LACO: Lightweight Three-Factor Authentication, Access Control and Ownership transfer scheme for e-health systems in IoT

Seyed Farhad Aghili, Hamid Mala, Mohammad Shojafar,
Pedro Peris-Lopez

PII: S0167-739X(18)33129-7
DOI: https://doi.org/10.1016/j.future.2019.02.020
Reference: FUTURE 4773

To appear in: *Future Generation Computer Systems*

Received date : 15 December 2018
Revised date : 4 February 2019
Accepted date : 11 February 2019

Please cite this article as: S.F. Aghili, H. Mala, M. Shojafar et al., LACO: Lightweight Three-Factor Authentication, Access Control and Ownership transfer scheme for e-health systems in IoT, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.02.020

LACO: Lightweight Three-Factor Authentication, Access Control and Ownership Transfer Scheme for E-Health Systems in IoT

Seyed Farhad Aghili[a], Hamid Mala[a,*], Mohammad Shojafar[b], Pedro Peris-Lopez[c]

[a]*Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran*
[b]*Department of Computer Science, Ryerson University, Toronto, Canada*
[c]*Department of Computer Science, University Carlos III of Madrid, Avda. de la Universidad 30, 28911 Leganés*

Abstract

The use of the Internet of Things (IoT) in the electronic health (e-health) management systems brings with it many challenges, including secure communications through insecure radio channels, authentication and key agreement schemes between the entities involved, access control protocols and also schemes for transferring ownership of vital patient information. Besides, the resource-limited sensors in the IoT have real difficulties in achieving this goal. Motivated by these considerations, in this work we propose a new lightweight authentication and ownership transfer protocol for e-health systems in the context of IoT (LACO in short). The goal is to propose a secure and energy-efficient protocol that not only provides authentication and key agreement but also satisfies access control and preserves the privacy of doctors and patients. Moreover, this is the first time that the ownership transfer of users is considered. In the ownership transfer phase of the proposed scheme, the medical server can change the ownership of patient information. In addition, the LACO protocol overcomes the security flaws of recent authentication protocols that were proposed for e-health systems, but are unfortunately vulnerable to traceability, de-synchronization, denial of service (DoS), and insider attacks. To avoid past mistakes, we present formal (i.e., conducted on ProVerif language) and informal security analysis for the LACO protocol. All this ensures that our proposed scheme is secure against the most common attacks in IoT systems. Compared to the predecessor schemes, the LACO protocol is both more efficient and more secure to use in e-health systems.

Keywords: E-Health Systems, Internet of things (IoT), Cybersecurity, Personal data, Three-factor authentication protocol, Ownership transfer protocol.

1. Introduction

Health-care is an indispensable part of human life. In addition, in recent decades there has been an increase in life expectancy. Because of this, there has been an increase in the population over the age of 65 who regularly

*Corresponding author
Email addresses: sf.aghili@eng.ui.ac.ir (Seyed Farhad Aghili), h.mala@eng.ui.ac.ir (Hamid Mala), mohammad.shojafar@ryerson.ca (Mohammad Shojafar), pperis@inf.uc3m.es (Pedro Peris-Lopez)

demand medical services of some kind. Due to the large number of patients, the provision of high-quality care to at-risk patients may be interrupted or the quality of service may deteriorate. While technology cannot reduce the demand for health services, it can at least offer potential solutions by integrating traditional health-care systems with electronic devices [1]. Recent health-care systems, called e-health systems, are supported by electronic devices with wireless connectivity, which are currently communicated through a central device (gateway) which usually transmits the collected data to a cloud [2, 3] –in the future, the devices will be able to communicate directly with each other. The use of these systems provides virtual consultations to patients such that the vast majority of them can rest at home and be treated with telemedicine, which is provided by doctors and hospitals [4, 1]. With advances in the Internet of Things (IoT) systems, many medical and wearable devices, equipped with sensors and placed in or on the patient's body, can collect the vital real-time data and transmit it to a base station [5, 6]. This base station could be a kind of smartphone or tablet carried by the patient and would send the collected information to the hospital server [7, 8]. Finally, authorized users such as doctors and nurses can access these data to do or decide the best. As for the user's connection to the medical server, the user must be authenticated at an early stage, usually using a smart card [9]. Likewise, for some devices communication is bi-directional and authorized entities such as physicians can change the reprogramming of patient devices [10, 11].

