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 our protocol. 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:

a) the protocols of Chiou et al. (2016) and Mohit et al. (2017) inspired our work 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.

b) 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;

c) the security flaws of Chiou et al. (2016) and Mohit et al. (2017) are avoided in our protocol 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.

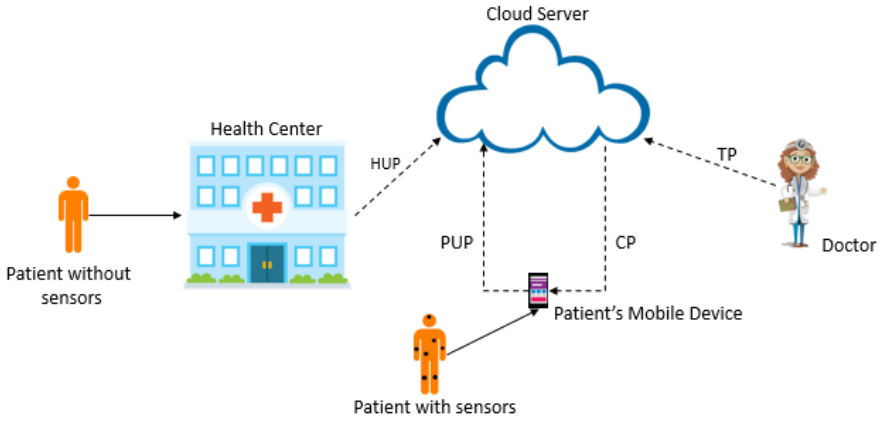

Figure1. System's architecture of the protocol.

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 y.
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.

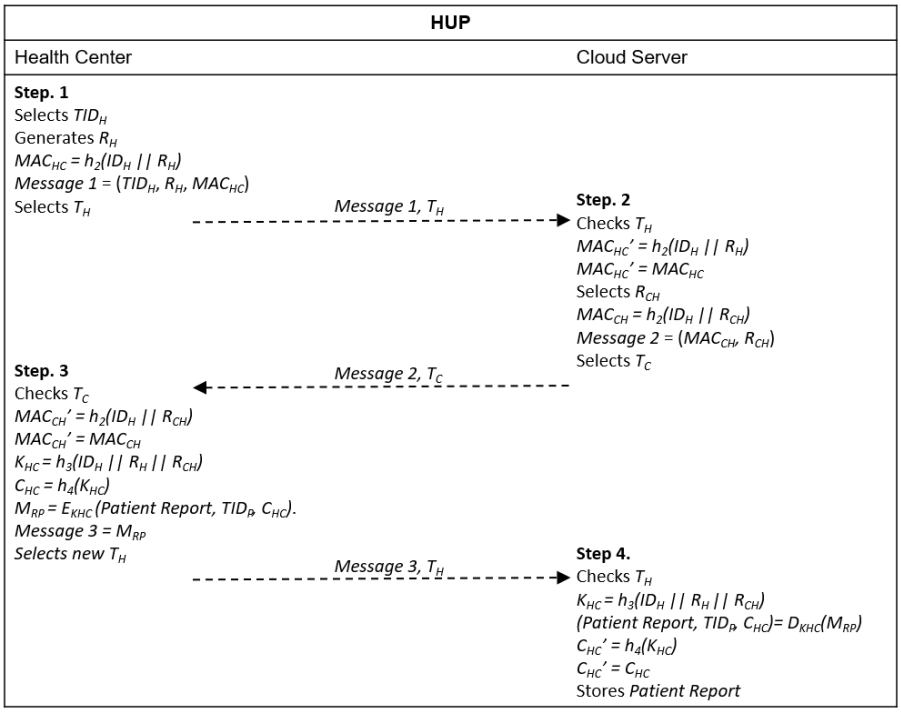

Figure 2. Message exchange in HUP.

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.

