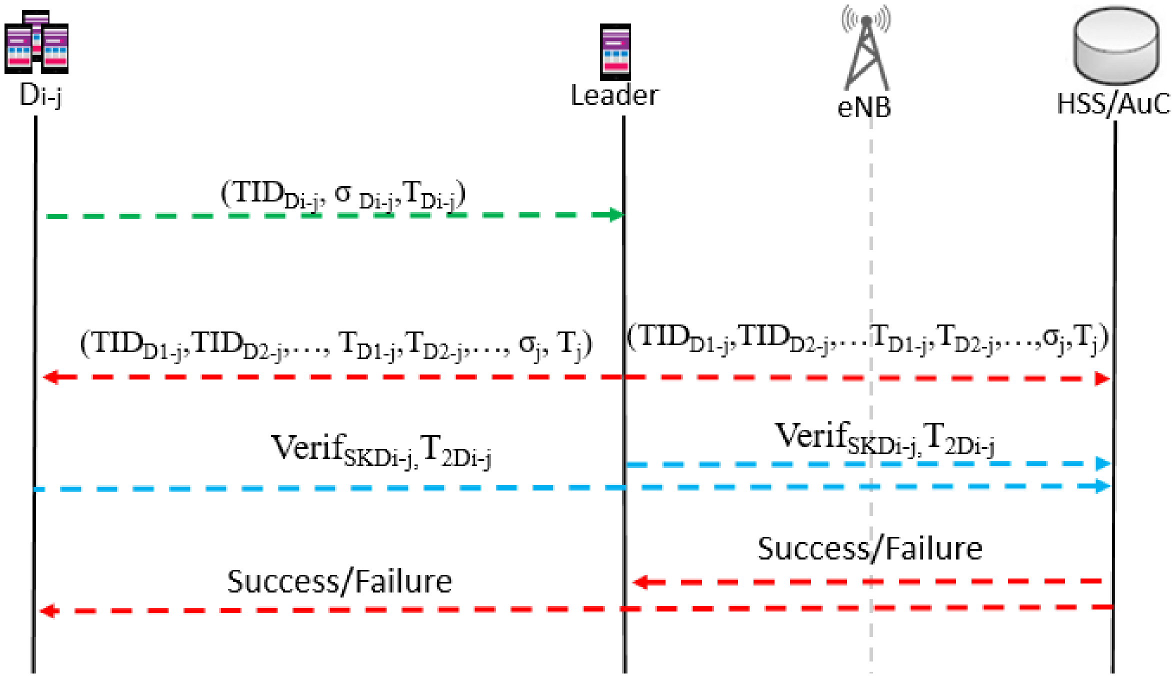

Di-j
Leader
eNB
HSS/AuC
$(TID_{Di\text{-}j}, \sigma_{Di\text{-}j}, T_{Di\text{-}j})$
$(TID_{D1\text{-}j}, TID_{D2\text{-}j}, \ldots, T_{D1\text{-}j}, T_{D2\text{-}j}, \ldots, \sigma_j, T_j)$
$(TID_{D1\text{-}j}, TID_{D2\text{-}j}, \ldots T_{D1\text{-}j}, T_{D2\text{-}j}, \ldots, \sigma_j, T_j)$
$Verif_{SKDi\text{-}j,} T_{2Di\text{-}j}$
$Verif_{SKDi\text{-}j,} T_{2Di\text{-}j}$
Success/Failure
Success/Failure

Figure 4.4. Mutual authentication phase.

4.3.5 Devices Entering and Leaving the Group

Two cases must be considered in the behavior of the devices, i.e., when they are leaving or entering a D2D group. Each case has different procedures, described as follows:

4.3.5.1 A device leaves the group

A device might leave the group if it has completed its tasks or left the specific coverage area and has shown battery exhaustion. If a device desires to leave the group, first it must send to the group leader a request containing its $TID_{Di-j.}$ After receiving the device's temporary identity, the leader forwards an exit request encrypted with the current group key GK_j to HSS/AuC:

$$Out_{Di-j} = GK_j\ (TID_{Di-j} \parallel GID_j) \tag{4.24}$$

HSS/AuC receives the request, decrypts the message and obtains the TID_{Di-j} of the device that is leaving. It then revokes GID_j and GK_j and all session keys linked to this device, which is disassociated from its leaf in the binary tree. Next, HSS/AuC chooses a new random number $R_G \in Z_p$ and calculates new GID_j and GK_j using secrets SEC_{Di-j} of the devices that remained in the group:

$$GID_j = H_1\ (R_{Gj}) \tag{4.25}$$

$$GK_j = H_4\ (SEC_{1-j} \oplus SEC_{2-j} \oplus ... \oplus SEC_{i-j} \parallel R_{Gj}) \tag{4.26}$$

It then encrypts the new parameters generated with each device secret:

$$NewG_{Di-j} = SEC_{Di-j}\ (TID_{Di-j} \parallel GID_j \parallel GK_j) \tag{4.27}$$

The devices receive the message, decrypt it and obtain the renewed parameters.

4.3.5.2 A device enters the group

If a device needs to enter the D2D group, it performs its registration with HSS/AuC and receives all parameters distributed in the registration/group authentication phase and a free leaf from the binary tree containing a $Secret_{Node\_x}$. A new group key and group identity containing the entering device leaf's secret are calculated. The newly generated parameters are encrypted with the old $GKold_{Di-j}$ and broadcasted to the group of devices. Below is their generation procedure:

$$GID_j = H_1\ (R_{Gj}) \tag{4.28}$$

$$GKnew_j = H_4\ (SEC_{1-j} \oplus SEC_{2-j} \oplus ... \oplus SEC_{i-j} \parallel R_{Gj}) \tag{4.29}$$

$$NewG_{Di-j} = GKold_{Di-j}\ (TID_{Di-j} \parallel GID_j \parallel GKnew_j) \tag{4.30}$$

Additionally, the new device performs an authentication procedure to obtain a session key with each device of the group. The procedure is the same as that presented in the mutual authentication section. However, it is performed among a group of devices and a single device. If the procedure is successful, a session key is generated.

4.4. Security Analysis

This section reports a security analysis and discussion comprehending the security objectives accomplished by the proposed protocol and resistance to several possible attacks.

Since Dolev-Yao model [20] is the attack (adversary) model adopted, messages can be composed and replayed by an adversary; however, they cannot be deciphered if the correct keys are not known.

4.4.1 Mutual Authentication and key agreement – The mutual authentication is accomplished by a bilinear pairing operation conducted by all devices in the group. After receiving signatures and other parameters from the group leader, each device performs the verification expressed by equation (17). If the verification succeeds, all devices in the group are authenticated, since a single mutual authentication procedure authenticates them.

4.4.2 Confidentiality – The confidentiality of the system is guaranteed by the session keys (SK_{Di-k}) generated at the end of the mutual authentication phase. Each session key is calculated by both entities involved in the process and is not sent over insecure channel. Then, any message containing user's data is encrypted with the respective session key established before it is sent over an insecure channel. Therefore, user's data is only available to entities which have the session key that can decrypt the respective message.

4.4.3 Anonymity – The anonymity of devices is guaranteed by the use of temporary identities $TID_{Di-j.}$ The messages exchanged over an insecure channel do not contain the permanent identity of the devices and only the temporary identity is exchanged over an insecure channel. The temporary identities are generated in each authentication session, therefore, an attacker cannot recognize a device by linking new and old messages to the same temporary identity, which guarantees the protection of the devices' anonymity.

4.4.4 Privacy - The privacy of the devices is acquired by the anonymity and confidentiality properties. Neither the real identity of a device (anonymity), nor details of the data exchanged (confidentiality) can be known. Therefore, the proposed protocol avoids the exposure of private information of the entities involved for agents not authorized.

4.4.5 Forward/Backward Secrecy – Both forward and backward secrecies are assured through the use of freshly generated group keys in each authentication session and when devices leave or enter the group. No device entering the group can access messages exchanged before its entry using the newest key. Similarly, no device that left the group can access the data exchanged using an old key.

4.4.6 Non-Repudiation – The use of temporary identities and secret key SEC_{i-j} for the calculation of signatures of devices in the mutual authentication phase assures the authorship of each signature. Only the device and HSS/AuC know the secret key SEC_{i-j} and the identities linked to it. Therefore, no device can deny the authorship of its signatures.

4.4.7 Session Key Security – The session keys are generated in each device, without being exchanged over an insecure channel. The secrets $Secret_{Node\_x}$ in common among a pair of devices involved in the key generation is used in the calculation of the session key. The group of secrets is unique for each pair of devices, which prevents the action of an attacker.

4.4.8 Resistance to replay attack – The replay attack is avoided by the use of newly generated timestamps T_j and random values R_{IDi-j}, in the calculation of temporary identities, signatures and session keys in each authentication session, as seen in the equations (1), (6), (7) and (8). The freshness of such values guarantees no attacker can use old parameters to forge authentication requests. Additionally, an attacker cannot forge any of those parameters because they depend on confidential information, such as the permanent identity ID_{Di-j} of entities and random values R_{Di-j}, which are not disclosed on a public channel. For example, if an attacker tries to use an old TID discovered, the system can detect the respective TID has been already used and finishes the authentication procedure banishing all intruders with old/invalid parameters from the authentication procedure.

4.4.9 Resistance to Denial of Service (DoS) attack – DoS attack is avoided through the verification of the timestamp’s validity before any more complex calculations are performed by the entity that has received a respective message. Consequently, if an attacker shoots a DoS attack attempt at a device, it is quickly avoided by the timestamp verification because the service will not be knocked down.

4.4.10 Resistance to man-in-the-middle-attack – The man-in-the-middle attack cannot be performed over the proposed protocol because the session key does not depend only on values exchanged on an insecure channel; the nodes’ secret $Secret_{Node\_x}$ of all nodes considered in the calculation of a specific session key must be accessed; however, these data are only available to HSS/AuC, which is a trusted entity.

4.4.11 Resistance to redirection attack – The redirection attack is avoided through the use of the LAI of the eNB where the devices are registered. An intruder eNB might try to redirect the infrastructure assistance through itself; however, it fails, since it does not have the valid LAI of the original eNB. The LAI is known only by HSS/AuC and valid devices and transmitted to the group of devices over a secure channel.

4.4.12Resistance to impersonation attack – No permanent identity ID_{Di-j} is disclosed over an insecure channel, only temporary identities TID_{D1-j}. Therefore, an attacker cannot impersonate the devices and generate valid signatures because they do not have access to their permanent identities. Despite the old temporary identity, no impersonation would be possible, since each temporary identity is renewed in each authentication procedure, as addressed in the resistance to replay attack topic. If an attacker accesses an old temporary identity and tries to use it in a newer authentication procedure, the core network knows the respective TID has already been used and immediately stops the authentication session. Consequently, no impersonation attack is possible in the proposed scenario.

4.4.13 Resistance to attackers inside the group – The protocol proposed in this chapter avoids attackers' actions inside the group through the use of different session keys for each pair of devices (or each connection) and different parameters, as identities, random values and signatures generated. Therefore, no malicious infiltrated device can impersonate other valid devices or use their session keys to access unauthorized data.

The protocols PPAKA-HMAC and PPAKA-IBS [5] fail to guarantee confidentiality and privacy because the session key generated at the end of each session is the same for all devices in the group, which enables them to obtain confidential information from the other devices in the group. Additionally, the proposals of [5], [6] and [7] are vulnerable to DoS (since they have no mechanism, such as timestamps or nonces, to verify complex calculations prior to the execution of the protocol) and redirection attacks (because they do not validate the eNB involved in the procedure by checking the identification of its location area - LAI).

The protocol proposed in this chapter accomplishes all the security objectives analyzed and is resistant to all the attacks considered in this D2D communication scenario. Consequently, it has shown the most robust regarding security. Table 4.3 shows a comparison among the protocol proposed in this chapter and those of [5], [6] and [7].

Table 4.3. Security objectives.

Security Objectives	PPAKA-HMAC [5]	PPAKA-IBS [5]	CN-GD2C [6]	GRAAD [7]	Proposed Protocol
Mutual Authentication	Yes	Yes	Yes	Yes	Yes
Confidentiality	No	No	Yes	Yes	Yes
Privacy	No	No	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes	Yes	Yes
Forward/ Backward Secrecy	Yes	Yes	Yes	Yes	Yes
Non-Repudiation	Yes	Yes	Yes	Yes	Yes
Session Key Security	Yes	Yes	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes
Resistance to DoS attack	No	No	No	No	Yes
Resistance to man-in-the-middle-attack	Yes	Yes	Yes	Yes	Yes
Resistance to redirection attack	No	No	No	No	Yes
Resistance to impersonation attack	Yes	Yes	Yes	Yes	Yes
Resistance to attacks inside the group	No	No	Yes	Yes	Yes

4.5. Performance Analysis

This section addresses a performance analysis of the protocol proposed in this chapter and comparisons among those proposed by [5], [6] and [7]. A computational cost analysis is provided followed by a communication cost analysis. The costs of a leader election and D2D discovery were not accounted, as in [16], [21] and [22], for leader election, and [5], [6] and [7] for D2D discovery.

4.5.1 Computational Cost

This section reports the results of a computational cost analysis conducted in a scenario with *n* devices in a group and a comparison among the protocol proposed in this chapter and others. Table 4.4 shows the costs and descriptions of each calculation in milliseconds (ms). The values adopted were those considered by [6] and [7] for a configuration based on Smartphone HTC One X with Android 4.1.1, 1.5 GHz Quad-core ARM Cortex-A9 CPU, 1GB RAM.

Table 4.4. Cost of each operation executed.

Notation	Devices (ms)	Network (ms)	Description
T_{hash}	0.006	0.002	Cost of a one-way hash operation
T_{mul}	0.04	0.013	Cost of a multiplication operation over an elliptical curve
T_{Exp}	0.37	0.123	Cost of an exponential operation
T_{pair}	0.06	0.02	Cost of a bilinear pairing operation
T_{es}	0.0068	0.0023	Cost of an AES encryption operation.
T_{IBE}	0.806	0.269	Cost of a Boneh and Franklin BF-IBE encryption operation.
T_{PK}	6.62	2.20	Cost of an IND-CCA encryption.

Table 4.5 shows the results of the computational cost analysis and the respective comparisons. The protocol proposed in this chapter required the second lowest computational cost in comparison to [5], [6] and [7], while PPAKA-HMAC [5] showed the lowest computational cost. However, the protocol of [5] is not safe against attackers inside the group of devices and suffers from several security flaws, as lack of confidentiality among devices in the same group, which are shown in Table 4.3.

Table 4.5. Comparison of computational costs.

Protocol	Devices (ms)	Server Network (ms)	TOTAL (ms)
PPAKA-HMAC [5]	$4nT_{Exp}+3.5nT_{Hash} +0.5n^2T_{Hash} + 2n\,T_{Mul}$ $= 1.58n + 0.001n^2$	-	$0.001n^2 + 1.58n$
PPAKA-IBS [5]	$0.5n^2T_{Exp} + 7.5nT_{Exp} + 0.5 n^2T_{Mul}+ 4.5nT_{Mul}+ n^2T_{Hash} + 3nT_{Hash} + n^2T_{Pair} + nT_{Pair}$ $=0.27n^2 + 3.03n$	$nT_{Exp} + nT_{Hash}$ $= 0.125n$	$0.27n^2+0.428n$
CN-GD2C [6]	$2n^2T_{Exp} - 2nT_{Exp} + 2n^2T_{Hash} - 2nT_{Hash} +n^2T_{ES} - nT_{ES} + n^2T_{PK} - nT_{PK}$ $= 7.38n^2 -7.38n$	$2n^2T_{Hash} - 2nT_{Hash} +n^2T_{ES} - nT_{ES} + n^2T_{PK} - nT_{PK} = 2.2n^2 – 2.2n$	$9.58n^2 - 9.58n$
GRAAD [7]	$3n^2T_{Exp} - 3nT_{Exp} + 5n^2T_{Hash} - 5nT_{Hash} +2n^2T_{ES} - 2nT_{ES} + n^2T_{IBE} - nT_{IBE}$ $= 1.96\,n^2 -1.96n$	$4n^2T_{Hash} - 4nT_{Hash} +2n^2T_{ES} - 2nT_{ES} + n^2T_{IBE} - nT_{IBE} + n^2T_{Pair} + nT_{Pair}$ $= 0.3n^2 - 0.3n$	$2.26\,n^2 -2.26n$
Proposed Protocol	$2nT_{Pair} + 3.5nT_{Hash} + 2.5nT_{Mul} + 0.5n^2T_{Hash} + 0.5n^2T_{Mul}$ $= 0.023\,n^2+0.241n$	$4.5nT_{Hash} + 0.5nT_{Mul} + 0.5n^2T_{Hash} + 0.5n^2T_{Mul} + T_{Hash} = 0.008\,n^2+0.016n + 0.002$	$0.031n^2+0.257n + 0.002$

First, the devices take $2nT_{Mul} + nT_{Hash}$ to calculate signature $\sigma_{Di\text{-}j}$. Then, $2nT_{Hash}$ is consumed for the calculation of public key $QID_{Di\text{-}j}$ and $h_{Di\text{-}j}$ of all devices in the group. $2nT_{Pair}+ nT_{Mul}$ is spent on the verification and authentication devices. Finally, they expend $[(n^2\text{-}n)/2] * (T_{Hash} + T_{Mul})$ to calculate a different session key for each pair of devices and nT_{Hash} is consumed for the generation of $Verif_{SKDi\text{-}j}$. Consequently, the total computational cost for the devices is $2nT_{Pair} + 3.5nT_{Hash} + 2.5nT_{Mul} + 0.5n^2T_{Hash} + 0.5n^2T_{Mul}$. If we consider the cost of operations presented in Table 4.5, the computational cost for the devices can be expressed as $0.023\,n^2 + 0.241n$.

The involvement of the HSS/AuC in the authentication procedure is minimal, as seen in Table 4.1, in the server network column. The cost of the construction and management of the binary tree is 2nThash - Thash. The server network takes $2T_{Hash}$ to generate $GID_{j,}$ and GK_j for the group. Then, $2nT_{Hash} + nT_{Mul}$ is consumed for the calculation of a temporary identity $TID_{Di\text{-}j}$ and a public key $QID_{Di\text{-}j}$ for each device. Finally, $[(n^2\text{-}n)/2] * (T_{Hash} + T_{Mul})$ is spent on the calculation of a different session key for each pair of devices and nT_{Hash} is expended to generate $Verif_{SKDi\text{-}j}$'. Therefore, the computational cost for the server network is $4.5nT_{Hash} + 0.5nT_{Mul} + 0.5n^2T_{Hash} + 0.5n^2T_{Mul} + T_{Hash}$. If we consider the cost of operations presented in Table 4.5, the computational cost for the server network can be expressed as $0.008\ n^2+0.016n + 0.002$.

The computational costs are reduced with the aggregation of signatures at the group leader, which provides the authentication of all group members in a single bilinear pairing operation. Figure 4.5 shows the results.

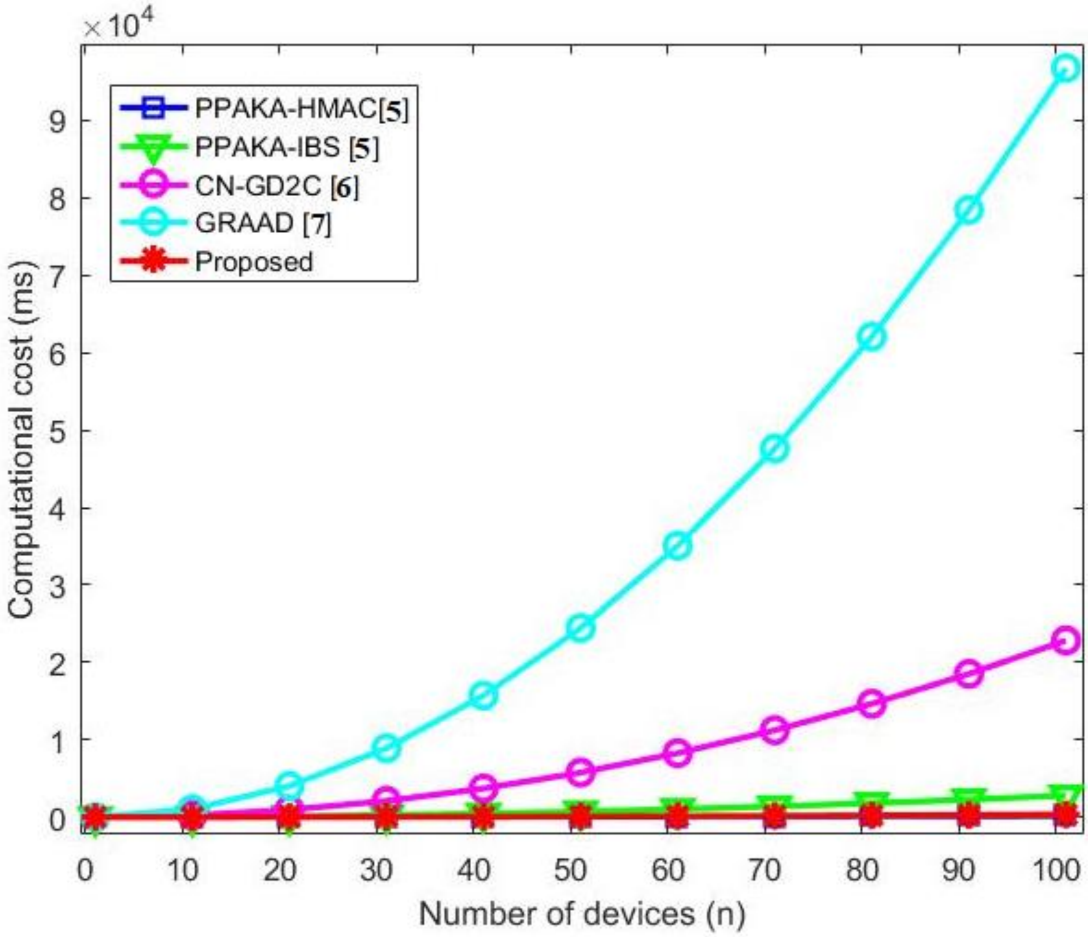


Figure 4.5. Comparison of computational costs.