Such a system, in which the patient is equipped with different sensors and a doctor can monitor her/him remotely and instantly and know her/his vital signs online, is called Internet of Medical Things (IoMT) [12, 13, 14]. In Figure 1 we can see the different environments and possible entities. Various classifications of the IoMT, its possible applications, and the associated security and privacy problems are presented in [15, 16]. In IoMT system, patient privacy is crucial and an unauthorized user should not be able to link any information to a particular patient [17]. In addition, each user can access the part of the data to which s/he has access. This access control mechanism is defined by the medical server and provided to the user by the policies stored on the smart-card. Additionally, the current owner of this privilege should be able to give up it to another user with the help of the medical server. To access the information, the legitimate user must be logged into the system and go through the authentication process. The user can then set a session key with the sensors (e.g., pacemaker or smart ECG T-shirt [18]) that collect patient information [19, 20]. The most relevant issue in this system is that the communication channels between the user, medical server and the patient are public channels that are insecure and the adversary can easily eavesdrop all the messages exchanged on these channels.

1.1. Scheme requirements

The proposed scheme for IoMT system should meet the following requirements, in which (F), (S) and (P) indicate the functional, security and privacy requirements respectively.

($F1$) *Access control*: Any legitimate user (doctor) can only access the part of the patient information allowed by the access control mechanisms defined by the medical server.

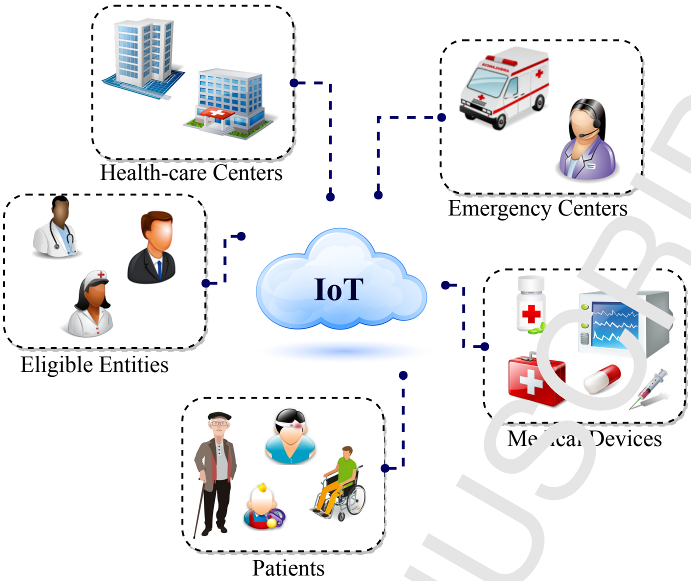

Figure 1: IoMT system main elements

(*F*2) *Energy consumption*: The scheme for IoT systems with resource-constrained sensors should be efficient in terms of computation and communication.

(*F*3) *Ownership transfer*: Accessibility to patient information can be revoked from one doctor and transferred to another.

(*S*1) *Mutual authentication*: The legitimacy of each entity must be validated before establishing the session key and transferring information.

(*S*2) *Confidentiality*: Only authorized users (doctors) should be able to access patient medical information.

(*S*3) *Integrity*: The freshness and integrity of all messages must be provided to ensure that the messages received have not been altered during transmission.

(*S*4) *Availability*: All users (doctors) must have easy access to the patient's medical data (collected by the user sensors).

(*P*1) *Entity privacy preserving*: An adversary should not be able to extract any information related to the doctor's identity. In addition, patient privacy must be preserved.

(*P*2) *Untraceability*: No attacker should be able to track the target user.

(*P*3) *Old owner privacy preserving*: When ownership of the patient's information is transferred to a new owner, the new owner should be unable to trace back any previous communication between the previous owner and the patient.

(*P*4) *New owner privacy preserving*: When the ownership of the old owner is revoked, the old owner should not be able to track any current communication between the new owner and the patient.

1.2. Threat Model

The assumed threat model for IoMT system mainly is based on the model proposed by the Dolev-Yao [21]. In this model, the adversary can intercept all the messages transferred in the protocol (passive adversary). S/he can also modify, delete and block messages that are transferred through the insecure channel (active adversary). We assume that the adversary can also execute a side channel attack and then can get the secrets stored on the smart card and the data stored on the medical server. In addition, the adversary can perform an insider attack to capture the private information stored in the server's database.

1.3. Motivation

Under the above system requirements and threat model, the proposal of a secure authentication protocol for IoMT systems is an important issue and raises a number of issues (i.e., security, privacy, access control, and ownership transfer). Because of these challenges, several authentication protocols have been recently proposed in literature [22, 23, 24], but most of them have security faults or are not compatible with all required features.