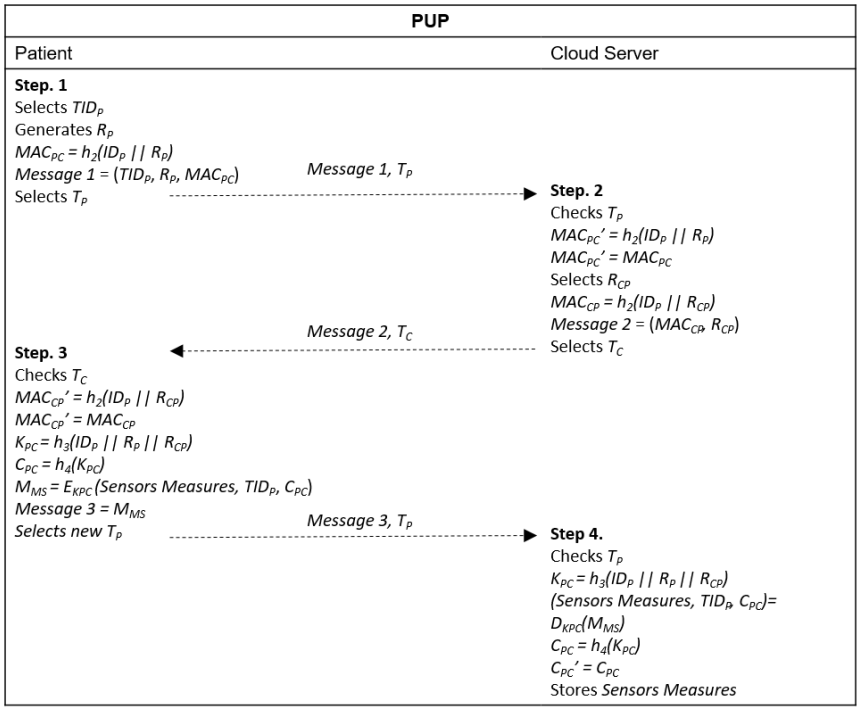

Figure 3. Message exchange in PUP.

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 $K_{PC} = h_3(ID_P \,||\, R_P \,||\, R_{CP})$, 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, 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 $MAC_{DC} = h_2(ID_D \,||\, R_D)$ and sends *Message 1* = (TID_D, R_D, MAC_{DC}) with a timestamp T_D to the cloud server.