The curves in Figure 4.5 show the good results of the protocol proposed in this chapter good regarding computational costs. The curves of PPAKA-HMAC [5], PPAKA-IBS [5] and the protocol proposed in this chapter are slightly different due to similar costs, thus their computational costs could be considered similar, for N<100. Nonetheless, the many security issues of PPAKA-HMAC [5] and PPAKA-IBS [5] compromise their general performance, as justified in section 4.4. The protocols of [6] and [7] required the highest costs, since they must perform a mutual authentication procedure for each pair of devices (or each connection established). All calculations made for each connection generated an overhead that is a function of $(n^2-n)/2$.

The reduction in computational costs, achieved by the protocol proposed in this chapter, impacts on the D2D communication development due to the devices' resource limitations and is accomplished by the aggregation of signatures at the group leader. Only one authentication procedure is required for the authentication of all devices of a group, which reduces the number of times each parameter is calculated.

4.5.2 Communication Cost

This section is devoted to the communication cost of the proposed protocol and a comparison among [5], [6] and [7] in a scenario with *n* devices in a group. Table 4.6 shows the size of each parameter in bits. The values were the same assumed in [5], [6] and [23].

Table 4.6. Size of each parameter.

Parameter	Size (bits)
ID/TID/PID	128
SID	64
Rand/ MAC	128
Hash/Exp	160
AES/Asymmetric Enc.	256
Timestamp	32
LAI	40

Table 4.7 shows the results of the communication cost analysis. All the parameters exchanged since the D2D discovery were considered.

Table 4.7. Comparison of communication costs.

Protocol	Devices	Server	TOTAL (bits)
PPAKA-HMAC [5]	$128n^2 + 1696n$	$128n^2+352n$	$256n^2+2048n$
PPAKA-IBS [5]	$128n^2 + 1696n$	$128n^2+256n$	$256n^2+1952n$
CN-GD2C [6]	$2304n^2-2304n$	$1120n^2-1120n$	$3264n^2-3264n$
GRAAD [7]	$2352n^2-2352n$	$864n^2-864n$	$2464n^{2-}2464n$
Proposed Protocol	1120n-160	576n+264	1696n-104

The protocol proposed in this chapter required the lowest communication cost, which is a consequence of the reduction in the volume of messages and parameters exchanged achieved by the aggregation and redistribution of authentication parameters in the group leader.

First, in the device discovery, each device broadcasts its TID_{Di-j} to devices nearby and receives the TID_{Di-j} of the interested ones. After device discovery and group generation, devices send TID_{Di-j}, σ_{Di-j}, and T_{Di-j} to the group leader, which generates a cost of 480(n-1) bits. Then, it broadcasts the subsequent parameters (TID_{D1-j},TID_{D2-j},…, T_{D1-j},T_{D2-j},…, σ_j, T_j), which costs 320n+192 bits. Finally, the devices send $Verif_{SKDi-j}$ and a timestamp T_{2Di-j} to HSS/AuC, which generates a cost of 192n bits. Consequently, the communication cost for the devices is 992n-288.

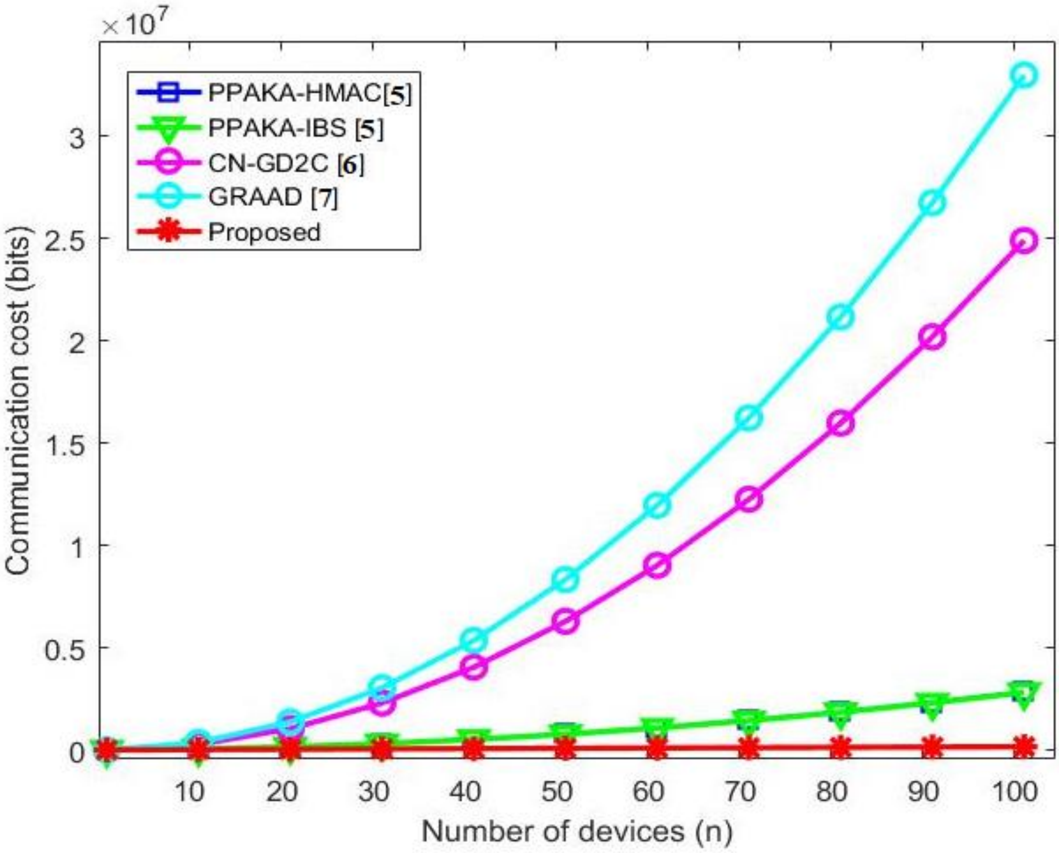


Figure 4.6. Comparison of communication costs.

In the registration/group organization phase, the server network sends $TID_{Di\text{-}j}$, $QID_{Di\text{-}j}$, $S_{Di\text{-}j}$, GID_j, GK_j, $SEC_{Di\text{-}j}$, and LAI_j to the devices, which generates 576n + 264 bits of communication cost. Therefore, the total communication cost of the protocol proposed in this chapter is 1568n-24 bits.

The protocols CN-GD2C [6] and GRAAD [7] required the highest communication cost, since they perform an authentication procedure for each pair of devices connected, which increases the number of messages exchanged in a quadratic order. Figure 4.6 shows the results. The protocol proposed in this chapter clearly shows the best communication cost in comparison to [5], [6] and [7].

4.5.3 Energy Cost

The energy cost evaluation is based on the proposals presented in Kumar et al. [24] and He et al. [25], which consider that the maximum CPU power of devices (W) is approximately 10.88 Watts. The energy overhead was calculated in the following way: ETotal = CCTotal × W, where CCTotal is the computational cost calculated of each operation performed as seen in section 4.5.1. Table 4.8 and Figure 4.7 show the comparison of energy costs among the protocol proposed in this chapter and other protocols from the literature.

Table 4.8. Comparison of energy costs.

Protocol	Devices	Server	TOTAL (mJ)
PPAKA-HMAC [5]	$0.0326n^2 + 17.2n$	-	$0.0326n^2 + 17.2n$
PPAKA-IBS [5]	$0.7178n^2 +2.81n$	$1.36n$	$0.7181 n^2 + 4.17n$
CN-GD2C [6]	$80.29n^2 + 80.29n$	$24.39n^2-24.39n$	$104.681n^2-104.681n$
GRAAD [7]	$24.7n^2-24.7n$	$3.38n^2-3.38n$	$28.08n^2 - 28.08n$
Proposed Protocol	$0.2502n^2+1.7n$	$0.0109n^2 + 0.1687n+0.02176$	$0.2611n^2+1.87n+0.02176$

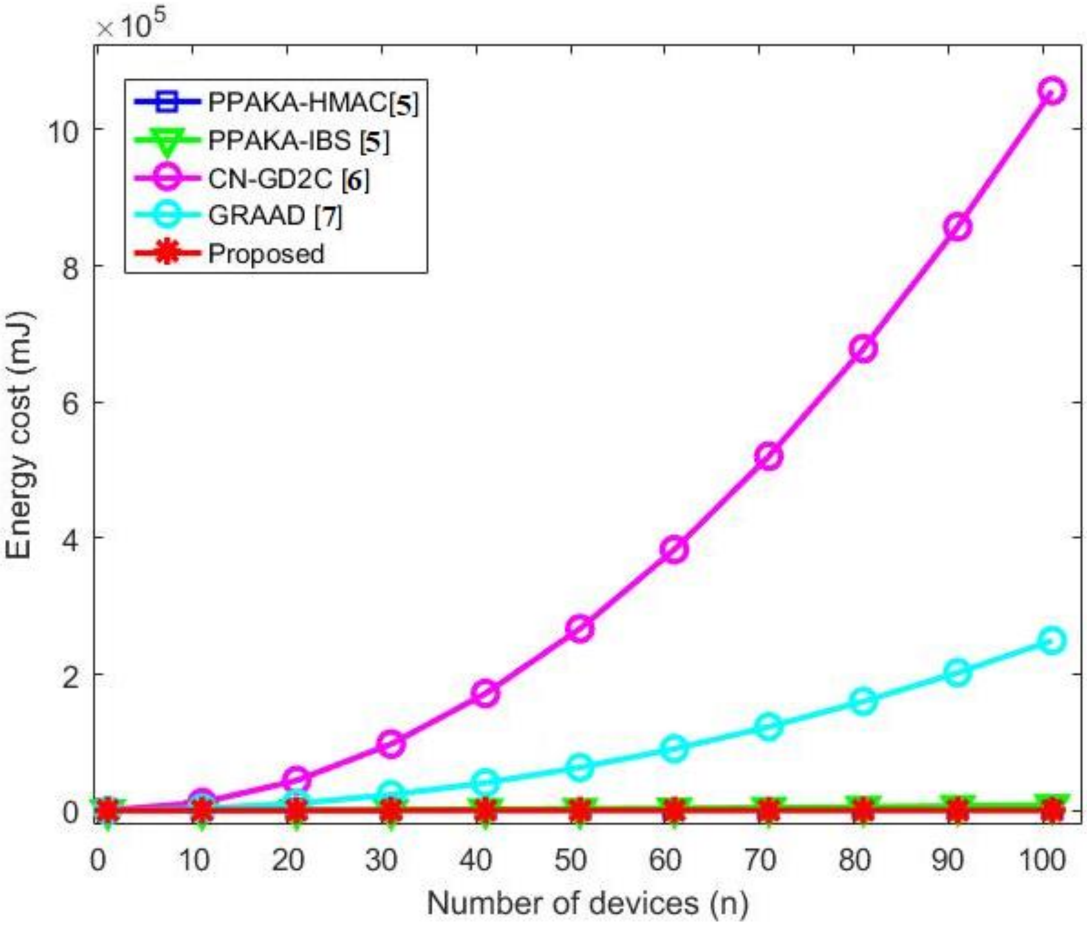


Figure 4.7. Comparison of energy costs.

The curves in Figure 4.7 are based on the results showed in Table 4.8; the proposed protocol has the lowest energy consumption when compared to the protocols of [5], [6] and [7]. This good result is obtained because the energy consumption of the protocols is directly associated to the message processing effort. Consequently, a good energy efficiency is related to a reduced computational cost.

4.6. Formal Validation

The protocol proposed in this chapter was formally validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) software [26], which simulates security-sensitive protocols. The language used in the simulations is High-level Protocol Specification Language (HLPSL) and its code is divided into roles, one for each entity involved in the authentication procedure (see Appendix A, Figures 4.11, 4.12 and 4.13).

The objectives verified were the ability of the protocol to perform D2D mutual authentication and key agreement and the secrecy of parameters, as session keys, GID_j, and LAI_j (see Figure 4.8).

The analysis was based on On-the-Fly-Model-checker (OFMC) [27] and Constraint Logic-Based Attack Searcher (CL-AtSe) [28], backends available at AVISPA. Both backends return "SAFE" if the protocol analyzed is considered safe; otherwise, they return "UNSAFE". According to Figures 4.9 and 4.10, the protocol proposed in this chapter was considered safe by the analysis.

goal
    authentication_on Di-j_Dk-j
    authentication_on Dk-j_Di-j
    secrecy_of $SK_{D1\text{-}leader}$
    secrecy_of $SK_{D1\text{-}2}$
    secrecy_of $SK\__{D2\text{-}leader}$
    secrecy_of GIDj
    secrecy_of LAIj
end goal

Figure 4.8. Simulation goals

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/validacaoCerto.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.32s
  visitedNodes: 46 nodes
  depth: 7 plies
```

Figure 4.9. OFMC backend result.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/validacaoCerto.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 58 states
  Reachable  : 34 states
  Translation: 0.19 seconds
  Computation: 0.00 seconds
```

Figure 4.10. CL-AtSe backend result.

4.7. Conclusions and Future Work

The development of the 5th generation of mobile networks is directly related to the IoT, hence, D2D communication, which provides a direct communication between two devices without the intermediation of network infrastructure, as the 3GPP core network. Since D2D is still in its early stages, some concerns must be considered for its full implementation.

Some of them are related to security and performance of devices, as addressed in [8], [9] and [10]. A properly performed AKA would avoid several security problems.

The traditional authentication and key agreement schemes cannot perform D2D authentication, therefore, new protocols must be designed towards fulfilling such a demand. Since many devices aim at exchanging data) directly with each other, group organization might facilitate the authentication procedure.

The protocol was designed towards ensuring a set of security properties and minimizing resource consumption. The security objective was validated by AVISPA and the resource consumption was evaluated with a focus on computational and communication costs.

A group leader election based on computational power, storage, battery life, and accessibility (as in [16], [21], [22]) is important for an extended network lifetime. Since the leader receives authentication parameters and signatures from all devices in the group and aggregates them into one single signature, which enables the authentication of a group of devices in a single bilinear pairing operation. The leader redistributes the information received from the other devices in the group through a single broadcast message. Moreover, the use of aggregated signatures reduces the communication costs, since they also reduce the number of messages exchanged between devices.

Another relevant characteristic of proposed protocol is related to the core network (HSS/AuC) participation in the authentication of devices, which is minimum and consists of the generation of some parameters and messages of verification. Consequently, the costs generated by its involvement are considerably reduced if compared with a scenario where the HSS/AuC participation is predominant. The computational cost is reduced because a single session performed can authenticate all the legit devices in a group. The communication costs are reduced because the authentication messages are aggregated in the group leader prior to be sent to the HSS/AuC, reducing the communication overhead of the network. The energy cost is educed because it is directly proportional to the computational cost, which had the cost reduction justified previously.

Therefore, the protocol proposed in this chapter showed the best performance when compared to the protocols of [5], [6] and [7]. Reductions in costs are essential for the implementation of D2D due to the resource-constraint nature of the devices, which are accomplished due to the use of aggregated signatures.

The proposed protocol authenticates groups of devices in D2D communication, and has proven robust regarding security, since it accomplished several security properties and resisted attacks, as addressed in section 4.4. It has also shown safer than the protocols designed by [5], [6] and [7], which faced confidentiality and privacy problems and lack of resistance to DoS and redirection attacks. A semi-automated formal validation by AVISPA [26] proved the security of the protocol.

Future work includes trust management for D2D communications and the adaptation of the protocol for scenarios involving e-health/m-health and smart cities. Additionally, characteristics and properties of Cyber Physical Systems (CPS) have been considered for the design of authentication protocols for D2D communication.

Appendix: An HLPSL Code Defining the role of devices, leader and HSS.

```
role role_Dij(Dij, Dkj ,Dleader, HSS:agent, P, GID,TIDdij, TIDdkj, TIDdleader, GK,
SECi, SECREa, SKik, SKileader, Sidi:text,
           Key_set_Dij_HSS:(symmetric_key) set,SND,RCV:channel(dy))
played_by Dij def=    local
                State:nat,
                Tdij, Tdkj, Hi, Ri, PK, U, V:text,
                H3:function,
                Key_1:symmetric_key
        init
                State := 0
        transition        1. State=0 /\ RCV(start) =|>
                   State':=1 /\ Tdij':=new()
                /\ Ri':=new()
                /\ TIDdij':=new()
                /\SND(TIDdij',mul(Sidi,TIDdij'),mul(Hi,SECi,Ri',PK),TIDdij')
                3. State=1 /\ RCV(U',V') =|>
                  State':=2 /\secret(SKileader', SKD1-leader
                  ,{})    /\ secret(SKik', SKD1-2,{})
                  /\ Key_1':=new()
                  /\Key_set_Dij_HSS':=cons(Key_1',Key_set_Dij_HSS)
                  /\SND({H3(SKik.SKileader)}_Key_1')
end role
```

Figure 4.11. Device role.

```
role role_Dleader(Dleader, Dkj, Dij, HSS:agent,
            P, GID, TIDdij, TIDdkj, TIDdleader, GK, SECleader, SECREc,
            SKileader, Sidleader:text,
            Key_set_Dleader_HSS:(symmetric_key) set,SND,RCV:channel(dy))
played_by Dleader
def=
        local
        State:nat,
        Tdkj, Tdij, Hleader, U, V, Ui, Vi, Uk, Vk, Rleader, PK, SKkleader:text,
        H3:function,
        Key_1:symmetric_key
        init
                State := 0
        transition
                1. State=0 /\ RCV(Ui',Vi') =|> State':=1
                2. State=1 /\ RCV(Uk',Vk') =|> State':=2 /\
                SND(TIDdleader,mul(Sidleader,TIDdleader'),mul(Hleader,SECleader,
                Rleader,PK),TIDdleader,U,V)
                /\
                SND(TIDdleader,mul(Sidleader,TIDdleader'),mul(Hleader,SECleader,
                Rleader,PK),TIDdleader,U,V)
                /\ SKkleader':=new()
                /\ secret(SKkleader', SK_D2-leader,{})
                /\ secret(SKileader', SKD1-leader,{})
                /\ Key_1':=new()
                /\ Key_set_Dleader_HSS':=cons(Key_1',Key_set_Dleader_HSS)
                /\ SND({H3(SKileader.SKkleader')}_Key_1')
end role
```

Figure 4.12. Leader device role.

```
role role_HSS(Dkj, Dij, Dleader, HSS:agent,
        P, GID, TIDdij, TIDdkj, TIDdleader, GK, SECk, SECi, SECleader, SECREa,
        SECREb, SECREc, SKik, SKileader, SKkleader:text,
        Key_set_Dij_HSS:(symmetric_key)    set,Key_set_Dkj_HSS:(symmetric_key)
        set,Key_set_Dleader_HSS:(symmetric_key) set,SND,RCV:channel(dy))
played_by HSS def=
        local
                State:nat,
                H3:function,
                Key_3,Key_2,Key_1:symmetric_key
        init
                State := 0
        transition
                5. State=0 /\ in(Key_1',Key_set_Dij_HSS)
                /\RCV({H3(SKik.SKileader)}_Key_1') =|>
                State':=1
                /\Key_set_Dij_HSS':=delete(Key_1',Key_set_Dij_HSS)
                /\ secret(SKileader', SK$_{D1-leader}$,{})
                /\ secret(SKik', SK$_{D1-2}$,{})
                6. State=1
                /\ in(Key_2',Key_set_Dkj_HSS)
                /\RCV({H3(SKik.SKkleader)}_Key_2') =|>
                State':=2
                /\Key_set_Dkj_HSS':=delete(Key_2',Key_set_Dkj_HSS)
                /\ secret(SKkleader' SK_$_{D2-leader}$
                ,{})
                /\ secret(SKik', SK$_{D1-2}$,{})
                7. State=2
                /\ in(Key_3',Key_set_Dleader_HSS)
                /\ RCV({H3(SKileader.SKkleader)}_Key_3') =|>
                State':=3
                /\ Key_set_Dleader_HSS':=delete(Key_3',Key_set_Dleader_HSS)
                 /\ secret(SKkleader' SK_$_{D2-leader}$,{})
                 /\ secret(SKileader', SK$_{D1-leader}$,{})
end role
```

Figure 4.13. HSS role.