Furthermore, the sensors used in these systems have resource limitations, so the authentication protocol proposed for these systems must not only be secure but also sufficiently efficient. As a result, using lightweight cryptographic primitives can be a good solution to this problem.

1.4. Contribution

The contributions of this article are summarized below.

- We show how the Zhang et al. scheme (called ZZTL) [22] does not guarantee, contrary to what the authors claim, many of the security properties that are required of an authentication protocol in an IoMT system. In particular, we present several attacks against the ZZTL scheme including user traceability, desynchronization, DoS and insider attacks. To increase the level of security offered by the ZZTL protocol, we solve all the security problems found in this scheme.
- We propose a new architecture that is composed of three main entities: 1) user group (doctors, nurses and hospital managers); 2) medical server; and 3) patient group (see Section 3.3). The proposed protocol (called LACO) provides authentication and key agreement. Privacy and access control are also guaranteed. Therefore, only authorized entities can access sensitive patient information.
- We consider the situation where the patient's current doctor wants to transfer her or his privileges to a new doctor. To deal with this situation, we propose an ownership transfer phase in the LACO scheme.
- The security of the proposed scheme is examined from both a formal (ProVerif language [25]) and an informal point of view (see Section 7).
- The efficiency of our proposal, as shown in Section 8, is higher than that of the predecessor schemes. Therefore, our scheme can be used for resource-constrained sensors in IoMT systems.

1.5. Paper organization

The rest of the paper is organized as follows. The related work is presented in Section 2. Preliminaries and notations are explained in Section 3. The Section 4 provides a review of the ZZTL protocol and its drawbacks. In the Section 5, we present the security analysis of the ZZTL protocol. Our new scheme is proposed in Section 6. The security analysis and performance evaluation of the proposed scheme are discussed in Section 7 and Section 8, respectively. Finally, we draw some conclusions in Section 9.

2. Related work

In this section, we provide a holistic review of the literature that addresses security problems and solutions in the medical field. In particular, several e-health security schemes have been proposed in recent years (e.g., [26, 27, 28]) to solve the problem of pair-wise shared keys between various entities (i.e., patient, sensors, and server). In [29], the authors provide an in-depth review of authentication schemes based on Elliptic-curve cryptography (ECC) and show how most of the existing schemes are not suitable for IoTM systems due to their security vulnerabilities and/or the large number of resources they consume.

In [26] Le et al. present a mutual authentication protocol, which supports access control using Elliptic-curve cryptography. They indicate that the scheme consumes little energy and is secure against some common attacks such as DoS and reply attacks. However, the authors in [27] found some security vulnerabilities in [26]. To be precise, Kumar et al. in [27] present a two-factor authentication mechanism that provides mutual authentication and access control between the user and the medical sensor. Their system relies its security on asymmetric cryptography. Although the proposal is interesting, it lacks to consider the privacy and security of the ownership transfer problem. Subsequently, Chang et al. introduce a biometrics-based user authentication scheme that allows the legitimate user/patient to access the remote medical server using a collision-resistant one-way hash function [28]. This method prevents the modification of the transmitted data through by a malicious user, but according to [30] it fails to manage the data flows in the login, authentication and password exchange phases. In addition, it cannot protect the system against well-known attacks, such as an insider or man-in-the-middle attack. Indeed, Das and Goswami in [30] present an enhancement scheme and formally validate its security using AVISPA. Their authentication mechanism uses a symmetric secret session key between the user and the server to protect communications between both entities. Note that these last two mentioned protocols do not support the ownership transfer and three-factor authentication, nor the validation of privacy and security for the access control that is done in the LACO proposal.

In 2015, Amin et al. [31] found important security faults in [30]. These problems include user anonymity problem, off-line password guessing attack, smart card theft attack, user impersonation attack, server impersonation attack, and session key disclosure attack. To fix all this, they propose a robust remote user authentication scheme for e-health systems. For validation, they use the BAN logic to ensure the security of the mutual authentication and session key agreement schemes. After a thorough review of the paper, we realized that in [31] the patient can be tracked. Also, the scheme does not validate the password used for authentication and there is no mechanism to combat DoS attacks. Conversely, all these characteristics are covered in LACO proposal. Wang et al. [32] present an interesting review of two-factor authentication schemes. The authors point out how smart card breach attacks could compromise the entire system if the verification value is stored in the smart card. In addition, the attacker can easily guess the user password within polynomial time. In [24], the authors analyzed the security of several authentication schemes [23, 33, 27] and proposed a novel two-factor authentication scheme for health care systems. Unfortunately, their improved scheme remains vulnerable to off-line password guessing and de-synchronization attacks. Therefore, the two-factor model is not a secure model. Furthermore, these techniques cannot securely handle access control and ownership transfer, as is the case in the proposal presented in this article.