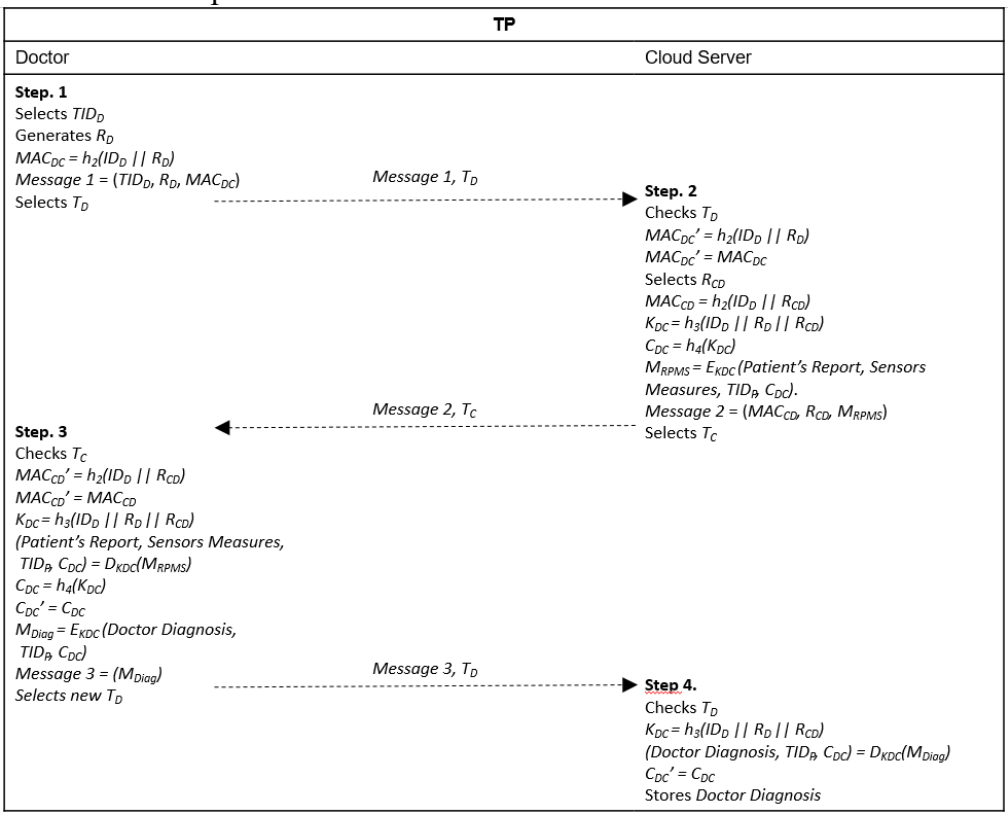

Figure 4. Message exchange in TP.

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.

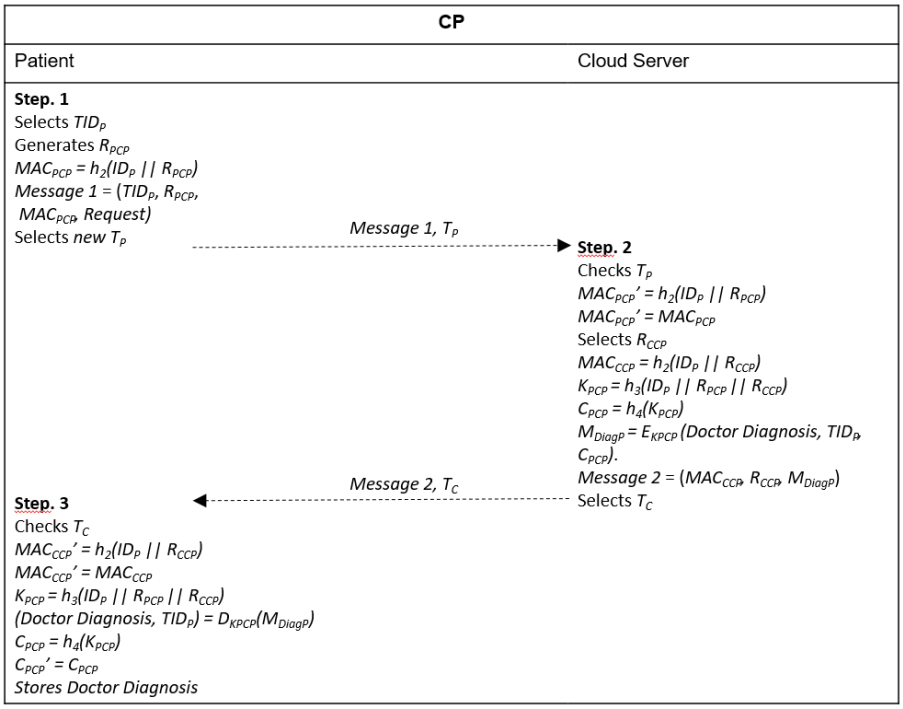

Figure 5. Message exchange in CP.

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 our 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.

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 our protocol 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 our protocol 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.	Our 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. (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. Our protocol 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 our protocol 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 our protocol 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	$5T_S + 11T_P + 8T_E + 32T_H = 2.43s$	$4T_S + 9T_E + 35T_H = 1.42s$	$12T_E + 32T_H = 1.2s$

Our protocol 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).

The message exchange over an insecure channel was analyzed in each of the four common phases performed by our protocol 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 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

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

Our protocol 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. Our protocol 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. Our protocol 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.