4.8. References

[1] A., Zhang, J., Chen, R. Q., Hu, and Y. Qian, "SeDS: Secure data sharing strategy for D2D communication in LTE-Advanced networks.", *IEEE Transactions on Vehicular Technology*, v.65, n.4, pp.2659-2672, 2016.

[2] M., Wang, Z., Yan, and V., Niemi, “UAKA-D2D: Universal authentication and key agreement protocol in D2D communications”, *Mobile Networks and Applications*, v.22, n.3, pp.510-525, 2017.

[3] 3GPP System Architecture Evolution (SAE). Security Architecture, 3GPP TS 33.401, V8.2.1, 2009. https://www.3gpp.org/ftp/Specs/archive/23_series/23.303/ Visited on June 2019.

[4] 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 June /2019.

[5] R. H., Hsu, and J., Lee, “Group anonymous D2D communication with end-to-end security in LTE-A.”, *Communications and Network Security (CNS), 2015 IEEE Conference,* pp. 451-459, 2015.

[6] R. H., Hsu, J., Lee, T. Q., Quek, and J. C., Chen, “GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks.”, *IEEE Transactions on Information Forensics and Security,* v.13, n.2, pp.449-464, 2018.

[7] M., Wang, and Z., Yan, “A survey on security in D2D communications”, *Mobile Networks and Applications*, v.22, n.2, pp.195-208, 2017.

[8] M., Wang, and Z., Yan, “Privacy-preserving authentication and key agreement protocols for D2D group communications”, *IEEE Transactions on Industrial Informatics*, v.14, n.8, pp.3637-3647, 2018.

[9] A., Zhang, and X., Lin, “Security-aware and privacy-preserving D2D communications in 5G”, *IEEE Network*, v.31, n.4, pp.70-77, 2017.

[10] 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*, v.19, n.2, pp.1054-1079, 2017.

[11] W., Diffie, and M., Hellman, “New directions in cryptography”, *IEEE Transactions on Information Theory,* v.22, n.6, pp.644-654, 1976.

[12] Abd-Elrahman, E., Ibn-Khedher, H., and Afifi, H., “D2D group communications security.”, *2015 International Conference on Protocol Engineering (ICPE) and*

International Conference on New Technologies of Distributed Systems (NTDS), pp.1-6, 2015.

[13] Kwon, H., Kim, D., Hahn, C., and Hur, J., “Secure authentication using ciphertext policy attribute-based encryption in mobile multi-hop networks”, *Multimedia Tools and Applications*, v.*76*, n.19, pp.19507-19521, 2017.

[14] P., P., Tayade and P., Vijayakumar, "Enhancement of Security and Confidentiality for D2D Communication in LTE-Advanced Network Using Optimised Protocol", *Wireless Communication Networks and Internet of Things,* Lecture Notes in Electrical Engineering 493. Springer, pp.131-139, 2018.

[15] Cao, J., Ma, M. and Li, H. GBAAM: Group-based Access Authentication for MTC in LTE Networks. *Security and Communication Networks*, v.8, n. 17, p.3282-3299, 2015.

[16] D., Choi, S., Hong, and H. K., Choi, “A group-based security protocol for Machine Type Communications in LTE-Advanced.”, *Wireless Network*, v. 21, n. 2, pp.405-419, 2015.

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

[18] K., Hussain, A. H., Abdullah, K. M., Awan, F., Ahsan, and A. Hussain, “Cluster head election schemes for WSN and MANET: a survey”, *World Applied Sciences Journal*, v.23, n.5, pp.611-620, 2013.

[19] F. S., Gharehchopogh, and H., Arjang, “A Survey and Taxonomy of Leader Election Algorithms in Distributed Systems”, *Indian Journal of Science and Technology*, v.7, n.6, pp.815-830, 2014.

[20] Dolev, D., and Yao, A., “On the security of public key protocols.” *IEEE Transactions on information theory,* v.29, n.2, pp.198-208, 1983.

[21] C., Lai, R., Lu, D., Zheng, H., Li, and X. S., Shen, “GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications”, *Computer Networks,* v.99, pp.66-81, 2016.

[22] J., Cao, M., Ma, and H., Li, “GBAAM: group-based access authentication for MTC in LTE networks” *Security and Communication Networks*, v.8, n.17, pp.3282-3299, 2015.

[23] C., Lai, R., Lu, and X., Shen, “SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks”, *Computer Networks*, v. 57, n. 17, pp. 3492-3510, 2013.

[24] Kumar, A., and Om, H., "Handover Authentication Scheme for Device-to-Device Outband Communication in 5G-WLAN Next Generation Heterogeneous Networks", *Arabian Journal for Science and Engineering*, n.43, v.12, pp.7961-7977, 2018.

[25] He, D., Chan, S., and Guizani, M., "Handover authentication for mobile networks: security and efficiency aspects", *IEEE Network*, n.29, v.3, pp.96-103, 2015.

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

[27] D., Basin, S., Moedersheim, and L., Vigano, "OFMC: A symbolic model checker for security protocols", *International Journal of Information Security*, v.4, n.3, pp. 181-208, 2005.

[28] M., Turuani, "The CL-Atse protocol analyser", *International Conference on Rewriting Techniques and Applications,* pp. 277-286, Springer, Berlin, Heidelberg, 2006.

[29] D., Boneh, and M., Franklin, "Identity-based encryption from the Weil pairing", *Annual international cryptology conference,* pp. 213-229, Springer, Berlin, Heidelberg, 2001.

Chapter 5

A LIGHTWEIGHT AUTHENTICATION SCHEME FOR D2D COMMUNICATION IN M-HEALTH WITH TRUST EVALUATION

Abstract. Mobile health (m-health) has promised to provide fast and reliable medical monitoring and solutions to thousands of patients with a mobile device and sensors coupled to their bodies. Device-to-Device (D2D) communication is a strong candidate to enable communication among m-health and patients' mobile devices, since it is one of the most expected technologies to be used in the Internet of Things (IoT). Its security regarding m-health requires attention due to the delicacy of the data exchanged in the respective process. Traditional authentication and key agreement (AKA) schemes are not suitable for D2D scenarios, since they might expose patients to security vulnerabilities. This research proposes a secure and lightweight scheme for the mutual authentication of m-health devices in D2D communication. The scheme is based on Shamir secret sharing and aims at security robustness and reduction in resources consumption. The chapter also addresses the trustworthiness of devices involved in data relay and device discovery procedures.

Keywords: D2D, M-health, Trust, ProSe, authentication.

5.1. Introduction

Mobile devices 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., smartphones, smartwatches, smartTVs, smart homes and vehicles, and smart metering). Intel [20] expects more than 200 billion devices will be connected by 2020. 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 twenty-four hours a day and seven days a week towards preventing the escalation of diseases and affording immediate relief in emergencies. The literature reports several reviews on the advantages of m-health applications for improving health service quality (e.g. Free et al. [23]).

The m-health system works with a group of sensors coupled to a patient’s body and a mobile device that receives the measurements from such sensors and send 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 channel through which messages are exchanged.

Device-to-Device communication (D2D) is a strong candidate to enable the communication of devices involved in m-health applications. According to Wang and Yan [2], it has improved the efficiency of communication systems by reducing delays and power dissipation and fostering multifarious new applications. Additionally, the discovery of devices by nearby devices enables 3GPP D2D proximity services (ProSe), hence, D2D communication among close devices through a communication channel. The technical specification regarding D2D ProSe, TS 36.843 [3], specifies the requirements and procedures for devices discovery as stated by 3GPP.

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. [4] and include authentication, privacy, anonymity, non-repudiation, 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 the mutual authentication among devices and also among devices and the core of the 3GPP network. Harn [5] proposed three interesting authentication schemes for a group of devices based on the secret sharing scheme created by Shamir [6] and the Lagrange interpolation formula. Their security is based on the Discrete Logarithm Problem (DLP), according to which secrets are computationally unbreakable.

To the best of our knowledge, an issue not yet addressed 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 not all devices are trustworthy to perform such a task. Consequently, trust assurance and evaluation become a critical problem for D2D m-health applications~~.~~

This research proposes a mutual authentication and key agreement scheme for D2D devices in m-health for enabling patients to securely send their medical information to a health center and doctors. It has been designed to forecast the relay of data when devices are outside the 3GPP coverage area, or inside of it, but with no access to the network, considering the necessity of computational trust.

The main contributions of the present research include:

1. a secure secret sharing scheme for D2D m-health applications that fulfills all security aspects discussed in the 3GPP D2D security specification TS 33.303 [8];
2. a mutual authentication scheme for D2D m-health groups of devices;
3. an adaptation of the trust mechanism based on the local trust concept proposed by Yan et al. [7] that enables D2D devices to choose the most reliable device in their proximity to perform the relay of their data; in the protocol proposed in this chapter, the local trust secret key encryption is based on symmetric cryptography, producing reduced computational costs when compared with [7], which is based on asymmetric cryptography;
4. an evaluation of computational, communication and energy costs of the proposed scheme;
5. an assessment of the security properties of the scheme and possible protection against attacks and threats; and
6. a semi-automated formal validation of the protocol.

The remainder of the chapter is organized as follows: Section 5.2 discusses some related work; Section 5.3 presents the 3GPP reference architecture for proximity services (ProSe); Section 5.4 introduces the scheme, its phases, the key agreement process, and the trust evaluation; Section 5.5 provides a security analysis and comparisons with other protocols; Section 5.6 reports a performance evaluation with computational and communication costs; finally, conclusions and future work are discussed in Section 5.7.

5.2 Related Work

M-health security has been the focus of several studies. Zhang et al. [9] 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 [10], the scheme proposed by [9] shows some security weaknesses, such as vulnerability to an insider attack, which affect its confidentiality of the system. Zhou enhanced it by improving the CLGSC scheme and proposed a certificateless signcryption scheme for m-health [13], a towards correcting the above-mentioned vulnerabilities in CLGSC scheme. According to the author, it uses some extra variables in the authentication procedure in comparison to [9], which enables attackers to obtain some authentication parameters through queries.

Harn [5] presented three authentication schemes based on Shamir's secret sharing [6], 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 [5] 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 [11] 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 [6], 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. [7] 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: in 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 levels, or only GT, 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 [8], 3GPP TS 23.303 [12] and TR 36.843 [3]. 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 Unit Entities (UEs), and data transfer between ProSe Function and ProSe-enabled UE. The second specification [8] 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 the direct communication range. The technical report in [3] addresses enhancements for ProSe UE-to-network relay for commercial and public safety applications, as wearables and IoT devices.

5.2.1 3GPP Reference Architecture for ProSe Services

5.2.1.1 Functional Description

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

Home Subscriber Server (HSS) – part of the Evolved Packet Core (EPC) of LTE networks that contains users' and subscribers' information, supports authentication and authorization of devices, and manages mobility;

ProSe Function Server (PFS) – the logical function used for network-related actions required for ProSe that plays different roles for each feature of ProSe [12], such as generation of trust tokens and identities in the management of D2D communication;

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

User Equipment (UE) – a mobile device associated with each user; and

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

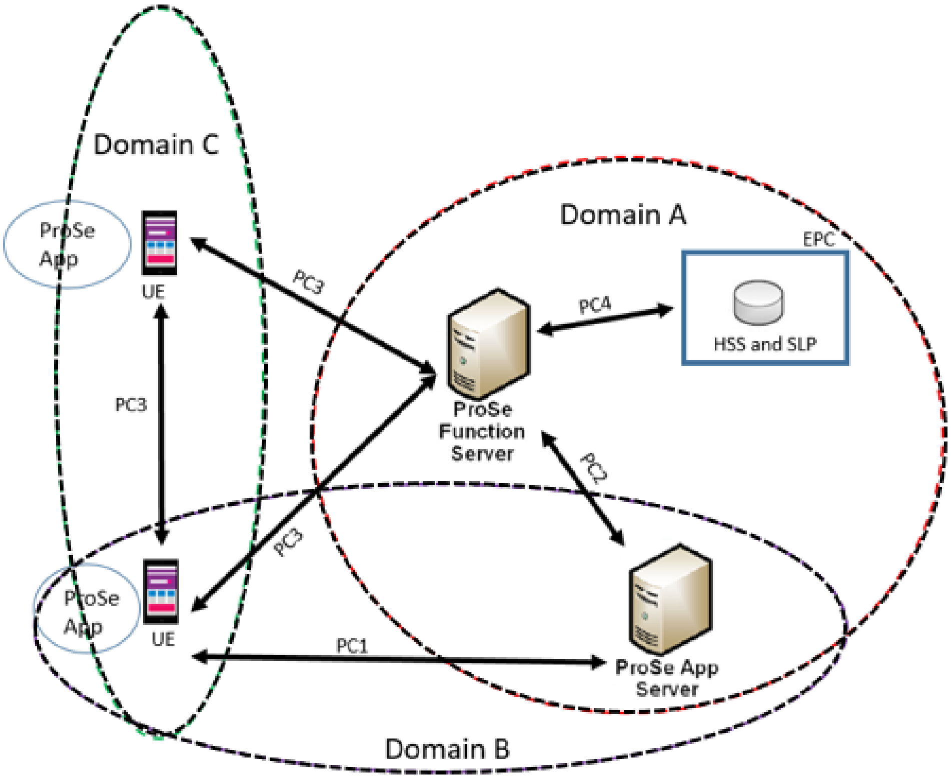


Figure 5.1. 3GPP ProSe reference architecture (based on Yan et al. [7].

Figure 5.1 shows the reference architecture proposed by 3GPP for Proximity Services [7]. Domain A is inside the red dotted circle and comprises the security domain of the 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 by users' equipment.

5.2.1.2 Reference Points

Below is a list of reference points of 3GPP TS 23 303 [12], as shown in Figure 5.1:

- PC1: the reference point between the ProSe application in the UE and the ProSe Application Server that defines application-level signaling requirements;
- 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 targeted application user. PAS determines if the originating UE is allowed to discover the targeted UE;
- 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;
- PC4: the reference point between HSS and PFS used by the latter to retrieve EPC-level discovery-related subscriber data; and
- PC5: the reference point between UEs used for control and user plane for direct discovery.

5.2.1.3 3GPP ProSe device discovery

The 3GPP device discovery is detailed in technical specification TS 23303 [12] and involves the detection and identification of other devices (UEs) located in the proximities using E-UTRAN or WLAN direct radio signals. The device discovery can be open if no permission is necessary from the UE being discovered or restricted if permission is required. It can also be used by applications to initiate ProSe Direct Communication.

It has two models for operations:

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

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. It is only executed only restricted discovery.

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, the devices can send a request to the PFS to be authorized to monitor. If they succeed, and have a Discovery Filter, they can start monitoring. Finally, when the monitoring devices detect one or more devices that matched the filter, they report them to the PFS.

For a more detailed description, readers should consult 3GPP TS 23.303 [12].

5.2.1.4 Security Requirements

The several security requirements and aspects expected by the 3GPP standardization [8] for D2D communication that uses the ProSe architecture include:

Avoidance of attacks – The proposed systems must be resistant to several attacks, e.g., replay and impersonation.

Authorization of devices – The system must allow only currently authorized devices to be discovered by other UEs.

Tracking of devices –The tracking of devices based on their discovery messages should be minimized.

Authentication of devices and PFS – The devices involved must authenticate the source of the received data communications. UE and PFS must authenticate each other.

Integrity and Confidentiality – The integrity and confidentiality of data exchanged among the entities must be guaranteed.

Privacy – The privacy of the users must be provided.

5.3 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 have direct access to the 3GPP infrastructure. In the second case, the D2D nodes operate as the relay of a network, as proposed by Zhang et al. [9] and Zhou [13]. Moreover, computational trust is fundamental for a proper operation of the system. HSS manages the device authentication and keys 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 5.2 shows the architecture of the protocol, derived from 3GPP ProSe [12] standards, with all entities involved.

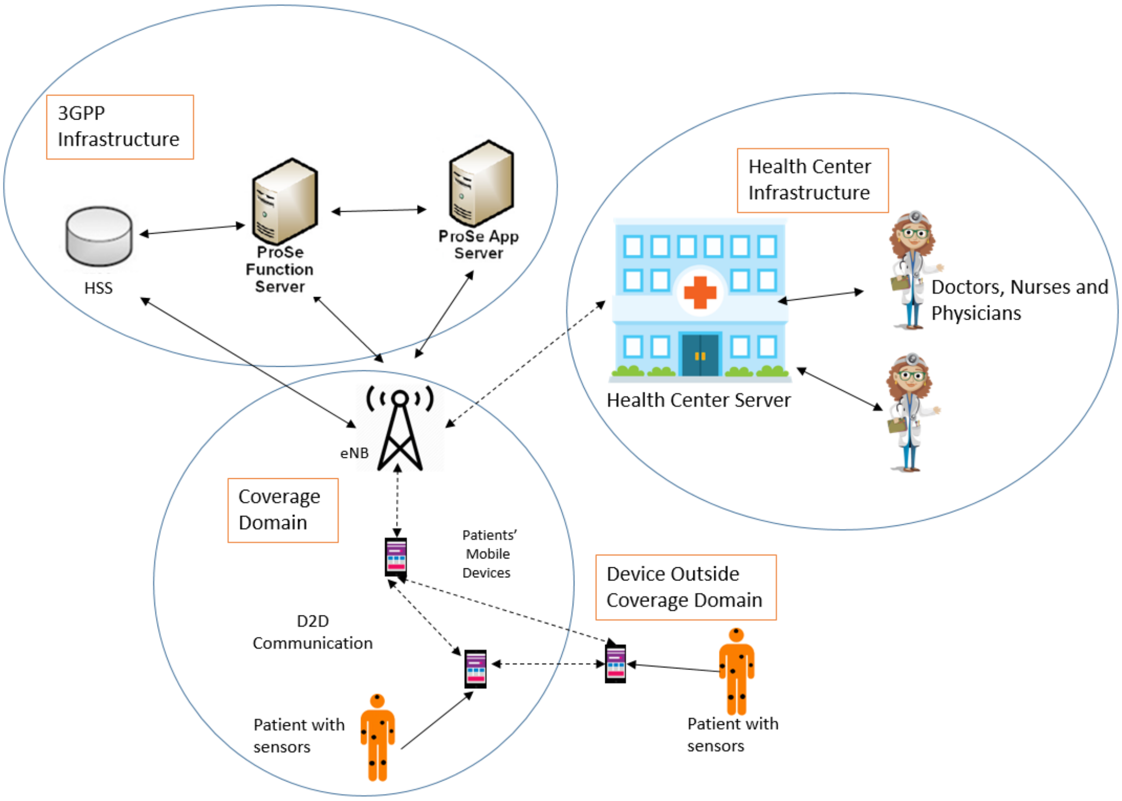


Figure 5.2. System architecture.

Table 5.1 shows the main symbols and parameters used in the proposal. Some basic assumptions are:

1. The health center infrastructure is considered trustworthy and secure;
2. The entities of the 3GPP infrastructure and their communication channels are considered trustworthy and secure;
3. The channel between the patients' device and their respective body sensors is considered safe; and
4. The D2D communication channels and the channel between devices and eNB are considered unsafe and 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 following five phases must be executed for a patient outside the coverage area to send their data: initialization, registration, mutual authentication, trust evaluation, and encryption/decryption. Table 5.1 shows the main symbols and parameters used in the proposal.

Table 5.1. Parameters used in the protocol.