To solve the two-factor problem, researchers add biometric features to the two-factor model and present three-factor schemes. Several researchers have introduced three-factor authentication schemes for the medical context [34, 35, 36]. In [34] Farash present a user authentication and key agreement scheme that is robust, among others, against smart card attack, man-in-the-middle attack, untraceability and insider attack, being validated with BAN-logic and AVISPA tools. Nevertheless, as described in [35], the above scheme has some shortcomings. First, it is vulnerable to off-line password-guessing and user impersonation attacks. Secondly, it suffers from a lack of preservation of users' anonymity. Motivated by this, Amin et al. [35] design a secure three-factor user authentication protocol for the IoT system and present formal and informal validation against active and passive attacks. After that, Arasteh et al. [36] discover replay and DoS attacks against [35]. In addition, in [37] Jian et al. show several attacks against [35] including traceability and session key disclosure. They then propose a new scheme based on the Rabin's cryptosystem. Later, the same authors in [38] enhance the 3FA protocol of Lu et al. [39] to overcome its security pitfalls such as identity disclosure and user/server impersonation attacks. Although their proposal is novel and efficient, it lacks for management in the ownership transfer and data integrity.

In 2017, Liu and Chung [40] introduce a user authentication scheme using bilinear pairing and a trusted authority to authenticate the user. They also establish secure communication between a user and a sensor node. The scheme turned out not to be as secure as it was supposed to be [41]. For this reason, Challa et al. present a three-factor authentication and a key agreement scheme suitable for wireless health-care sensor networks, which is based on lightweight ECC [41]. Recently, in [22], Zhang et al. propose a three-factor authenticated key agreement scheme for e-health systems to protect user privacy through the use of a dynamic authentication mechanism. The authors state that their proposed scheme is proved to be semantic secure under the real-or-random model. Despite this, in Section 5 we show how the above protocol suffers from several attacks including de-synchronization, DoS, and insider attacks. LACO scheme aims to address the security weaknesses of all its predecessors and the details are found in the following sections.

3. Preliminaries and Notations

This is followed by a presentation of the Biohash function, the access control string and a description of the overall structure of the IoT system.

3.1. Biohash function

The biohash function converts the biometric template of the human fingerprints into a bits vector. This function [42, 43] has the following main properties:

- This function must have a low false rejection of the valid user.
- It should be computationally unfeasible for an adversary to revert the bits vector into its original feature vector.

3.2. Access control string

In our scheme, we suppose that the medical server provides a string called *HACO*, displayed in Fig. 2, for the user (U_i). This string has the following properties:

- It is the *output* of an irreversible hash function with a constant length of 160 bits like SHA-1. The use of a hash function guarantees the anonymity of the input string.
- As an *input* of the hash function, the medical server uses the user identity, dynamic attributes (e.g., location, time, noise), static attributes (e.g., the role of the user, hospital) and a user password. Fig. 2 presents an example of the input string.

This string is stored in the medical server and indicates that the owner has access to which sensors.

3.3. Proposed architecture

Our e-health system architecture is comprised of three main entities as shown in Fig. 3. To be precise, i) Medical server (S) that can collect information from patients using base stations (e.g. smart-phone or tablet) and provides the access control mechanisms for users to access vital patient data; ii) Group of users (U_i) that can be doctors, nurses and hospital managers. These entities must register on S using their smart-card. Through the use of this smart-card, the legitimate user can access to the part of the information sensed by the sensors for which the patient is authorized; and,

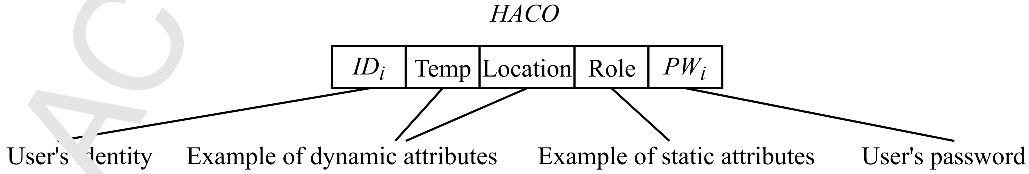

Figure 2: An example of user (doctor) access control string (*HACO*).

iii) Group of patients (P_j) that are equipped with wearable-medical-devices or implantable sensors. These sensors can collect the vital information related to the patient's body condition and then send these data to S with the help of the base stations.

3.4. Notation

The notation used in this paper is summarized in Table 1.

Table 1: Notation

Notation	Description
S	The medical server
U_i	The *i-th* user (doctor) of the e-health system
ID_i	The identity of the *i-th* user
PID_j	The identity of the *j-th* sensor
IDS_i	The identity of the smart-card given to the *i-th* user
PW_i	The password linked to the *i-th* user
B_i	The biometric traits belonging to the *i-th* user
r_x and K_x	The random numbers
T_x	The current time stamp
s	The master key of the medical server
SK_u, SK, SK	The session key calculated respectively by the doctor, the medical server and the sensor node of the patient
$HACS_j$	The hash of the access control string
$h(\cdot)$	A one-way hash function
$h_{bio}(\cdot)$	A secure biohash function
$\oplus$	XOR operation
$\|$	Concatenation operation

4. Review of ZZTL scheme

In this section, we briefly introduce the ZZTL authentication protocol [22], which consists of the user registration, login and authentication phases [22].

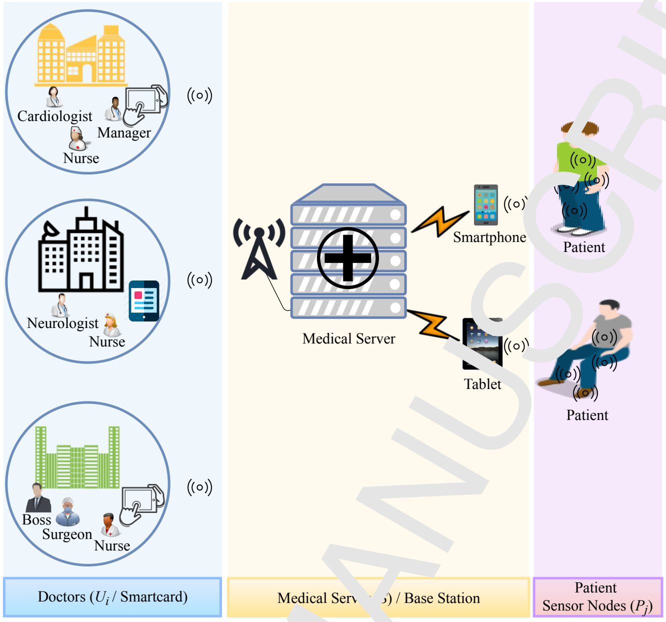

Figure 3: Our proposed architecture

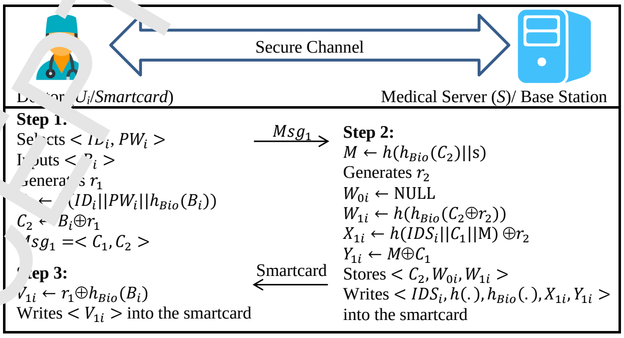

Figure 4: Registration phase of ZZTL scheme

4.1. Registration phase

In this phase of the protocol, the user U_i uses a secure channel to execute the following steps in conjunction with the medical server S.

Step 1. The user U_i chooses an identity ID_i and the password PW_i and then extracts her/his biometric data B_i and finally generates the random number r_1. Then, s/he computes $C_1 = h(ID_i \| PW_i \| h_{Bio}(B_i))$ and $C_2 = B_i \oplus r_1$ and sends the tuple $\langle C_1, C_2 \rangle$ to the S as shown by Msg_1 in Fig. 4.

Step 2. Upon receiving the registration request, the medical server S uses its master key s to compute $M = h(h_{Bio}(C_2) \| s)$. Next, S generates a random number r_2 and calculates $W_{1i} = h(h_{Bio}(C_2 \oplus r_2))$ and stores both value of C_2 and W_{1i} in its database along with W_{0i} that is NULL at first. Then, S computes $X_{1i} = h(IDS_i \| C_1 \| M) \oplus r_2$ and $Y_{1i} = M \oplus C_1$ and stores $\langle IDS_i, h(\cdot), h_{Bio}(\cdot), X_{1i}, Y_{1i} \rangle$ into the smart-card is given to the user U_i.