Symbol	Description
Di	Patient i or device i, where i = 1,2,3,…, 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 with order q.
g	Point Generator of Gq. i = 1,2,3…
$f_l(x)$	Random polynomial, l = 1,2,…
z	Master private key.
M_{Kpub}	Master public key.
SK_{Di}	Private key of device D_i, pair (x, Y_{Di})
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 = Di, HSS. i = 1,2,3…
h_1	Temporary identity generation hash function. H_1: {0,1}* $\longrightarrow$ Z_p*
h_2	Device’s partial private key generation hash function. H_2: {0,1}* $\longrightarrow$ G_q
h_3	Symmetric key generation hash function. H_4: {0,1}* x G_q $\longrightarrow$ G_p
H	Shamir’s secret hash function.
LT_i	Local trust level of device i.
$LTK_{Du\text{-}Du`}$	Local trust secret key.
$\longrightarrow$	Secure channel.
$\dashrightarrow$	Insecure channel.

The protocol uses asymmetric and symmetric cryptography: asymmetric cryptography is used in the generation of private keys and temporary identities for mutual authentication, while symmetric cryptography is used in trust evaluation to reduce costs when compared to [7]. It is based on the Asynchronous (t; m; n) Group Authentication Scheme (GAS) with Multiple Authentications, proposed by Harn [5], because it provides a way of sharing a secret among 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.

5.3.1 Initialization

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

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 ϵ Zq* as the master private key and calculates M_{Kpub} = z*g to obtain the master public key.

HSS selects three hash functions, h_1(.), h_2(.) and h_3(.), (described in Table 5.1) for the mutual authentication phase and generates j random numbers, $R_j \in Z_p^*$, (j = 1, 2, ..., i) for each device and for itself for the calculation of a set of temporary identities TID_{Di} :

$$TID_{Di} = h_1(ID_{Di} || R_j * z) \quad (5.1)$$

It also selects its own TID_{HSS}:

$$TID_{HSS} = h_1(ID_{HSS} || R_j * z) \quad (5.2)$$

Next, it sends each device its respective set of TID_{Di}. A different TID_{Di} 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 (similarly to [9]), chooses a y ϵ Zq, and calculates:

$$Y_{Di} = h_2(TID_{Di} \,||\, y)*M_{Kpub} \quad (5.3)$$

Finally, it sets the partial key calculated for each respective device and publishes the following parameters: {g, G, E(Zp*), M_{Kpub}, TID_{HSS}, h_1(.), h_2(.), h_3(.), H(.)}.

5.3.2 Registration

The ProSe device discovery mechanism is applied in this phase for the discovery of nearby devices, as described in [12]. 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 [9]) choosing x ϵ Zq* and calculates its public key:

$$PK_{Di} = x * Y_{Di} *g \qquad (5.4)$$

Then, it sends TID_{Di}, PK_{Di} and a timestamp T_{Di_1} to the other devices, and the nearby device sends HSS all the information received from relay devices.

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

Next, each device chooses an integer $v_{Di\text{-}Dj}$ ϵ Zq* as a secret value to be sent to other devices and HSS, encrypted with its public key, as follows:

$$V_{Di-Dj} = EPK_{Di}(v_{Di-Dj}) \qquad (5.5)$$

j is 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 the secret values necessary for the generation of session keys.

The asynchronous mode of the group authentication protocol designed by Harn [5] 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 the sake of determining the membership of a user to a group. Therefore, considering what is proposed by [5], we have designed our authentication scheme:

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

$$f_l(x) = \sum_{i=0}^{t-1} a_i x^i \bmod p \qquad (5.6)$$

All coefficients a_i are in finite field Zp*.

HSS generates two tokens for each device calculating $f_l(\mathrm{TID_{Di}})$. Each $\mathrm{TID_{Di}}$ has its respective token. HSS also calculates its own two tokens $f_l(\mathrm{TID_{HSS}})$ and finds integers $w_{i,j}, d_{i,j}, j = 1{,}2$ ϵ Zp*, such that $\mathrm{S} = \sum_{j=0}^{k} d_{i,j} f_j(w_{i,j})$, where $w_{i,1} \neq w_{i,2}$ for every pair *i* and *j*. It then generates a secret S, as in [5]:

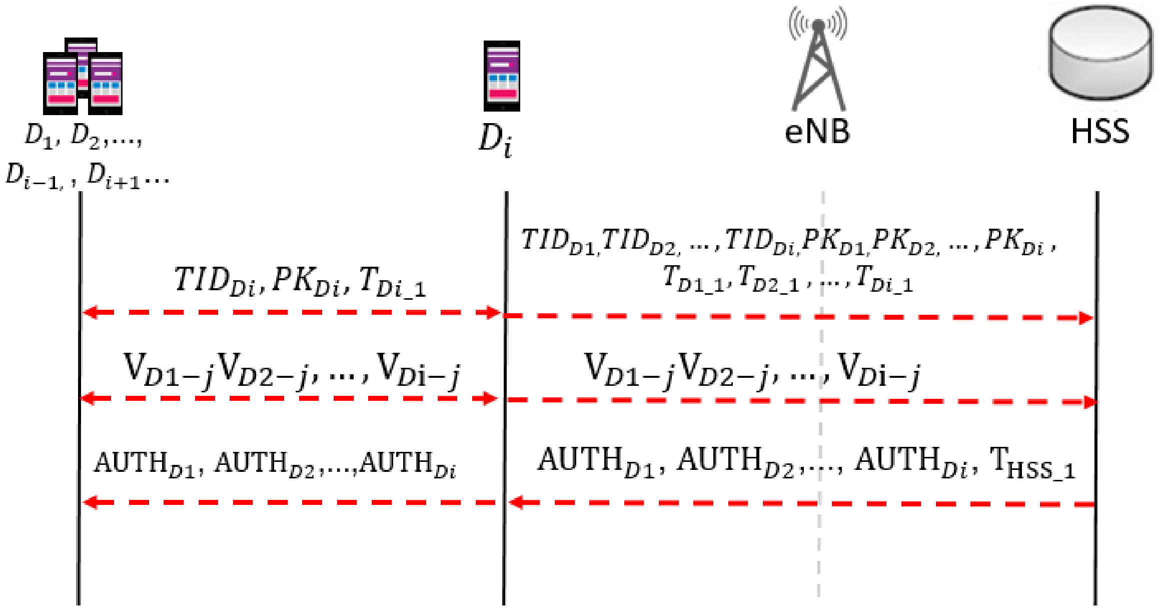
$$\mathrm{S} = f_l(0) = a_0 \tag{5.7}$$

$$\mathrm{S} = g^{\sum_{j=1}^{2} d_{i,j} f_j(w_{i,j}) \bmod q} \bmod p \tag{5.8}$$

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

$$\mathrm{AUTH_{Di}} = [\mathrm{EPK_{Di}}(\mathrm{TID_{HSS}}, \mathrm{H(S)}, w_{i,j}, d_{i,j}, f_l(\mathrm{TID_{Di}}), \mathrm{v_{HSS-Di}}), \mathrm{TID_{Di}}, \mathrm{T_{HSS\_1}}] \tag{5.9}$$

The devices decrypt the message and store the parameters. According to [5], the same token can be used multiple times. The generation of new tokens is optional after the first registration of devices.



$D_1, D_2, \ldots,$
$D_{i-1}, , D_{i+1} \ldots$
D_i
eNB
HSS
$TID_{Di}, PK_{Di}, T_{Di\_1}$
$TID_{D1,} TID_{D2,} \ldots, TID_{Di,} PK_{D1,} PK_{D2,} \ldots, PK_{Di},$
$T_{D1\_1}, T_{D2\_1}, \ldots, T_{Di\_1}$
$\mathrm{V}_{D1-j}\mathrm{V}_{D2-j}, \ldots, \mathrm{V}_{Di-j}$
$\mathrm{V}_{D1-j}\mathrm{V}_{D2-j}, \ldots, \mathrm{V}_{Di-j}$
$\mathrm{AUTH}_{D1}, \mathrm{AUTH}_{D2}, \ldots, \mathrm{AUTH}_{Di}$
$\mathrm{AUTH}_{D1}, \mathrm{AUTH}_{D2}, \ldots, \mathrm{AUTH}_{Di}, \mathrm{T_{HSS\_1}}$

Figure 5.3. Messages exchanged in the registration phase.

5.3.3 Mutual Authentication

Since the devices still must authenticate each other and HSS, each device selects a pair of non-used TID_{Di} and respective tokens $f_l(TID_{Di})$, l = 1,2, and computes its Lagrange component,(an adaptation of what is proposed in [5]), LC_{Di} through the Lagrange interpolating formula:

$$LC_{Di} = \sum_{l=1}^{2} d_{i,l} f_l(\mathrm{TID_{Di}}) \prod_{q=1;q\neq i}^{n} \frac{w_{i,l} - \mathrm{TID_{Di-q}}}{\mathrm{TID_{Di}} - \mathrm{TID_{Di-q}}} \, mod\ q \qquad (5.10)$$

Next, they calculate $e_{Di} = g^{LC_{Di}} mod\ p$.

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

$$LC_{HSS} = \sum_{l=1}^{2} d_{i,l} f_l(\mathrm{TID_{HSS}}) \prod_{q=1;q\neq i}^{n} \frac{w_{i,l} - \mathrm{TID_{Di-q}}}{\mathrm{TID_{HSS}} - \mathrm{TID_{Di-q}}} \, mod\ q, \qquad (5.11)$$

and its own $e_{HSS} = g_i^{LC_{HSS}} mod\ p$.

It generates a random value $r_{Di} \epsilon\ Zq *$. The devices send $TID_{Di}, e_{Di}, r_{HSS}$ and a timestamp T_{Di_2} to the other devices in the relay group and to HSS, which also send $TID_{HSS}, e_{HSS}, T_{HSS\_2}, r_{HSS}$ to the other devices in the relay group. After receiving such parameters, the entities verify the validity of the timestamp in order to avoid denial of service (DoS) attack. They proceed with the authentication procedure only if the timestamp is valid. Otherwise, they discard the respective entity. When each entity has a complete set of e_{Di} and e_{HSS}, a secret S' is calculated:

$$S' = (\mathrm{e_{HSS}} * \prod_{i=1}^{n} \mathrm{e_{Di}})\ mod\ p = g_i^{(LC_{HSS} + \sum LC_{Di}) mod\ q} mod\ p \qquad (5.12)$$

Again, an attacker must solve the DLP problem to obtain e_{Di}, e_{HSS} and S', as in [5].

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.

$$SKey_{Di-k} = h_3[(\mathrm{S} || r_{Di} || r_u || v_{Di-u} \oplus v_{u-Di}] \qquad (5.13)$$

where k is either a device D_k or HSS.

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.

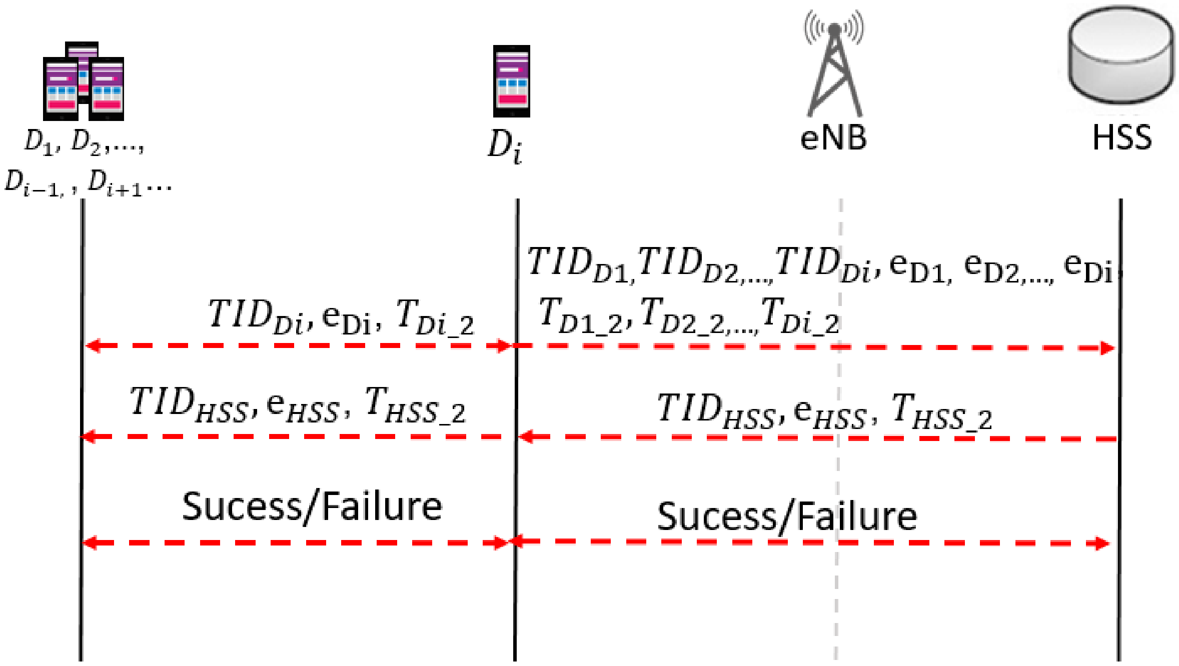

$D_1, D_2, \ldots,$
$D_{i-1}, , D_{i+1} \ldots$
D_i
eNB
HSS
$TID_{D1}, TID_{D2,\ldots,} TID_{Di}, e_{D1}, e_{D2,\ldots,} e_{Di}$
$T_{D1\_2}, T_{D2\_2,\ldots,} T_{Di\_2}$
$TID_{Di}, e_{Di}, T_{Di\_2}$
$TID_{HSS}, e_{HSS}, T_{HSS\_2}$
$TID_{HSS}, e_{HSS}, T_{HSS\_2}$
Sucess/Failure
Sucess/Failure

Figure 5.4. Message exchanged in the mutual authentication phase.

5.3.4 Trust evaluation

This phase is executed whenever a patient must send his/her health information to the doctor/physician but is not inside 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. [7]. However, we have created our own calculations that are different from those of [7], due to the use of symmetric cryptography aiming to cost reduction.

This phase is performed over an insecure channel. An architecture involving the use of relay devices, as shown in Figure 5.2, is employed in the proposed scheme. After the measurement of local trust, all devices considered trusted are candidates to be relay devices. Some calculations are made regarding trust indicators, as seen in section 5.3.4.1.

5.3.4.1 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 $LT_{Du'}$ 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 $LT_{Du'}$ of a device $D_{u'}$, it sends the $\mathrm{TID}_{\mathrm{Du}'}$ to another device D_k , which once has communicated with device $D_{u'}$, to request its local trust evaluation $LT_{Du'}$. D_k generates local trust level $LT_{Du'}$ 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 Du':

$$\mathrm{B}_{\mathrm{Dk-Du}} = \mathrm{ESKey}_{\mathrm{Dk-Du}}(\mathrm{LT}_{\mathrm{Du}'}) \qquad (5.14)$$

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

$$LTK_{Du-u'} = h_3(\mathrm{S}||\, r_{Di} \,||\, r_u \,||\, v_{Di-u} \oplus v_{u-Di} ||\mathrm{LT}_{\mathrm{Du}'}) \qquad (5.15)$$

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

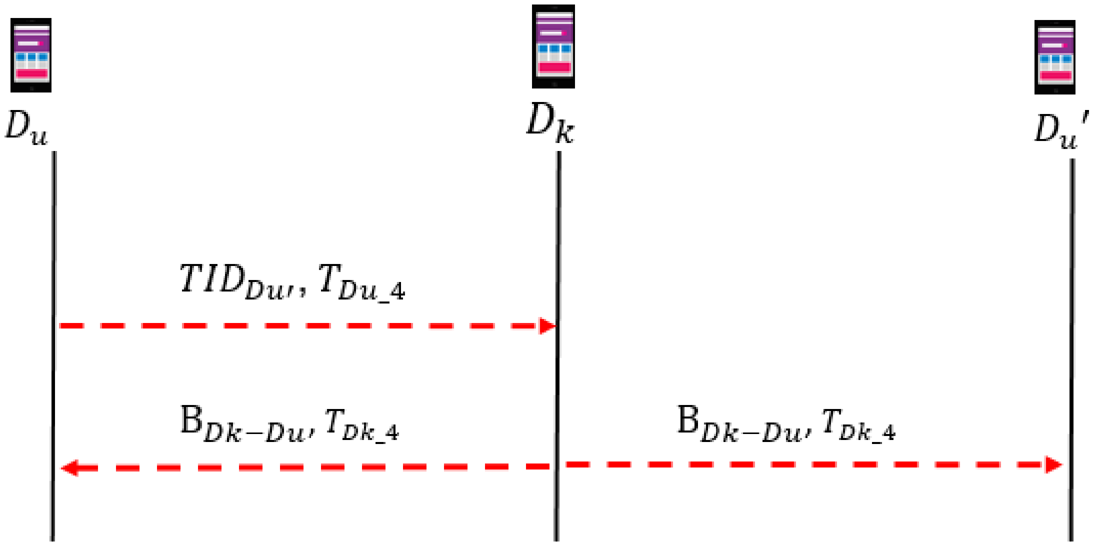

D_u
D_k
D_u'
$TID_{Du'}, T_{Du\_4}$
$\mathrm{B}_{Dk-Du'} T_{Dk\_4}$
$\mathrm{B}_{Dk-Du'} T_{Dk\_4}$

Figure 5.5. Messages exchanged when LT is used.

5.3.5 Encryption/Decryption

Finally, after the tests, the original device encrypts the data with session key $SKey_{Di-HSS}$:

$$M = ESKey_{Di-HSS}\text{ (Data)} \tag{5.16}$$

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

$$C = ELTK_{Du-Du'}(\mathrm{M}) \tag{5.17}$$

The message is sent to the most adequate device in the relay group with T_{Di_5} and $ESKey_{Du-Du'}(LT_{Du'})$. Then, D$_{u'}$ calculates the secret key:

$$LTK_{Du-u'} = h_3[(\mathrm{S}||\, r_{Di} \,||\, r_u \,||\, v_{Di-u} \oplus v_{u-Di} \,||\mathrm{LT}_{\mathrm{Du}'}\,], \tag{5.18}$$

Decrypts the message, and obtains M:

$$M = DLTK_{Du-u'}(C) \tag{5.19}$$

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_{Di_5}. The process is repeated until the device nearest the 3GPP infrastructure has been reached. This device sends M with a timestamp T_{Dm_5} to HSS.

HSS decrypts M using session key $SKey_{Di-HSS}$ 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.

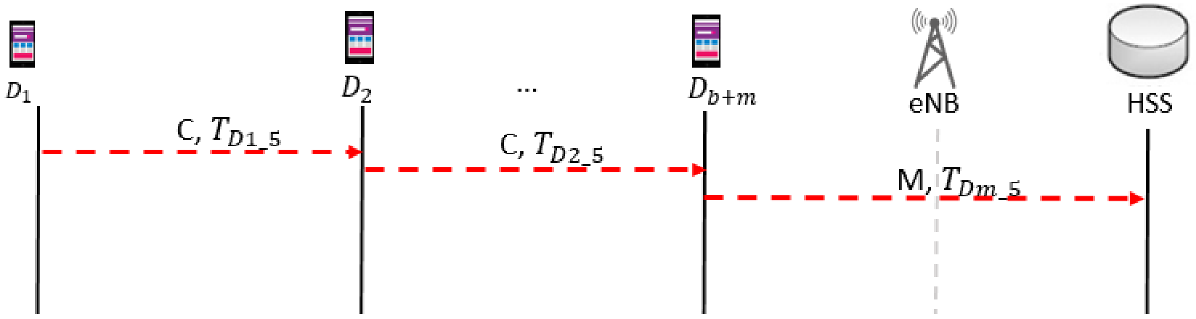


Figure 5.6. Encrypted data sent to HSS.

5.4 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.

5.4.1 Mutual Authentication – devices perform mutual authentication to authenticate the other devices in the relay group. Each device calculates its Lagrange component (LC_{Di}) and e_{Di} , and they share e_{Di} 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_{Di} and sends it with the respective TID_{Di} to HSS, which calculates MAC'_{Di} and checks if MAC'_{Di} = MAC_{Di}. If the values are equal, HSS authenticates the devices and proceeds. Otherwise, the operation is terminated. Then, HSS generates its own Lagrange component LC_{HSS} and $\mathrm{e}_{\mathrm{HSS}}$ and sends $\mathrm{e}_{\mathrm{HSS}}$ to the group of relay devices. Each device recalculates its own Lagrange component $\mathrm{LCnew}_{\mathrm{Di}}$, e_{Di-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 to be computationally infeasible.

5.4.2 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_{Di}, timestamps T_{Di} and session keys SK_{Di-HSS}/SK_{Di-Dk} in each authentication procedure.

5.4.3 Confidentiality – The scheme provides confidentiality of patients' data by generating a different session key SK_{Di-HSS}/SK_{Di-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.

5.4.4 Integrity – Data integrity is guaranteed by the encryption of the data sent by each patient through a securely established session key SK_{Di-HSS}/SK_{Di-Dk} before it is sent over an insecure channel. When HSS decrypts the messages with the appropriated session key, it knows the information was generated by the genuine source and was not modified on the way to the destination.

5.4.5 Anonymity – The anonymity of entities, devices, and HSS is safeguarded through the exchange of only temporary identities ($\mathrm{TID}_{\mathrm{Di}}$, $\mathrm{TID}_{\mathrm{HSS}}$) 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 acquired offline.

5.4.6 Non-Repudiation – Non-repudiation certifies an entity cannot deny its actions. In the proposed scheme, it is guaranteed through the use of permanent ($\mathrm{ID}_{\mathrm{Di}}$) and temporary identities ($\mathrm{TID}_{\mathrm{Di}}$, $\mathrm{TID}_{\mathrm{HSS}}$) and private and public keys.

5.4.7 Session Key Security – The security of the session key is ensured by confidential information Y_{Di} and ID_{Di} in its generation process:

$$SKey_{Di-k} = h_3[(S || r_{Di} || r_u || v_{Di-u} \oplus v_{u-Di}] \tag{5.13}$$

5.4.8 Resistance to Impersonation Attack – Impersonation attack is avoided through the use of 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.

5.4.9 Resistance to Replay Attack – Replay attack is avoided by freshly generated parameters, such as random values r_{Di} and timestamps T_{Di} in the mutual authentication phase, generation of session keys, and use of different $\mathrm{TID}_{\mathrm{Di}}$ and $\mathrm{TID}_{\mathrm{HSS}}$ in each session.

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

5.4.11 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.

Some of such security objectives were not accomplished by [9], [13] and [11]. First, any of the compared protocol performs the trust evaluation of the relay devices. Secondly, as shown by [10], the scheme proposed in [9] 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 [9] and [13] 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 [11] does not protect the anonymity of devices because it does not mention the use of temporary or pseudo-identity instead of their permanent identities. Table 5.2 shows a comparison among our scheme and those of [9], [13] and [11].

Table 5.2. Comparison of security objectives among protocols.

Security Objectives	Zhang et al. [9]	Mustafa and Philip [11]	Zhou [13]	Proposed Protocol
Mutual Authentication	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
Non-Repudiation	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
Resistance to impersonation attack	Yes	Yes	Yes	Yes

5.5 Performance Analysis

This section reports on a performance analysis of the protocol proposed in this chapter regarding computational, communication and energy costs in each authentication session executed, and a performance comparison among our scheme and those of [9] and [13].

5.5.1 Computational Cost

The values in Table 5.3 are based on Choi et al. 2014[14] and Hsu et al. 2018 [15] and configured as follows: Intel Core Duo 1.86 GHz and 2 gigabyte RAM under an Ubuntu 11.10 operating system [14]; smartphone HTC One X with Android 4.1.1, 1.5 GHz Quad-core ARM Cortex-A9 CPU, 1GB RAM [15].

Table 5.3. Cost of each 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 operation.

Table 5.4 shows a comparison of the computational costs (in milliseconds) among protocol and those designed by [9] and [13]. An environment with n devices 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. The calculations from the Trust Evaluation phase were performed only by the devices involved in the relay of data.

Table 5.4. Comparison of computational costs.

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

The devices take $nT_{mul} + 3nT_{ECC}$ in the registration phase to calculate their partial public key and encrypt/decrypt secret values v_{Di-Dj}. Then, they take $3nT_{mod} + nT_{hash} + nT_{exp}$ in the mutual authentication and key agreement phase to calculate their Lagrange component, secret S' and session key $SKey_{Di-k}$. Next, mT_{hash} + mT_{AES} is required for the encryption of local trust result $LT_{Du'}$ and calculate local trust secret key $LTK_{Du-u'}$. Finally, the devices expend mT_{hash} + $(2m+1)T_{AES}$ to encrypt the patients' information generating M and encrypting/decrypting M with local trust secret key $LTK_{Du-u'}$. Consequently, the total computational cost for the devices is nT_{mul} + $3nT_{mod}$ + $(n+2m)T_{hash}$ + nT_{exp} + $3nT_{ECC}$ + $(3m+1)T_{AES}$. According to the computational costs shown in Table 5.4, the computational cost for the devices is 6.827n + 1.851m + 0.483 ms.

The 3GPP network takes $(n+1)T_{mul}$ + $2nT_{hash}$ to calculate temporary identities and partial private keys for each device and its master public key in the initialization phase. It takes $(2n+4)T_{mod}$ + $(2n+3)T_{exp}$ + nT_{ECC} to generate tokens for each device and for itself in the registration phase. Next, it requires $3T_{mod}$ + nT_{hash} + T_{exp} to calculate the Lagrange component of HSS, secret S', and session keys $SKey_{Di-HSS}$. Finally, the network takes T_{AES} to decrypt message M and obtain the source patient's information. Therefore, the computational cost for the core network is $(n+1)T_{mul}$ + $(2n+7)T_{mod}$ + $3nT_{hash}$ + $(2n+4)T_{exp}$ + nT_{ECC} + T_{AES}. According to the operation cost in Table 5.4, the computational cost for the network is 1.674n + 2.133 ms.

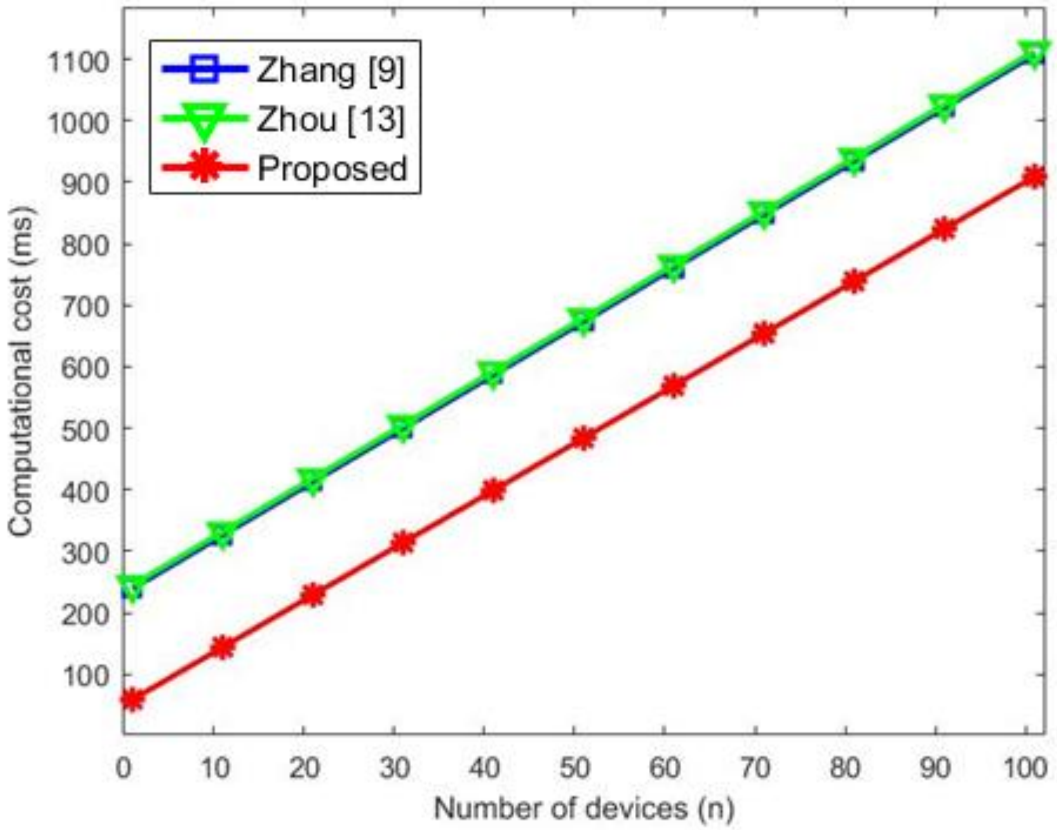


Figure 5.7. Comparison of computational costs.

The lines in Figure 5.7 show satisfactory results of the protocol proposed in this chapter regarding computational costs. A situation in which 25% of devices are involved in the relay of data was considered. The protocol clearly shows better costs than [9] and [13], which yielded slightly different results, since [13] is an improvement of [9] and, consequently, most calculations are similar to those in [9]. The main difference between [9] and [13] is the correction of security vulnerabilities by using more variables in the authentication procedure. In terms of operations, [13] only requires the calculation of an extra elliptic curve cryptography-based (ECC-based) scalar multiplication on G_1 when compared to [9].

Our scheme has shown excellent computational performance regarding all subjects addressed. The use of Shamir's secret sharing and the scheme proposed by [5] 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.

5.5.2 Communication Cost

This section is devoted to the evaluation and comparison of the communication cost (in bits) of the protocol proposed in this chapter. The scheme was compared with [9] and [13], in a scenario with n registered devices and m devices involved in the relay of data. Table 5.5 shows the size of each parameter in bits. The values were the same as those assumed in [14] and [15].

Table 5.5. Size of each parameter exchanged.

Parameter	Size (bits)
ID/TID	128
Hash/Exp	160
AES	256
Asymmetric/Symmetric key	256
ECC	512
Timestamp	32

Table 5.6 displays the results of the communication cost analysis and the comparison with [9] and [13]. All messages and parameters exchanged for the adequate functioning of the protocol, since the initialization phase, were considered.

Table 5.6. Comparison of communication costs.

Protocol	Devices (bits)	Network (bits)	TOTAL (bits)
Zhang et al. [9]	800n + 2144m + 2144	832n + 2816	1632n + 2144m + 4960
Zhou [13]	800n + 2016m + 2016	832n + 2016	1632n + 2016m + 4704
Proposed Protocol	1248n + 736m + 288	416n + 320	1664n + 736m + 608

The protocol proposed in this chapter required the lowest communication resources consumption, and yielded better results than [9] and [13] due to the reduced number of parameters exchanged by relays (a function of m), which compensates for the higher costs for the total of devices (a function of n).

First, the devices send TID_{Di}, PK_{Di}, and a timestamp T_{Di_1} to the other devices and to HSS, which costs 416n bits. Then, they send their secret value to other devices and HSS, encrypted with their public key, V_{Di-Dj} , which costs 512n bits. Next, they send $TID_{Di}, e_{Di}, T_{Di\_2}$ to the other devices in the relay group and to HSS, which costs 320n bits. The devices involved in the relay exchange $TID_{Du'}$, $T_{Du'\_4}$, $SKey_{Di-k}$ and $T_{Dk'\_4}$ in the local trust phase, which costs 448m bits. Finally, in the encryption phase, the devices involved in the relay of data exchange C, T_{Di_5} and the border device send M and T_{Dm_5}, which costs 288m+288 bits. Therefore, the communication cost for the devices is 1248n + 736m + 288 bits.

The core of the 3GPP network sends TID_{Di}, PK_{Di} and a timestamp T_{Di_1} to each device in the registration phase, which costs 416n bits. Then, it sends TID_{HSS}, e_{HSS}, T_{HSS_2} to devices in the mutual authentication phase, which costs 320 bits. Consequently, the communication cost for the network is 416n + 320 bits.

The schemes proposed by [9] and [13] show higher communication costs, since they exchange more parameters to both authenticate devices and send patients' encrypted information. The protocol proposed in this chapter has shown better communication costs than [9] and [13], even with an additional phase to perform trust evaluation. The schemes in

[9] and [13] also require similar communication costs, since most part of the message exchange of [13] is similar to that of [9], because it exchanges an extra parameter of order Zq*. The main differences between [9] and [13] which produces consequences in terms of communication costs are due to the use of more variables in the authentication procedure by the proposal in 13].

Figure 5.8 shows a graphic with the results and comparison among the proposed scheme and [9] and [13] in a situation in which 25% of the devices are involved in the relay of data.

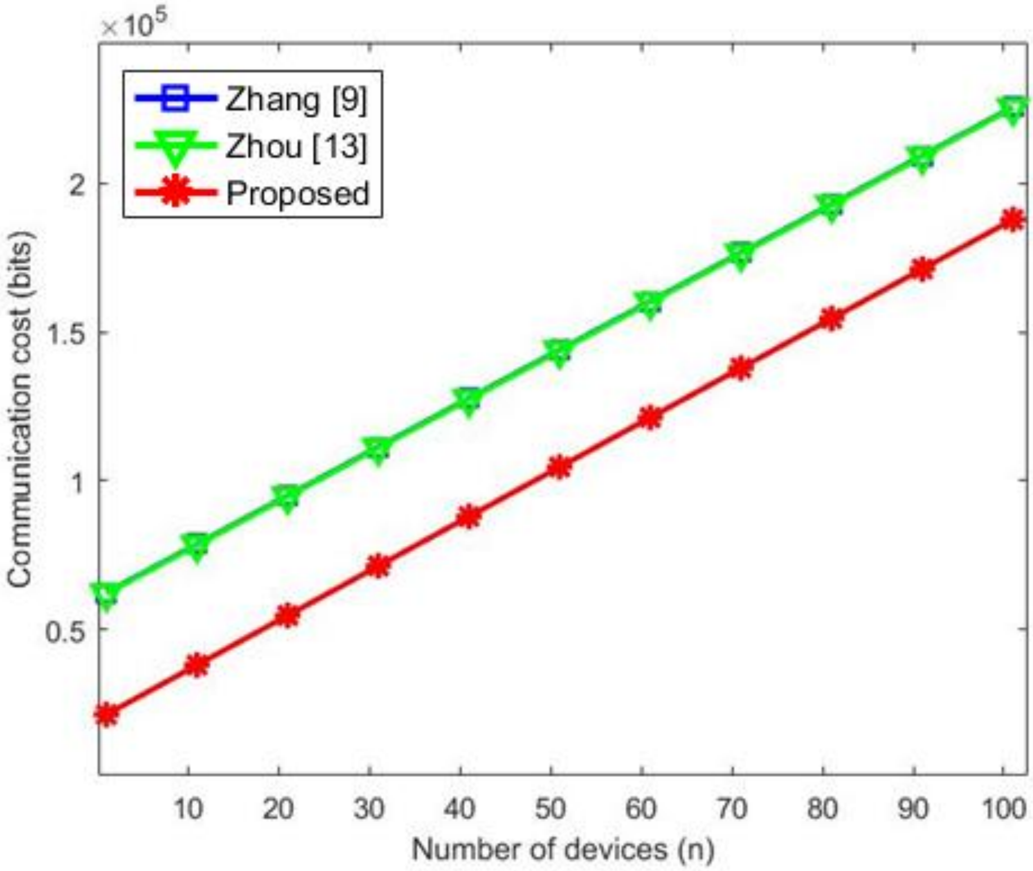


Figure 5.8. Comparison of communication costs.

5.5.2 Energy Cost

This section reports on an analysis of the energy cost of the protocol proposed in this chapter and a comparison with [9] and [13]. The evaluation is based on the proposals presented in Kumar et al. [18] and He et al. [19], 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 (see section 5.5.1). Table 5.7 shows the results.

Table 5.7. Comparison of energy costs.

Protocol	Devices	Network	TOTAL (mJ)
Zhang et al. [9]	80.3n + 85.2m + 243.93	14.14n + 55.81	94.44n + 85.19m + 299.53
Zhou [13]	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	92.48n + 20.13m + 28.4

Figure 5.9 shows a comparison based on the energy costs displayed in Table 5.7. The red line representing the protocol proposed in this chapter proves its lower energy consumption in comparison to the protocols of [9] and [13]. As occurred in the evaluation of communication and computational costs, the energy costs of [9] and [13] were similar. The main differences between [9] and [13] which produces consequences in terms of energy costs are the same as presented for computational costs, since energy cost depends on it. Our scheme also shows higher energy efficiency due to reduced processing efforts and computational cost.

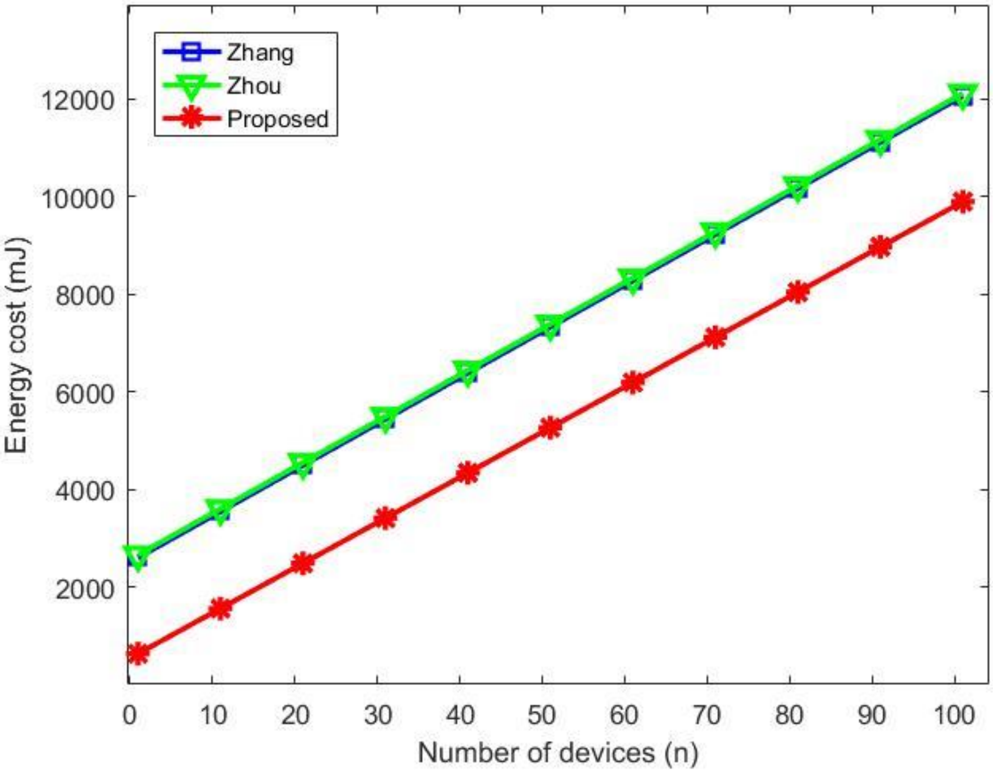


Figure 5.9. Comparison of energy costs.

5.6 AVISPA verification

The protocol proposed in this chapter was validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) [26], which simulates the messages exchanged among entities involved in an authentication scheme. AVISPA simulation is written in High-level Protocol Specification Language (HLPSL), which divides the message exchanges into roles that represent each of the entities involved in the authentication procedure. Figure 5.10 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 parameters, as session keys, GID_j, and LAI_j.

```
role
role_Dkj(Dkj:agent,Dij:agent,HSS:agent,TIDdij:text,TIDdkj:text,TIDhss:text,PKdkj:text,Vdkdj:text,AUTHdk:te
xt,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:symmetric_key,Key_4:symme
tric_key,Key_3:symmetric_key,Key_2:symmetric_key,Key_1:symmetric_key
        init
                State := 0
        transition
        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 5.10. Role of a device in HLPSL language for AVISPA software.

AVISPA has four backends to verify security. We used two of them, namely On-the-Fly-Model-checker (OFMC) [27] and Constraint Logic-Based Attack Searcher (CL-AtSe) [28]. Both backends return “SAFE” if the protocol analyzed is considered safe and “UNSAFE” if it has found an issue that might compromise security. According to Figures 5.11 and 5.12, the protocol proposed in this chapter was considered safe by the analysis.