Step 3. Once the user receives the smart-card, s/he computes $V_{1i} = r_1 \oplus h_{Bio}(B_i)$ and writes it to the smart-card.

4.2. Login phase

When the user U_i wants to access the data stored on the medical server S, s/he inserts her/his smart-card into the terminal and performs the following steps to log into the system.

Step 1. U_i inserts her/his ID'_i and PW'_i and also allows the acquisition of her/his biometric information B_i using the terminal's sensor device.

Step 2. U_i generates a new random number r_3. Using the nformation stored on the smart-card, U_i calculates the messages $C'_1 = h(ID'_i \| PW'_i \| h_{Bio}(B'_i))$, $M' = Y_{ni} \oplus C'_1$, $r'_2 = X_{ni} \oplus h(IDS_i \| C'_1 \| M')$, $r'_1 = V_{ni} \oplus h_{Bio}(B'_i)$, $C_3 = h_{Bio}(B'_i \oplus r'_1 \oplus r'_2)$, $C_4 = B'_i \oplus r'_1 \oplus h(M' \| r_3)$ and $C_5 = r_3 \oplus h_{Bio}(B'_i \oplus r'_1)$ and sends the message Msg_2, which consists of tuple $\langle C_3, C_4, C_5 \rangle$, to the medical server S through an insecure channel.

4.3. Authentication and key agreement phase

In this phase, the user U_i executes five authentication steps to prove her/his legitimacy to S (see Fig. 5).

Step 1. After receiving the message Msg_2 from the login phase, S calculates $W'_{ni} = h(C_3)$ and then searches for the same value in its database. If it can find $W_{1i} = W'_{ni}$, it obtains the related C_2. If not, it does the searching again in the column W_{0i} to find if $W_{0i} = W'_{ni}$. Eventually, if a matching it is found, it extracts the related C_2. Otherwise, it finally aborts the connection –note that if $W_{0i} = W'_{ni}$, then S sets $W_{1i} = W_{0i}$.

Step 2. Then, S generates the new random number r_4 and computes $M^{\star} = h(h_{Bio}(C_2) \| s)$, $r'_3 = C_5 \oplus h_{Bio}(C_2)$ and $B_i \oplus r'_1 = C_4 \oplus h(M^{\star} \| r'_3)$. Next, it checks if $B_i \oplus r'_1$ and C_2 are within a defined threshold. If the threshold cannot satisfy the assumed value stored in the database, the session ends. Otherwise, S computes $C_6 = r_4 \oplus h(B_i \oplus r'_1)$ and $C_7 = h((B_i \oplus r'_1) \| r'_3 \| r_4)$ and then sends the Msg_3 (i.e., $\langle C_6, C_7 \rangle$) to U_i.