```
% 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: 0.00s
  searchTime: 0.72s
  visitedNodes: 63 nodes
  depth: 8 plies
```

Figure 5.11. OFMC analysis.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

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

Figure 5.12. CL-AtSe analysis.

5.7 Conclusions and Future Work

The Internet of Things (IoT) is expected to provide the connection of 200 billion devices by 2020 [20]. Such devices have been designed for several new applications and creation of a framework of benefits to improve services and people's life quality, assure safety and security, and reduce expenses [21]. 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 [22]. 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, 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 [7]. The protocol proposed in this chapter has been designed to provide a new AKA scheme; it aims at fulfilling the security properties detailed by 3GPP specifications TS 23.303 [12] and TS 33.303 [8] and reducing resource consumption regarding computational and communication costs. Such a reduction has been achieved by the scheme adopted, as proposed by Harn [5], based on the Shamir's secret sharing [6]. A trust evaluation verified the close devices suitable for the relay of data. It was based on the scheme developed by [7], to guarantee the delivery of data from the source device to the health center.

The robustness of the protocol proposed in this chapter has been demonstrated by several security properties and resistance to attacks, as addressed in section 5.4. The scheme has proven safer than those of [9] and [13]. The protocol designed by [9] showed confidentiality issues and, consequently, is not resistant to attacks (e.g., insider and man-in-the-middle). The scheme of [13] is not resistant to DoS attack, and the one developed by [11] shows anonymity problems, since it offers no protection to devices' real identities. The protocol proposed in this chapter has proven to be the safest, because it has fulfilled all security objectives required by [12] and [8], as shown in Table 5.2, and achieved better performance, in comparison to [9] and [13], 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,

Future work will include the proposal of authentication and authorization protocols based on cyber-physical systems ([24], [25]), as well as the formal validation of the protocol proposed in this chapter by tools, such as Proverif and Tamarin. The simulation of the protocol by NS-3 or OMNET++ tools has been considered for the evaluation of energy efficiency and influence of device mobility.

5.8 References

[1] Huang, C., Yan, K., Wei, S., and Lee, D. H., “A privacy-preserving data sharing solution for mobile healthcare”, *2017 International Conference on Progress in Informatics and Computing (PIC)* pp. 260-265, 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] 3GPP TS 36.843 “Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects.” Mar. 2014. https://www.3gpp.org/ftp/Specs/archive/36_series/36.843/ ,Visited on November 2019.

[4] Haus, M., Waqas, M., Ding, A. Y., Li, Y., Tarkoma, S., & Ott, J. (2017). Security and privacy in device-to-device (D2D) communication: A review. *IEEE Communications Surveys & Tutorials*, n.19, pp.1054-1079.

[5] Harn, L., "Group authentication", *IEEE Transactions on computers*, n.62, pp.1893-1898, 2013.

[6] Shamir, A., "How to share a secret" *Communications*, pp.612-613, 1979.

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

[8] 3GPP TS 33.303 "Proximity-based services (ProSe); Security Aspects" Jun. 2018. https://www.3gpp.org/ftp/Specs/archive/33_series/33.303/. Visited on November 2019.

[9] Zhang, A., Wang, L., Ye, X., and Lin, X., "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems", *IEEE Transactions on Information Forensics and Security,* n.12, pp.662-675, 2017.

[10] Zhou, C., "Comments on "Light-Weight and Robust Security-Aware D2D-Assist Data Transmission Protocol for Mobile-Health Systems", *IEEE Transactions on Information Forensics and Security,* v.13, pp.1869-1870, 2018.

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

[12] 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.

[13] Zhou, C., "An improved lightweight certificateless generalized signcryption scheme for mobile-health system". *International Journal of Distributed Sensor Networks,* v.15, n.1, pp.1-16, 2019.

[14] Choi, D., Hong, S. and Choi, H.K., "A group-based security protocol for Machine Type Communications in LTE-Advanced", *Wireless Network*, v. 21, n. 2, pp.405-419, 2015.

[15] R. H., Hsu, J., Lee, T. Q., Quek, and J. C., Chen, "GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks.", *IEEE Transactions on Information Forensics and Security*, v.13, n.2, pp.449-464, 2018.

[18] Kumar, A., and Om, H., "Handover Authentication Scheme for Device-to-Device Outband Communication in 5G-WLAN Next Generation Heterogeneous

Networks”, *Arabian Journal for Science and Engineering*, n.43, v.12, pp.7961-7977, 2018.

[19] He, D., Chan, S., and Guizani, M., “Handover authentication for mobile networks: security and efficiency aspects”, *IEEE Network*, n.29, v.3, pp.96-103, 2015.

[20] Intel. “A guide to the Internet of Things”, Available on: https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html, Visited on September 2019.

[21] Hung, Mark. "Leading the iot." *Gartner Insights on How to Lead in a Connected World [On-line], 2017.* Available on: https://www. gartner. com/imagesrv/books/iot/iotEbook_digital. Visited on September 2019.

[22] Sadegh, S. Saeedeh, et al. "A framework for m-health service development and success evaluation." *International Journal of Medical Informatics,* v.112, pp.123-130, 2018.

[23] Free, C., Phillips, G., Felix, L., Galli, L., Patel, V., & Edwards, P., "The effectiveness of M-health technologies for improving health and health services: a systematic review protocol.", *BMC research notes* v.3, n.1, pp. 250, 2010.

[24] Essa et al. "Cyber Physical Sensors System Security: Threats, Vulnerabilities, and Solutions". *2nd International Conference on Smart Grid and Smart Cities (ICSGSC),* 2018. DOI: 10.1109/ICSGSC.2018.8541316.

[25] Laura Vegh. "Cyber-physical systems security through multi-factor authentication and data analytics". *2018 IEEE International Conference on Industrial Technology (ICIT),* 2018. DOI: 10.1109/ICIT.2018.8352379.

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

[27] D., Basin, S., Moedersheim, and L., Vigano, “OFMC: A symbolic model checker for security protocols”, *International Journal of Information Security,* vol.4, no.3, pp. 181-208, 2005.

[28] M., Turuani, “The CL-Atse protocol analyser”, *International Conference on Rewriting Techniques and Applications,* pp. 277-286, Springer, Berlin, Heidelberg, 2006.

Chapter 6

CONCLUSION

This dissertation main objective was to develop new authentication and key agreement protocols for D2D communication. It was proposed three protocols aiming at different scenarios of D2D communication in general and m-health, considering the 3GPP architecture.

The first protocol focused in the creation of a mutual authentication protocol for e-health in Telecare Medical Information Systems (TMIS) using cloud servers. It considers the mutual authentication of patients' devices, doctors and health centers with the cloud server. Devices can perform their authentication in two distinct ways: by direct connection with 3GPP network or performing relay through D2D communication.

The second protocol was developed considering large amounts of devices wanting to be connected at the same time. Therefore, it was designed to authenticate groups of devices. It considers an architecture where devices are assisted by 3GPP network and it is based on asymmetric cryptography. It uses a scheme of aggregated signatures to authenticate all devices that are part of a determined group at the same time. It is a protocol designed to provide confidentiality, privacy and anonymity of devices and avoidance of attacks such as DoS and impersonation. The treatment of devices as groups helps reducing costs such as communication, because less messages are exchanged in the channel.

The third protocol permits the authentication of devices located inside and outside the coverage area of the 3GPP network. It meets requirements as the necessity of grouping devices to perform authentication, reducing costs and improving security and the demand of trust management of devices performing relay to other devices located outside the 3GPP infrastructure coverage area.

All three protocols were had their security and performance evaluated and compared to other proposals published in the literature. The security evaluation and comparison regard fulfillment of properties as confidentiality, integrity, privacy and anonymity and the resistance to several attacks such as man-in-the-middle, impersonation, replay, among others. The three proposals have proven to be more robust than the other proposals in the comparison.

The performance evaluation was composed of the measurement of three costs: computational, communication and energy. The computational costs were evaluated based on the computational time of operations necessary to be execute at each authentication session of the protocol. The cost of each operation was obtained from other authentication schemes published in the literature. The communication costs were measured in bits, considering all the parameters present in the messages exchanged among entities during an authentication session. The communication costs obtained represent the amount of bandwidth consumed by each protocol. The cost of each parameter in bits was also obtained from authentication schemes published in recent years. Finally, energy costs were calculated as a function of computational cost and is measured by multiplying it by 10.88W, which is the power consumed in each second spent during the authentication session. Additionally, the proposed protocols were validated by AVISPA tool and proved to be secure to be used.

APPENDIX 1 – **Mutual Authentication Protocol for Cloud-based E-health Systems**

Ana Paula Golembiouski Lopes[1], Paulo R. L. Gondim[1], Jaime Lloret[2]

[1]Department of Electrical Engineering – University of Brasília (UnB)
Campus Universitário Darcy Ribeiro – Asa Norte – DF – Brasil
[2] Integrated Management Coastal Research Institute - Polytechnic University of Valencia (UPV) – Valencia – Spain.
anagolembiouski@aluno.unb.br, pgondim@unb.br, jlloret@dcom.upv.es

Abstract. *The development of the Internet of Things predicts several new applications, of which some are designed to be incorporated to e-health systems. The assistance of cloud computing in the authentication procedure can relieve resource-constrained devices employed in Telecare Medicine Information Systems (TMIS). Their security is fundamental for the achievement of optimal performance, regarding the sensibility of e-health shared data and, especially, the anonymity of patients and other entities. This paper introduces a new mutual authentication protocol for e-health systems that ensures security and surpasses the performance and security of other authentication procedures reported in the literature.*

1. Introduction

Among the several applications for the development of Internet of Things (IoT), e-health/m-health aims at providing health services through information and communication technologies. Such applications include, for example, monitoring by sensors coupled to the body of patients and connected by Body Area Network (BAN), diagnosis and remote provisioning of health services to patients over public channels.

The assistance of cloud servers is an alternative for supplying the large demands of storage and processing generated by multiple medical service providers and increasing operational efficiency. According to Mohit et al. (2017), in Telecare Medical Information Systems (TMIS), doctors and patients would work together through the cloud server. Patients send to the cloud server a report containing sensor's measures and a doctor collects the data, provides a diagnosis and finally sends a diagnosis report to the cloud server. Both data exchanges are performed through public channels.

Additionally, the use of cloud servers as auxiliaries to the storage and processing in e-health/m-health/TMIS requires special attention, due to the high sensitivity of the information exchanged among the cloud server and the entities involved. Information of the sensor measurements report and patient diagnosis can be crucial for saving lives and must not be accessed or modified by possible attackers.

A good example is the anonymity of entities, since the user of those systems may not be interested in having his/her identity disclosed. In certain cases, the disclosure of a patient's identity can leave it vulnerable to the action of attackers against his/her life, or to the disclosure of personal information. One of the requirements to proper functioning of e-health/m-health/TMIS and other systems for IoT is to reduce the consumption of computational and communication resources towards energy-savings and reduction of congestion in communication channels, given the large number of new emerging devices. Most devices destined to e-health/m-health and IoT are small, as sensors, and do not show high processing capacity and long battery life. Therefore, computational costs must be reduced for the optimization of power resources.

This work proposes a new cloud-based mutual authentication and key agreement protocol for e-health/TMIS systems focused on reduction of computational and communication resources consumption, if compared with other protocols proposed in the literature. The remainder of the paper is structured as follows: Section 2 describes some related works; Section 3 introduces the proposed protocol; Sections 4 and 5 address security and performance analyses, respectively; finally, Section 6 presents the conclusions.

2. Related Works

The works of Chiou et al. (2016) and Mohit et al. (2017) consider a cloud server as an auxiliary entity that stores data of patients, as the measures collected from sensors coupled to their body. Such data are encrypted and transmitted over public channels, from the entities involved to the cloud server and vice versa, after the execution of mutual authentication and generation of a session key.

Chiou et al. (2016) and Mohit et al. (2017) designed protocols based on asymmetric and symmetric cryptography and composed of four phases, namely health center upload (HUP), patient upload (PUP), treatment (TP) and checkup (CP). A security analysis conducted revealed some issues in the protocol of Chiou et al. (2016). According to Mohit et al. (2017), it fails to preserve the system anonymity and security if the patient's device is lost or stolen. On the other hand, the protocol of Mohit et al. (2017) fails to avoid the Denial of Service (DoS) attack.

Jiang and Lian et al. (2016) and Li et al. (2016) also developed interesting approaches. Although they do not consider an auxiliary cloud server, (the entities authenticate themselves directly with the health center server through the Internet), they are based on TMIS, similarly to the protocol proposed in this chapter. The proposal of Li et al. (2016) is based on asymmetric cryptography, whereas the one designed by Jiang and Lian et al. (2016) is based on symmetric cryptography. Both are composed of three phases in common, namely Initialization, Registration and Authentication. Li et al. (2016) accomplished all the security objectives considered in the security analysis section of this manuscript. However, the proposal of Jiang and Lian et al. (2016) is vulnerable to the loss/stealing of a patient's device and shows some lack of confidentiality.

The protocols of Jiang and Khan et al. (2016), Amin et al. (2016) and Shen et al. (2018) differ from those of Chiou et al. (2016) and Mohit et al. (2017) because they consider only the communication channel between the user (patient) and the cloud, i.e., they are not TMIS. They also use asymmetric cryptography based on Elliptic Curves Discrete Logarithm Problem (ECDLP) and comprise three phases, namely initialization, registration and login/authentication. Jiang and Khan et al. (2016) and Amin et al. (2016) accomplished all the security objectives analyzed in this study, however, the protocol of Shen et al. (2018) shows some security issues, as lack of confidentiality and vulnerability to patient trackabillity due to loss/stealing of the patient's mobile device.

Below are some aspects compared with the above-mentioned works:

d) the protocols of Chiou et al. (2016) and Mohit et al. (2017) inspired this protocol in some aspects as system architecture and phases, aiming at the development of a TMIS and cloud-based authentication protocol of higher security and performance.

e) the protocols of Chiou et al. (2016) and Mohit et al. (2017) are based on asymmetric cryptography, while our approach is based on symmetric cryptography, which guarantees lower computational and communication costs;

f) the security flaws of Chiou et al. (2016) and Mohit et al. (2017) are avoided in the protocol proposed in this chapter by the use of access control to the patient's device, timestamps, temporary identities and freshly generated parameters in each authentication session.

3. Proposed Protocol

The system’s architecture is the same as that developed by Chiou et al. (2016) and Mohit et al. (2017) (Fig. 1) and composed of five phases, namely registration, health center upload (HUP), patient upload (PUP), treatment (TP) and checkup (CP). The protocol is also based on symmetric cryptography and composed of the following trustful entities: health center, patient, cloud server and doctor. Table 1 shows the notations used.

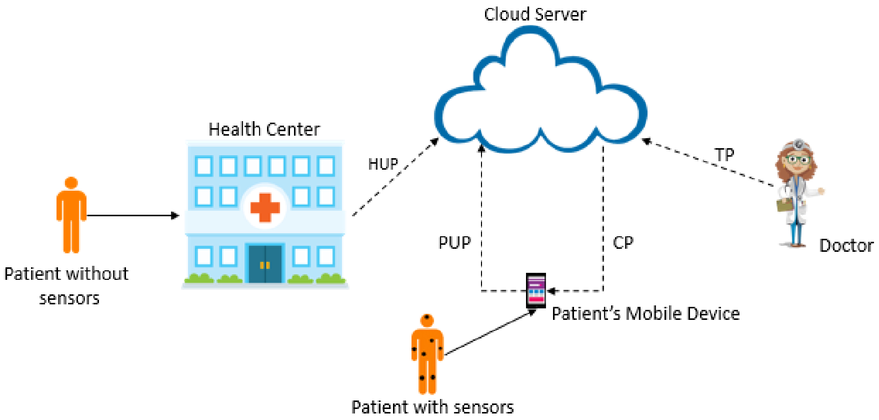


Figure 41. System’s architecture of the protocol.

The protocol is based on challenge-response and was developed as an alternative of secure and efficient mutual authentication scheme, without incurring in high computational and communication costs. The use of symmetric cryptography may generate security issues due to key exchanges over public channel. However, the proposed protocol does not exchange keys or real identities over insecure channel, as explained on sections 4.4 and 4.7, and consequently it is not affected by these problems.

3.1 Registration Phase

This phase is performed over secure channel and aims at registering the health center, patients and doctors in the cloud server. Each entity generates k different random numbers R_k and calculates a set of temporary identities, $TID_x = h_1(ID_x \| R_k)$, which are individually used at each authentication session initiated by the entities. The use of real identities associated with a random number in the calculation of temporary identities guarantees its uniqueness. They send their real identity ID_x and temporary identities TID_x to the cloud server, which stores the data to be used in the following phases. If all temporary identities of a certain entity are used, a new registration phase is performed. If a real identity is revoked, it is necessary to perform an especial registration phase, indicating which was the identity revoked and the new equivalent identity. Only registered entities can perform the following phases.

Table 1. Notations used in the protocol.

Symbol	Description
x, y	Entities: patient (P), health center (H), doctor (D), cloud server (C).
ID_x /TID_x	Real identity of entity x/ Temporary identity of entity x.
k	Random numbers generated in the registration phase.
R_k	k random number generated.
MACxy	Message Authentication Code generated from entity x to entity y.
R_x	Random number generated by entity x.
R_{Cy}	Random number generated by the cloud and sent to entity y.
T_x	Timestamp generated by entity x.
K_{xy}	Session key generated by entities x and y.
C_{xy}	Validator of the session key generated by x and y.
E_{Kxy} /D_{Kxy}	Encryption/Decryption operation that used the session key generated by x and
h_1	Temporary identity generation hash function.
h_2	MAC generation hash function.
h_3	Session key generation hash function.
h_4	Session key verifier generation hash function.
⟶	Secure channel.
- - - - -►	Insecure channel.

3.2 Health Center Upload Phase (HUP)

It is considered an insecure channel for this phase. Its aim is the mutual authentication among entities to allow secure transmission of the patient's collected data, from the health center to the cloud server. The complete procedure is shown in Figure 2. The HUP phase starts when the user goes to the health center for a health inspection and receives a login and a password to access the patient´s system in its mobile device. The patient can access his/her health information whenever wanted by inserting the login/password pair on his/her device. The health center stores the patient's temporary identity, TID_P, which is associated with the identity of its respective doctor.

Step 1. The health center selects a TID_H and generates a random number R_H. Then, it calculates $MAC_{HC} = h_2(ID_H \,||\, R_H)$ and sends to the cloud server *Message 1* = (TID_H, R_H, MAC_{HC}) with a timestamp T_H.

Step 2. After receiving *Message 1* and T_H from the health center, the cloud server verifies if T_H is valid. If the verification fails, the procedure is terminated. Otherwise, the cloud server calculates $MAC_{HC}' = h_2(ID_H \,||\, R_H)$ using the real identity of the health center received in the registration phase and the random number received in *Message 1*. It then verifies if $MAC_{HC}' = MAC_{HC}$. If the verification fails, the procedure ends because an intruder has been detected. Otherwise, the cloud server authenticates the health center, selects a random number R_{CH}, calculates $MAC_{CH} = h_2(ID_H \,||\, R_{CH})$ and sends *Message 2* = (MAC_{CH}, R_{CH}) with a timestamp T_C to the health center.

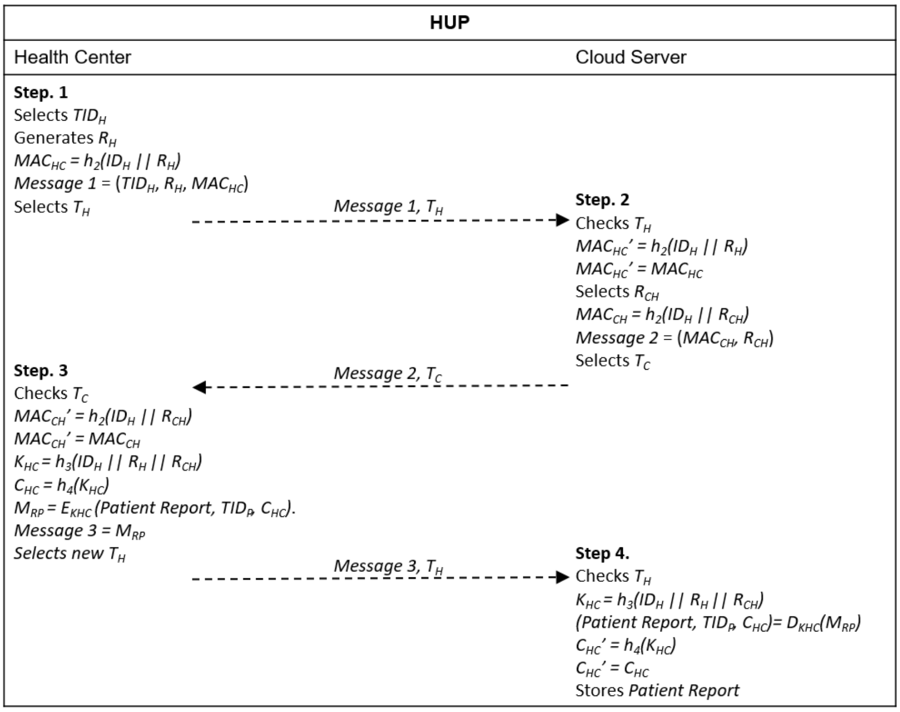

HUP
Health Center
Cloud Server
Step. 1
Selects TID_H
Generates R_H
$MAC_{HC} = h_2(ID_H \,||\, R_H)$
Message 1 = (TID_H, R_H, MAC_{HC})
Selects T_H
Message 1, T_H
Step. 2
Checks T_H
$MAC_{HC}' = h_2(ID_H \,||\, R_H)$
$MAC_{HC}' = MAC_{HC}$
Selects R_{CH}
$MAC_{CH} = h_2(ID_H \,||\, R_{CH})$
Message 2 = (MAC_{CH}, R_{CH})
Selects T_C
Step. 3
Message 2, T_C
Checks T_C
$MAC_{CH}' = h_2(ID_H \,||\, R_{CH})$
$MAC_{CH}' = MAC_{CH}$
$K_{HC} = h_3(ID_H \,||\, R_H \,||\, R_{CH})$
$C_{HC} = h_4(K_{HC})$
$M_{RP} = E_{KHC}$(Patient Report, TID_P, C_{HC}).
Message 3 = M_{RP}
Selects new T_H
Message 3, T_H
Step 4.
Checks T_H
$K_{HC} = h_3(ID_H \,||\, R_H \,||\, R_{CH})$
(Patient Report, TID_P, C_{HC})= $D_{KHC}(M_{RP})$
$C_{HC}' = h_4(K_{HC})$
$C_{HC}' = C_{HC}$
Stores Patient Report

Figure 2. Message exchange in HUP.

Step 3. The health center receives *Message 2* and T_C from the cloud server and checks if timestamp T_C is valid. If the validation fails, the procedure ends. Otherwise, the health center calculates $MAC_{CH}' = h_2(ID_H \,||\, R_{CH})$ and verifies if $MAC_{CH}' = MAC_{CH}$. If the verification fails, the procedure is terminated because an intruder has been detected. Otherwise, the health center authenticates the cloud server and generates the session key, $K_{HC} = h_3(ID_H \,||\, R_H \,||\, R_{CH})$ and the session key validator, $C_{HC} = h_4(K_{HC})$. It then uses the session key to encrypt the patient's report, $M_{RP} = E_{KHC}$ *(Patient Report,* TID_P, C_{HC}*)* and finally sends *Message 3* = M_{RP} and a new timestamp T_H to the cloud server.

Step 4. The cloud server receives {Message 3, T_H} and verifies T_H. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{HC} = h_3(ID_H \,||\, R_H \,||\, R_{CH})$ and decrypts the patient's report, *(Patient Report,* TID_P, C_{HC}*)* $= D_{KHC}(M_{RP})$. It then calculates

$C_{HC} = h_4(K_{HC})$ and verifies if $C_{HC}' = C_{HC}$. If the verification fails, it ends the procedure. Finally, the cloud server stores the patient´s report with the respective identities.

3.3 Patient Upload Phase (PUP)

The PUP phase is performed over an insecure channel. The focus of PUP is the mutual authentication between the patient and the cloud server and the generation of a session key to encrypt health information measured by the sensors attached to the user's body, prior to send it to the cloud server. The complete procedure is shown in Figure 3. The PUP phase starts when the patient's device requests, to the sensors attached to user's body, the health information measures collected and stores them.

Step 1. The patient selects one of his/her temporary identities TID_P, generates a random number R_P, calculates $MAC_{PC} = h_2(ID_P \| R_P)$ and sends $Message\ 1 = (TID_P, R_P, MAC_{PC})$ with a timestamp T_P to the cloud server.

Step 2. The cloud server receives *Message 1* and T_P and verifies if T_P is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{PC}' = h_2(ID_P \| R_P)$ and verifies if $MAC_{PC}' = MAC_{PC}$. If the verification fails, the procedure is interrupted. Otherwise, the cloud server authenticates the patient, selects a random number R_{CP}, calculates $MAC_{CP} = h_2(ID_P \| R_{CP})$ and sends $Message\ 2 = (MAC_{CP}, R_{CP})$ with a timestamp T_C to the patient.

Step 3. After receiving *Message 2* and T_C from the cloud server, the patient checks if T_C is valid. If the validation fails, the procedure ends. Otherwise, it calculates $MAC_{CP}' = h_2(ID_P \| R_{CP})$ and verifies if $MAC_{CP}' = MAC_{CP}$. If the verification fails, the procedure is terminated. Otherwise, the patient authenticates the cloud server, generates the session key $K_{PC} = h_3(ID_P \| R_P \| R_{CP})$ and calculates $C_{PC} = h_4(K_{PC})$. He/she then encrypts the sensors measures using the session key, $M_{MS} = E_{KPC}\,(Sensors\ Measures, TID_P, C_{PC})$ and sends $Message\ 3 = M_{MS}$ with a new timestamp T_P to the cloud server.

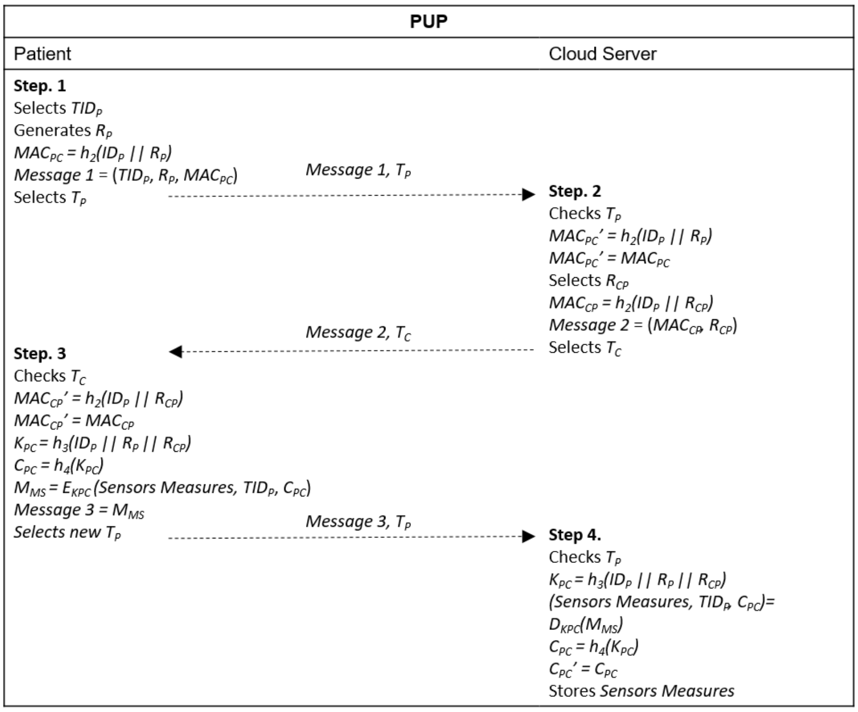

PUP
Patient
Cloud Server
Step. 1
Selects TID_P
Generates R_P
$MAC_{PC} = h_2(ID_P \mid\mid R_P)$
$Message\ 1 = (TID_P, R_P, MAC_{PC})$
Selects T_P
$Message\ 1, T_P$
Step. 2
Checks T_P
$MAC_{PC}' = h_2(ID_P \mid\mid R_P)$
$MAC_{PC}' = MAC_{PC}$
Selects R_{CP}
$MAC_{CP} = h_2(ID_P \mid\mid R_{CP})$
$Message\ 2 = (MAC_{CP}, R_{CP})$
Selects T_C
$Message\ 2, T_C$
Step. 3
Checks T_C
$MAC_{CP}' = h_2(ID_P \mid\mid R_{CP})$
$MAC_{CP}' = MAC_{CP}$
$K_{PC} = h_3(ID_P \mid\mid R_P \mid\mid R_{CP})$
$C_{PC} = h_4(K_{PC})$
$M_{MS} = E_{KPC}(Sensors\ Measures, TID_P, C_{PC})$
$Message\ 3 = M_{MS}$
Selects new T_P
$Message\ 3, T_P$
Step 4.
Checks T_P
$K_{PC} = h_3(ID_P \mid\mid R_P \mid\mid R_{CP})$
$(Sensors\ Measures, TID_P, C_{PC}) = D_{KPC}(M_{MS})$
$C_{PC} = h_4(K_{PC})$
$C_{PC}' = C_{PC}$
Stores *Sensors Measures*

Figure 3. Message exchange in PUP.

Step 4. The cloud server receives {*Message 3*, T_P} and verifies if T_P is valid. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{PC} = h_3(ID_P \mid\mid R_P \mid\mid R_{CP})$, decrypts the sensors measures, $(Sensors\ Measures, TID_P, C_{PC}) = D_{KPC}(M_{MS})$,calculates $C_{PC} = h_4(K_{CP})$ and verifies if $C_{PC}' = C_{PC}$. If the verification fails, it terminates the procedure. Otherwise, it stores the sensors measures with the respective identities.

3.4 Treatment Phase (TP)

This phase is performed over an insecure channel and aims at a mutual authentication between the doctor and the cloud server and generation of a session key for encrypting the patient's health report and sensors measures before they are sent to the doctor, and encrypting the doctor's diagnosis before it is sent to the cloud server. The complete procedure is shown in Figure 4.

Step 1. The doctor selects one of his/her temporary identities $TID_{D,}$ generates a random number R_D, calculates $MAC_{DC} = h_2(ID_D \mid\mid R_D)$ and sends *Message 1* = (TID_D, R_D, MAC_{DC}) with a timestamp T_D to the cloud server.

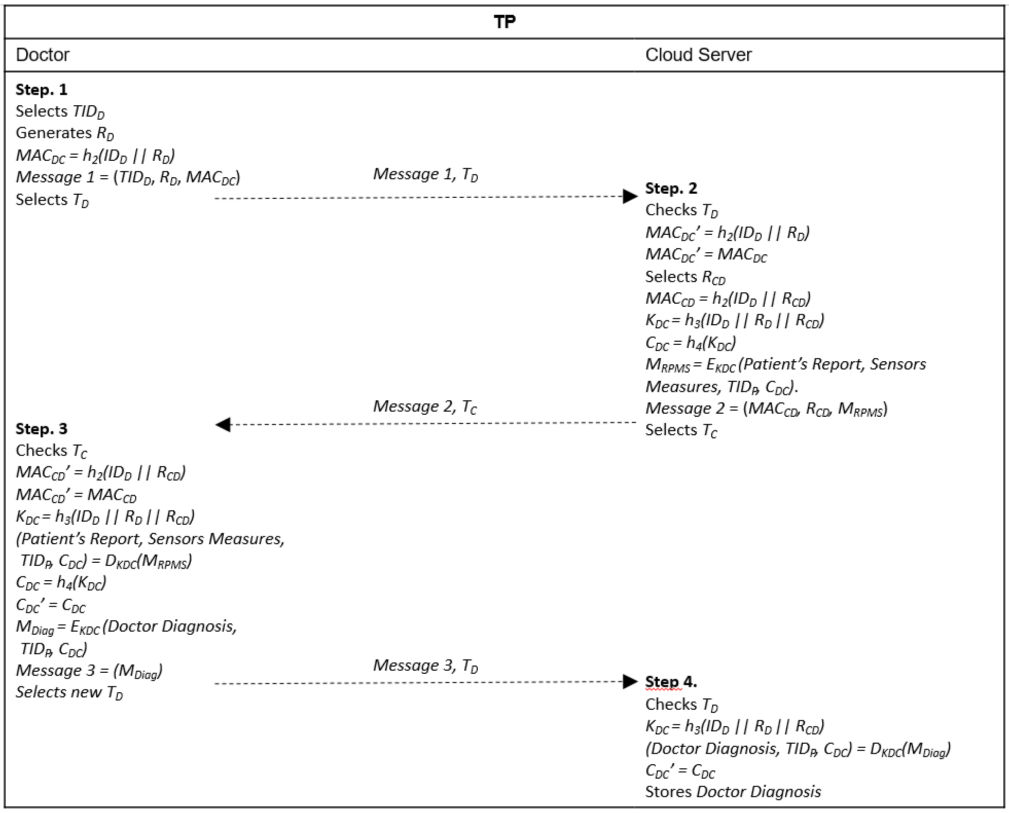

TP
Doctor
Cloud Server
Step. 1
Selects TID_D
Generates R_D
$MAC_{DC} = h_2(ID_D \,||\, R_D)$
Message 1 = (TID_D, R_D, MAC_{DC})
Selects T_D
Message 1, T_D
Step. 2
Checks T_D
$MAC_{DC}' = h_2(ID_D \,||\, R_D)$
$MAC_{DC}' = MAC_{DC}$
Selects R_{CD}
$MAC_{CD} = h_2(ID_D \,||\, R_{CD})$
$K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$
$C_{DC} = h_4(K_{DC})$
$M_{RPMS} = E_{KDC}$(Patient's Report, Sensors Measures, TID_P, C_{DC}).
Message 2 = $(MAC_{CD}, R_{CD}, M_{RPMS})$
Selects T_C
Message 2, T_C
Step. 3
Checks T_C
$MAC_{CD}' = h_2(ID_D \,||\, R_{CD})$
$MAC_{CD}' = MAC_{CD}$
$K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$
(Patient's Report, Sensors Measures, TID_P, C_{DC}) = $D_{KDC}(M_{RPMS})$
$C_{DC} = h_4(K_{DC})$
$C_{DC}' = C_{DC}$
$M_{Diag} = E_{KDC}$(Doctor Diagnosis, TID_P, C_{DC})
Message 3 = (M_{Diag})
Selects new T_D
Message 3, T_D
Step 4.
Checks T_D
$K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$
(Doctor Diagnosis, TID_P, C_{DC}) = $D_{KDC}(M_{Diag})$
$C_{DC}' = C_{DC}$
Stores Doctor Diagnosis

Figure 4. Message exchange in TP.

Step 2. The cloud server receives {*Message* 1, T_D} and verifies if T_D is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{DC}' = h_2(ID_D \,||\, R_D)$ and verifies if $MAC_{DC}' = MAC_{DC}$. If the verification fails, the procedure is interrupted. Otherwise, the cloud server authenticates the doctor, selects a random number R_{CD} and calculates $MAC_{CD} = h_2(ID_D \,||\, R_{CD})$, a session key $K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$ and $C_{DC} = h_4(K_{DC})$. It then uses the doctor's real identity to obtain the patient´s report and sensors health information measures previously stored in the cloud and prepares the information to be sent to the doctor, encrypting the data with the session key calculated, $M_{RPMS} = E_{KHC}$ *(Patient Report, Sensors Measures,* TID_P, C_{DC}*)*. Finally, it sends *Message 2* = (MAC_{CD}, R_{CD}, M_{RPMS}) with a timestamp T_C to the doctor.

Step 3. The doctor receives {*Message* 2, T_C} and checks if T_C is valid. If the validation fails, the procedure ends. Otherwise, the health center calculates $MAC_{CD}' = h_2(ID_D \,||\, R_{CD})$ and verifies if $MAC_{CD}' = MAC_{CD}$. If the verification fails, the procedure is terminated. Otherwise, the doctor authenticates the cloud server, generates the session key $K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$, decrypts M_{RPMS} to obtain the patient's report and the health information measured by the sensors, *(Patient's Report, Sensors Measures,* TID_P, C_{DC}*)* $= D_{KDC}(M_{RPMS})$, calculates $C_{DC}' = h_4(K_{DC})$ and verifies if $C_{DC}' = C_{DC}$. Then, he/she analyzes the data received, generates the patient's diagnosis, encrypts it, M_{Diag} = E_{KDC} *(Doctor Diagnosis, TID_P)* and finally sends *Message 3* = M_{Diag} and a new timestamp T_D to the cloud server.

Step 4. After receiving *Message 3* and $T_{D,}$ the cloud server verifies if T_D is valid. If the verification fails, it terminates the procedure. Otherwise, it calculates the session key $K_{DC} = h_3(ID_D \,||\, R_D \,||\, R_{CD})$, $C_{DC}' = h_4(K_{DC})$ and verifies if $C_{DC}' = C_{DC}$. If the verification fails, it interrupts the procedure because the message was not originated from the authenticated doctor and might have been forged by an intruder. If the verification succeeds, the cloud server uses the session key to decrypt the doctor's diagnosis and its respective temporary identity, *(Doctor Diagnosis, TID_D)* = $D_{KDC}(M_{Diag})$. Finally, it stores the doctor's diagnosis with its respective identities.

3.5 Checkup Phase (CP)

This phase is performed over an insecure channel and aims at a new mutual authentication between the patient and the cloud server and generation of a new session key for encrypting the doctor's diagnosis, before the cloud sends it to the patient. The complete procedure is shown in Figure 5.

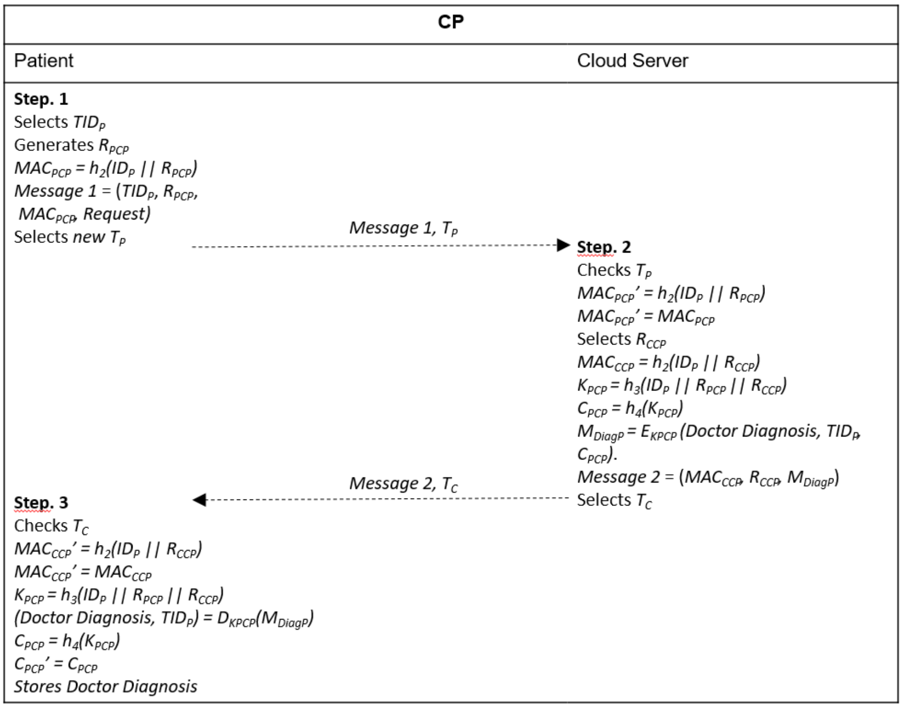

CP
Patient
Cloud Server
Step. 1
Selects TID_P
Generates R_{PCP}
$MAC_{PCP} = h_2(ID_P \,||\, R_{PCP})$
Message 1 = (TID_P, R_{PCP}, MAC_{PCP}, Request)
Selects new T_P
Message 1, T_P
Step. 2
Checks T_P
$MAC_{PCP}' = h_2(ID_P \,||\, R_{PCP})$
$MAC_{PCP}' = MAC_{PCP}$
Selects R_{CCP}
$MAC_{CCP} = h_2(ID_P \,||\, R_{CCP})$
$K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$
$C_{PCP} = h_4(K_{PCP})$
$M_{DiagP} = E_{KPCP}$(Doctor Diagnosis, TID_P, C_{PCP}).
Message 2 = (MAC_{CCP}, R_{CCP}, M_{DiagP})
Selects T_C
Message 2, T_C
Step. 3
Checks T_C
$MAC_{CCP}' = h_2(ID_P \,||\, R_{CCP})$
$MAC_{CCP}' = MAC_{CCP}$
$K_{PCP} = h_3(ID_P \,||\, R_{PCP} \,||\, R_{CCP})$
(Doctor Diagnosis, TID_P) = $D_{KPCP}(M_{DiagP})$
$C_{PCP} = h_4(K_{PCP})$
$C_{PCP}' = C_{PCP}$
Stores Doctor Diagnosis

Figure 5. Message exchange in CP.

Step 1. The patient generates a new random number R_{PCP}, calculates $MAC_{PCP} = h_2(ID_P \,||\, R_{PCP})$ and sends *Message 1* = (TID_P, R_{PCP}, MAC_{PCP}, *Request*) with a timestamp T_P to the cloud server.

Step 2. After receiving *Message* 1 and T_P, the cloud server verifies if T_P is valid. If the verification fails, the procedure is terminated. Otherwise, it calculates $MAC_{PCP}' = h_2(ID_P \,||$

R_{PCP}) and verifies if *$MAC_{PCP}' = MAC_{PCP}$*. If the verification fails, the procedure ends. Otherwise, it authenticates the patient, selects a random number *R_{CCP}*, calculates *$MAC_{CCP} = h_2(ID_P \| R_{CCP})$*, generates the session key *$K_{PCP} = h_3(ID_P \| R_{PCP} \| R_{CCP})$* and computes *$C_{PCP} = h_4(K_{PCP})$*. It then uses the session key to encrypt the doctor's diagnosis, *$M_{DiagP} = E_{KPCP}$ (Doctor's Diagnosis, TID_P, C_{PCP})* and sends to the patient *Message 2* = (*MAC_{CCP}*, *R_{CCP}*, *M_{DiagP}*) with a timestamp *T_C*.

Step 3. The patient receives {*Message* 2, *T_C*} and checks if *T_C* is valid. If the validation fails, the procedure is terminated. Otherwise, he/she calculates *$MAC_{CCP}' = h_2(ID_P \| R_{CCP})$* and verifies if *$MAC_{CCP}' = MAC_{CCP}$*. If the verification fails, the procedure is interrupted. Otherwise, he/she authenticates the cloud server, generates the session key *$K_{PCP} = h_3(ID_P \| R_{PCP} \| R_{CCP})$*, decrypts the doctor's diagnosis, *(Doctor's Diagnosis, TID_P, C_{PCP}) = $D_{KPCP}(M_{DiagP})$*, calculates *$C_{PCP} = h_4(K_{PCP})$* and verifies if *$C_{PCP}' = C_{PCP}$.* If the verification fails, it ends the procedure. Otherwise, the patient stores the doctor's diagnosis and looks for a convenient treatment.

4. Security Analysis

This section presents the security objectives accomplished by the protocol. Table 2 shows a security comparison between the proposed protocol and those designed by Choi et al. (2016) and Mohit et al. (2017).

4.1. Mutual Authentication

In the protocol proposed in this chapter, each entity calculates a MAC to perform mutual authentication with the cloud server and vice versa. For example, in the HUP phase, the health center calculates $MAC_{HC} = h_2(ID_H \| R_H)$ and sends it to the server cloud, which calculates $MAC_{HC}' = h_2(ID_H \| R_H)$ and verifies if $MAC_{HC}' = MAC_{HC}$. If the verification is successful, the server cloud authenticates the health center, calculates its own $MAC_{CH} = h_2(ID_H \| R_{CH})$ and sends it to the health center, which calculates $MAC_{CH}' = h_2(ID_H \| R_{CH})$ and verifies if $MAC_{CH}' = MAC_{CH}$. If the verification succeeds, the health center authenticates the server cloud and the mutual authentication procedure is complete. A similar procedure is performed in the PUP, TP and CP phases.

4.2 Forward/Backward Secrecy

The forward and backward secrecies are guaranteed by the use of random values (R_H, R_{CH}, R_P, R_{CP}, R_D, R_{CD}, R_{PC}, R_{CPC}) newly generated in each authentication session, during the calculation of the system keys, as the one generated in the PUP phase $K_{CP} = h_3(ID_P \| R_P \| R_C)$. Therefore, if an intruder discovers old system keys, it cannot use them in future authentication sessions (backward secrecy). On the other hand, if an intruder discovers future system keys, it cannot use them in past authentication sessions (forward secrecy).

4.3 Confidentiality

The system´s confidentiality is guaranteed by the access control of the patient’s mobile device. A possible user must insert login and password to access his/her information in the system. Consequently, sensitive information is available only to authorized users. An authentication procedure is performed between the cloud and an entity in each phase for the generation of a session key that will encrypt the patient’s data before it is exchanged on a public channel.

4.4 Non-Repudiation

At the beginning of each phase in the protocol, the entities send the cloud their temporary identities (TID_H, TID_P, TID_D) and a MAC calculated with their real identities (ID_H, ID_P, ID_D). The cloud also sends to the entities a MAC containing their real identities. Since real identities are known only by the cloud and each respective entity, a valid MAC can be generated only by them. The session keys established among the cloud and the entities also depend on their real identity, therefore, neither the cloud, nor the entities can deny the message they originated.

4.5 Anonymity

Anonymity is assured only by entities’ temporary identities (TID_H, TID_P, TID_D), while messages are exchanged on an insecure channel during the authentication procedure, which protects their real identities. The identity of the cloud server is protected because it is not used in the authentication procedure, hence, not exchanged on an insecure channel.

4.6 Non-Traceability

The use of different temporary identities and newly generated random numbers in each new authentication session generates different parameters exchanged. Therefore, outsiders cannot track patients by the parameters exchanged on a public channel.

4.7 Session Key Security