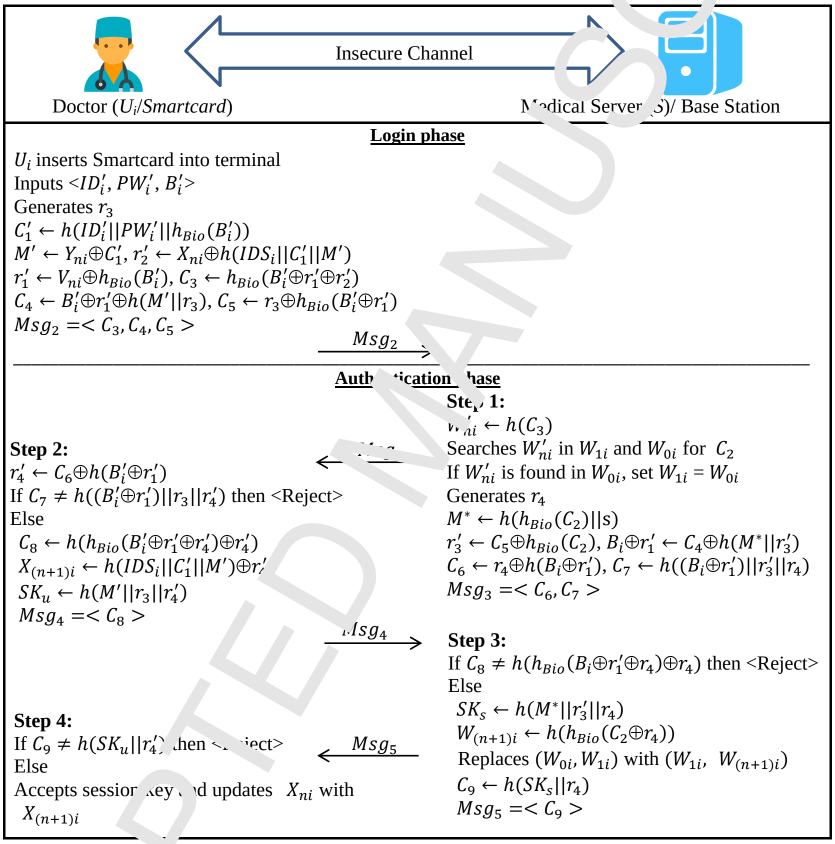

Figure 5: ZZTL login, authentication and key agreement phases

Step 3. Once U_i receives the Msg_3, s/he extracts $r'_4 = C_6 \oplus h(B'_i \oplus r'_1)$ and checks the correctness of C_7 received by comparing this value with the computed value of $h((B'_i \oplus r'_1)\|r_3\|r'_4)$. If the check fails, U_i terminates the connection. Otherwise, s/he computes $C_8 = h(h_{Bio}(B'_i \oplus r'_1 \oplus r'_4) \oplus r'_4)$ and $X_{(n+1)i} = h(IDS_i\|C'_1\|M') \oplus r'_4$. After this s/he calculates the session key $SK_u = h(M'\|r_3\|r'_4)$ and then sends $\langle C_8 \rangle$ to S as confirmation message Msg_4.

Step 4. After receiving the Msg_4, the S verifies the validity of C_8 by comparing this value with $h(h_{Bio}(B_i \oplus r'_1 \oplus r_4) \oplus r_4)$. If these two values are not equal, S aborts the connection. Otherwise, it computes the session key $SK_s = h(M^\star\|r'_3\|r_4)$ and also computes $W_{(n+1)i} = h(h_{Bio}(C_2 \oplus r_4))$. It then replaces $\langle W_{0i}, W_{1i} \rangle$ by $\langle W_{1i}, W_{(n+1)i} \rangle$. Finally S calculates $C_9 = h(SK_s\|r_4)$ and forwards the message $\langle C_9 \rangle$ to U_i as the message Msg_5.

Step 5. Once the message Msg_5 is received, U_i checks whether the equation $C_9 = h(SK_u\|r'_4)$ is satisfied. If not, it aborts the session. Otherwise, U_i accepts the session key SK_u and replaces X_{ni} by $X_{(n+1)i}$.

5. Security Analysis of the ZZTL Protocol

In ZZTL protocol [22], the authors stated that their scheme is not only secure against several attacks in IoT systems but also secure against insider attacks. In this scheme, the first protocol message sent in the login phase contains the constant value C_3 which is updated at the end of each protocol session. In this protocol, S stores the old dynamic string $W_{0i} = W_{ni}$ from the previous session and the new dynamic string $W_{1i} = W_{(n+1)i}$ from the current session to prevent de-synchronization attacks. S uses one of these values to verify the validity of the message C_3 sent by a valid user.

In this section, we show how an adversary can track a target U_i. We also present de-synchronization, DoS and insider attacks against ZZTL Protocol.

5.0.1. User traceability attack

In ZZTL protocol, the value of $C_3 = h_{Bio}(B_i \oplus r'_1 \oplus r'_2)$ is constant –note that the parameters B'_i and r'_1 are constant and the value of r'_2 is updated at the end of each protocol session. Therefore, if the adversary receives this message and blocks the server's response, s/he can track the *i-th* user in its next session. The success probability of this attack is 1.

5.0.2. De-synchronization attack

In our proposed de-synchronization attack the adversary follows the following steps.

- S/he eavesdrops $C_{3_{((n-1)\text{-}th\ session)}}$ of a successful session.
- In a new session, s/he replaces the current $C_{3_{((n)\text{-}th\ session)}}$ with the eavesdropped $C_{3_{((n-1)\text{-}th\ session)}}$, and sends message $Msg_2 = C_{3_{((n-1)\text{-}th\ session)}}, C_{4_{((n)\text{-}th\ session)}}, C_{5_{((n)\text{-}th\ session)}}$ to the server S;
- Upon receiving the message, S calculates $W'_{ni} = h(C_{3_{((n-1)\text{-}th\ session)}})$ and then searches its database for the same value. Consequently, it finds that $W_{0i} = W'_{ni}$, sets $W_{1i} = W_{0i}$, and extracts the related C_2. Then, it passes the