Session keys are not exchanged on a public channel, but securely calculated on each side involved in the authentication. Moreover, the security of the session keys established at each phase of the protocol is guaranteed through the use of entities’ real identities in the calculation, some secret information known only by the cloud server and the respective entities. For example, in HUP, the session key calculated is $KHC = h_3(IDH \parallel RH \parallel RCH)$, consequently, an intruder cannot obtain or calculate a valid session key.

4.8 Patient’s mobile device loss/stealing

The security objective is accomplished through the access control of the patient’s mobile device using login and password. The system is only accessible if a valid login and password pair is inserted. If the mobile device is stolen or lost, no unauthorized person can access the patient’s system, because it would not have a valid login and password pair.

4.9 Impersonation Attack

The impersonation attack is avoided because neither the cloud server’s real identity, nor the entities’ real identities are disclosed. Therefore, an attacker cannot impersonate them and generate a valid MAC, because its calculation depends on the entities’ real identities.

4.10 Replay Attack

The replay attack is avoided because all entities involved in the protocol proposed in this chapter use different random values freshly calculated in each authentication process. Therefore, an attacker cannot forge messages using old random values.

4.11 Denial of Service (DoS)

The prevention of this attack involves the inclusion of a verification parameter in each message exchanged in the authentication phases (HUP, PUP, TP, CP). The verification parameter used in the protocol proposed in this chapter was a timestamp and its validity was verified before the recipient processed each message. Therefore, if an attacker uses an invalid timestamp, the entire procedure is interrupted in time to prevent the DoS attack.

4.12 Man-in-the-Middle Attack

No intruder can perform a man-in-the-middle attack, because the session key cannot be forged with the use of only the parameters exchanged on the insecure communication channel. The session key calculation uses the entities’ real identities, which is a secret value not disclosed in the insecure channel.

According to Table 2, the protocol designed by Chiou et al. (2016) does not guarantee anonymity, non-traceability and resistance to patient’s mobile device loss/stealing, which are three critical failures. First, as detected by Mohit et al. (2017), in the protocol of Chiou et al. (2016), the patient’s real identity is sent in plain text through a public channel, which compromises its anonymity. We observed it also affects the patient’s non-traceability. Second, as detected by Mohit et al. (2017), the proposal of Chiou et al. (2016) fails to be resistant to patient’s mobile device loss/stealing, because it does not perform access control and requests login and password to the user, which makes the system vulnerable to the access of non-authorized people and hampers its confidentiality.

Table 2. Comparison of security objectives among protocols

Security Objectives	Chiou et al.	Mohit et al.	The protocol
Mutual Authentication	Yes	Yes	Yes
Forward/Backward Secrecy	Yes	Yes	Yes
Confidentiality	No	Yes	Yes
Non-Repudiation	Yes	Yes	Yes
Anonymity	No	Yes	Yes
Patient's Non-Traceability	No	Yes	Yes
Session Key Security	Yes	Yes	Yes
Resistance to patient's mobile device	No	Yes	Yes
Resistance to Impersonation attack	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes
Resistance to Denial of Service (DoS)	Yes	No	Yes
Resistance to man-in-the-middle attack	Yes	Yes	Yes

The protocol of Mohit et al. (2017) fails to prevent DoS attack. During the phases, no initial verification parameter is generated (timestamp, nonce, sequence number) or exchanged, and its validity is not verified before the recipient processes each message. Therefore, the protocol is vulnerable to DoS attacks. The protocol proposed in this chapter accomplished all security objectives analyzed and can, therefore, be considered safer than those designed by Chiou et al. (2016) and Mohit et al. (2017).

5. Performance Analysis

This section addresses a performance analysis of the protocol proposed in this chapter and a comparison with those developed by Chiou et al. (2016) and Mohit et al. (2017). The analysis evaluated and compared the computational and communication costs. The registration phase of the protocol was not included in the analysis because it is performed over a secure channel and the focus of the comparisons was on operations executed and parameters exchanged over an insecure channel.

5.1 Computational Cost

The execution time in seconds (s) of the operations considered is shown in Table 3. Chiou et al. (2016) and Mohit et al. (2017) adopted those values and performed tests with the following operational characteristics: CPU: Intel (R) Core (TM) 2 Quad Q8300, 2.50Hz; memory: 2GB; operational system: Windows 7 Professional.

Table 3. Execution time of each operation considered.

Symbol	Description	Cost
T_S	**Execute/Verify a Signature**	**0.3317s**
T_P	**Bilinear Pairing**	**0,0621s**
T_E	**Encrypt/Decrypt (Symmetric)**	**0.0087s**
T_H	**One Way Hash Function**	**0.0005s**

All the four phases were analyzed and all operations executed were considered. Table 4 shows a comparison of the computational costs among the protocol proposed in this chapter and those of Chiou et al. (2016), Mohit et al. (2017), details of the operations performed at each phase and the total time in seconds.

Table 4. Computational Cost of Protocols

	Chiou et al. (2016)	Mohit et al. (2017)	Proposed Protocol
HUP	$T_S + 3T_P + 2T_E + 7T_H$	$T_S + 3T_E + 11T_H$	$2T_E + 8T_H$
PUP	$T_S + 3T_P + 2T_E + 9T_H$	$2T_S + 2T_E + 10T_H$	$4T_E + 8T_H$
TP	$2T_S + 3T_P + 2T_E + 8T_H$	$2T_S + 2T_E + 9T_H$	$4T_E + 8T_H$
CP	$T_S + 2T_P + 2T_E + 8T_H$	$T_S + 2T_E + 5T_H$	$2T_E + 8T_H$
TOTAL	$5T_S + 11T_P + 8T_E + 32T_H$ = 2.43s	$4T_S + 9T_E + 35T_H$ = 1.42s	$12T_E + 32T_H$ = 1.2s

The protocol proposed in this chapter required the lowest computational cost, therefore, it performs the operations necessary in shorter time and offers the best computational cost, due to the exclusive use of symmetric criptografy (low communication cost) for the authentication procedures. Chiou et al. (2016) and Mohit et al. (2017) conducted some signature operations and bilinear pairing, which incurred in higher computational costs.

5.2 Communication Cost

The evaluation of the communication costs considered messages exchanged over an insecure channel and parameters and their respective costs in bits (see Table 5).

Table 5. Size of each parameter in bits.

Parameter	Cost
Random Number/Identity/Timestamp	**48 bits**
Bilinear Pairing/Hash	**160bits**
Symmetric Key	**128 bits**
Signature (symmetric algorithm)	**512 bits**

The message exchange over an insecure channel was analyzed in each of the four common phases performed by the protocol proposed in this chapter and those of Chiou et al. (2016) and Mohit et al. (2017). Table 6 shows comparisons of each phase and a comparison of the total communication cost of each protocol.

Table 6. Comparison of communication costs in bits.

	Chiou et al. (2016)	Mohit et al. (2017)	Proposed Protocol
HUP	704	592	736
PUP	1600	1744	736
TP	2112	1792	864
CP	1504	1184	736
TOTAL	6920 bits	4832 bits	3072 bits

The protocol proposed in this chapter required the lowest communication cost, hence, the best communication cost, due to the reduced number of parameters exchanged and choice of small parameters to be exchanged (identities, random numbers, timestamps). The proposals of Chiou et al. (2016) and Mohit et al. (2017) required higher communication costs, because of the exchange of some costly signature parameters. The protocol proposed in this chapter achieved the best performance, revealed by security and performance analyses.

6. Conclusions

The application of e-health/m-health to the monitoring, diagnosis and treatment of patients speeds up the provision of medical services. In many cases, the patient does not need to leave his/her home for a doctor´s appointment, which facilitates the access to medical advice for patients with limited mobility, the elderly or patients located in hard access areas.

The protocols analyzed showed interest in the development of efficient and safe e-health/m-health/TMIS systems for protecting patient’s data and their respective identities. The protocol proposed in this chapter showed suitable to TMIS and overperformed those of Chiou et al. (2016) and Mohit et al. (2017). The protocol designed by Chiou et al. (2016) does not control the access to patients’ mobile devices for avoiding their system´s exposure to intruders, if the device is lost or stolen, which is a problem with simple solution.

Furthermore, reductions in computational and communication costs are reinforced. Asymmetric cryptography is considered safer than symmetric cryptography, however, it demands more resource consumption than symmetric cryptography. The performance and security analyses conducted confirmed that resource consumption can be reduced with no impact on the system’s security through the use of symmetric cryptography, as explained in sections 4.4 and 4.7.

Future studies include a formal verification of the protocol, storage cost analysis and comparison with related works and development of other mutual authentication protocols based on asymmetric cryptography for cloud-based e-health systems that accomplish more security objectives with reduced resource consumption. They also aim at the development of authentication and authorization protocols, considering cooperation strategies for better confidentiality and integrity in m-Health systems (Silva et al. (2014), as well as security evaluation based on integrated systems of ambient assisted living (AAL) and e-health (as in Rghioui et al. (2016)).

References

Chiou, S., Ying, Z. and Liu, J., (2016) "Improvement of a privacy authentication scheme based on cloud for medical environment.”, In *Journal of medical systems*, v. 40, n. 4, p.101.

Mohit, P., Amin, R., Karati, A., Biswas, G. P., Khan, M. K., (2017) "A standard mutual authentication protocol for cloud computing based health care system.", *Journal of medical systems*, v.41, n. 4, p. 50.

Jiang, Q., Khan, K. M., Lu, X., Ma, J. and He, D., (2016) “A privacy preserving three-factor authentication protocol for e-health clouds.” In *The Journal of Supercomputing*, v. 72, n. 10, p. 3826–3849.

Jiang, Q., Lian, X., Yang, Chao., Ma, J., Tian, Y. and Yang, Y., (2016) “A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth.”, In *Journal of medical systems*, v. 40, n. 11, p. 4650–4666.

Li, X., Niu, J., Karuppiah, M., Kumari, S. and Wu, F., (2016) “Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications.”, In *Journal of medical systems*, v. 40, n. 12, p. 268.

Amin, R., Islam, S. K., Biswas, G. P., Giri, D., Khan, K. M. and Kumar, N., (2016“A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments.”, In *Security and Communication Networks*, v. 9, n. 17, p. 4650–4666.

Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H. and Tang, Y., (2018), “Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks.”, In *Journal of Network and Computer Applications*, v. 106, p. 117-123.

Rghioui, A., Sendra, S., Lloret, J., Oumnad, A., (2016) “Internet of things for measuring human activities in ambient assisted living and e-health”, In *Network Protocols and Algorithms,* v.8, n. 3, p. 15-28.

Silva, BMC., Rodrigues, JJPC., Canelo, F., Lopes, IMC., Lloret, J., (2014) “Towards a cooperative security system for mobile-health applications”, In *Electronic Commerce Research*, p. 1-27.

APPENDIX 2 – Papers submitted to journals

Paper of chapter 3: Ana Paula G. Lopes, Paulo R. L. Gondim, "MUTUAL AUTHENTICATION PROTOCOL FOR D2D COMMUNICATIONS IN A CLOUD-BASED E-HEALTH SYSTEM" - submitted to an international journal.

Paper of chapter 5: Ana Paula G. Lopes, Paulo R. L. Gondim, "A LIGHTWEIGHT AUTHENTICATION SCHEME FOR D2D COMMUNICATION IN M-HEALTH WITH TRUST EVALUATION" - submitted to an international journal.

APPENDIX 3 – Paper accepted for publication in journal

Paper of chapter 4: Ana Paula G. Lopes, Paulo R. L. Gondim, "GROUP AUTHENTICATION PROTOCOL BASED ON AGGREGATED SIGNATURES FOR D2D COMMUNICATION" - submitted to Computer Networks Journal (manuscript nr. COMNET_2019_1135);

Acceptance letter:

Ref: COMNET_2019_1135
Title: Group Authentication Protocol Based on Aggregated Signatures for D2D Communication
Journal: Computer Networks
Dear Professor. Gondim,

Thank you for submitting your manuscript to Computer Networks. I have received comments from reviewers on your manuscript. Your paper should become **acceptable for publication pending suitable minor revision** and modification of the article in light of the appended reviewer comments.

When resubmitting your manuscript, please carefully consider all issues mentioned in the reviewers' comments, outline every change made point by point, and provide suitable rebuttals for any comments not addressed.

To submit your revised manuscript:

- Log into EVISE® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=COMNET
- Locate your manuscript under the header 'My Submissions that need Revisions' on your 'My Author Tasks' view
- Click on 'Agree to Revise'
- Make the required edits
- Click on 'Complete Submission' to approve

What happens next?

After you approve your submission preview you will receive a notification that the submission is complete. To track the status of your paper throughout the editorial process, log in to EVISE® at: http://www.evise.com/evise/faces/pages/navigation/NavController.jspx?JRNL_ACR=COMNET.

Enrich your article to present your research with maximum impact. This journal supports the following Content Innovations:

Data in Brief (optional):

We invite you to convert your supplementary data (or a part of it) into an additional journal publication in Data in Brief, a multi-disciplinary open access journal. Data in Brief articles are a fantastic way to describe supplementary data and associated metadata, or full raw datasets deposited in an external repository, which are otherwise unnoticed. A Data in Brief article (which will be reviewed, formatted, indexed, and given a DOI) will make your data easier to find, reproduce, and cite.

You can submit to Data in Brief via the Computer Networks submission system when you upload your revised Computer Networks manuscript. To do so, complete the template and follow the co-submission instructions found here: www.elsevier.com/dib-template. If your Computer Networks manuscript is accepted, your Data in Brief submission will automatically be transferred to Data in Brief for editorial review and publication.

Please note: an open access Article Publication Charge (APC) is payable by the author or research funder to cover the costs associated with publication in Data in Brief and ensure your data article is immediately and permanently free to access by all. For the current APC see: https://www.elsevier.com/journals/data-in-brief/2352-3409/open-access-journal

Please contact the Data in Brief editorial office at dib-me@elsevier.com or visit the Data in Brief homepage (www.journals.elsevier.com/data-in-brief/) if you have questions or need further information.

I look forward to receiving your revised manuscript as soon as possible.

Kind regards,
Dr. Rudin
Special Issue Editor-in-Chief
Computer Networks

Comments from the editors and reviewers:
-Reviewer 1

The manuscript addresses the design and performance evaluation of an authentication protocol for Device-to-Device (D2D) Communications. It is devoted to groups of devices, and is based on aggregated signatures.

The paper is well organized and adequate for the mission and scope of the journal. Its applicability involves a broad set of applications and scenarios based on D2D communications, as an important characteristic to be provided by 5G networks. However, the text needs a review in terms of use of the English language.

In the Introduction section, the authors highlight the Communicating Things Networks (CTNs) paradigm and the characteristics of D2D communications, and describe the objectives and contributions of the paper.

In the second section, the set of references should be extended, in order to include more references about group authentication protocols for D2D-based communications systems, thus allowing discussion on the characteristics of the proposed protocol and to reflect more deeply on the SotA. Moreover, in Table 1 there are two columns which could be better explained and differentiated (Group Authentication and Group Management).

In the third section, the system architecture and some basic assumptions are described, followed by the description of the different phases of the protocol. It is suggested that the authors describe leader election and device discovery processes and-or indicate references for the interested readers.

In the fourth section, the major part of the set of security objectives is adequately treated. However, confidentiality and privacy objectives should be better discussed and more adequately justified.

In the fifth section, a performance analysis is made, considering a comparison with 4 (four) other authentication protocols, with calculation of communication, computing and energy costs. Figures 5, 6 and 7 allow discussion on the main results of the work, and show the better performance of the proposed protocol. As a possible improvement of the paper, the storage costs should be evaluated, especially considering aspects related to memory availability of D2D devices.

In the sixth section and in the Appendix, the security analysis is complimented by semi-formal verification using AVISPA tool and some backends.

In the Conclusions (seventh section), the authors discuss the main results of the work, which are adequately supported by the research.

-Reviewer2

The paper presents a group authentication protocol tailored for D2D communications. It is well written, well organized and interesting.

My only concern is the lacking of details in some parts. I mean, in my opinion, when the proposed solution leverages on another work, such as the binary tree used for group management, the authors should provide more details, a simple reference to that work is not enough.

Similarly, I would suggest providing some insight into the leader election procedure which is a bit unclear. How it is performed? Authors say: "Leader election and group organization is performed offline". How? I think it could be beneficial, especially because it is one of the characteristics which differentiates the proposed solution from available solutions as shown by authors in Table 1.

Furthermore, I think a more detailed presentation of the Local Area Identification of eNB is needed, in particular, because it is one of the key features adopted to provide resistance to the redirection attack.

Finally, in the performance evaluation section, the communication cost and the energy cost should be better described. I mean, Table 7 and Table 8 present the aggregated costs, however, I think a detailed presentation, as in Table 5 for computational costs, could be beneficial.