C_2 validity check. At this point, S computes a new random number $r_{4_{((n+1)\text{-}th\ session)}}$ and also calculates C_6 and C_7. Finally, S sends Msg_3 to U_i;

- After receiving the message Msg_3, U_i accepts the value of C_7 and sends the confirmation message C_8 to S;
- Now, S accepts the value of C_8 and calculates $W_{(n+1)i} = h(h_{Bio}(C_2 \oplus r_{4_{((n+1)\text{-}th\ session)}}))$. It then replaces $\langle W_{0i}, W_{1i}\rangle$ with $\langle W_{1i}, W_{(n+1)i}\rangle$, computes C_9, and sends the message C_9 to U_i;
- At this point, the adversary blocks the message C_9 and prevents U_i from accepting the updated value of $X_{(n+1)i}$.
- Therefore, U_i has $X_{(n)i} = h(IDS_i \| C_1 \| M) \oplus r_{4_{((n)\text{-}th\ session)}}$ and the server has $W_{0i} = h(h_{Bio}(B_i \oplus r_1 \oplus r_{4_{((n-1)\text{-}th\ session)}}))$ and $W_{1i} = h(h_{Bio}(B_i \oplus r_1 \oplus r_{4_{((n+1)\text{-}th\ session)}}))$ which are used to compute C_3,

  Since the value of C_3 computed by U_i can no longer satisfy the server-side checking process, the adversary leads the user in the de-synchronization state from this point on. The adversary success probability is maximum (i.e., p=1).

5.0.3. DoS attack

Since the server does not check the freshness of message Msg_2, and responds with Msg_3 through the calculated C_6 and C_7 values, the adversary can eavesdrop Msg_2 and resend this message a large number of times leaving the server out of service. This attack works until two successful sessions are established between the current user and the server.

5.0.4. Insider attack

By executing this attack, the adversary can obtain the information necessary to authenticate on the server without knowing the user's biometric template (user impersonation). The adversary does the following.

- S/he obtains $C_2 = B_i \oplus r_1$ from entire table stored in the server by executing an insider attack –note that the value of C_2 is constant.
- S/he obtains $r_{2_{(n\text{-}th\ session)}}$ from $C_{6_{((n-1)\text{-}th\ session)}}$ transmitted from the server to the user in the previous session (i.e., $(n-1)$-*th* session). Particularly, the equation $r_{2_{(n\text{-}th\ session)}} = r_{4_{((n-1)\text{-}th\ session)}} = C_{6_{((n-1)\text{-}th\ session)}} \oplus h(B_i \oplus r_1)$ is used.
- S/he employs C_2 and $r_{2_{(n\text{-}th\ session)}}$ to compute $C_{3_{(n\text{-}th\ session)}} = h_{Bio}(C_2 \oplus r_{2_{(n\text{-}th\ session)}})$.
- S/he generates a random number r_A and employs C_2 to compute $C_{4_{(n\text{-}th\ session)}} = C_2 \oplus h(M \| r_A)$ and $C_{5_{(n\text{-}th\ session)}} = r_A \oplus h_{Bio}(C_2)$.
- S/he uses the computed $C_{3_{(n\text{-}th\ session)}}$, $C_{4_{(n\text{-}th\ session)}}$, $C_{5_{(n\text{-}th\ session)}}$ as a message Msg_2 and sends it to the server S to establish a new session (i.e., (n)-*th* session).
- S responds to the user, who is actually the adversary, with the message $C_{6_{(n\text{-}th\ session)}}$.

- S/he obtains $r_{4_{(n\text{-}th\ session)}}$ from $C_{6_{(n\text{-}th\ session)}}$, by using the equation $r_{4_{(n\text{-}th\ session)}} = C_{6_{(n\text{-}th\ session)}} \oplus h(B_i \oplus r_1)$.
- S/he uses C_2 and $r_{4_{(n\text{-}th\ session)}}$ to compute $C_{8_{(n\text{-}th\ session)}} = h(h_{Bio}(C_2 \oplus r_{4_{(n\text{-}th\ session)}}) \oplus r_{4_{(n\text{-}th\ session)}})$;
- S/he uses the computed $C_{8_{(n\text{-}th\ session)}}$ as message Msg_4 and sends it to S.

Given that the message Msg_4 is valid for the medical server S, the adversary can establish a new successful session with S and impersonating a legitimated user. The adversary succeeds with a probability of 1.

6. Proposed LACO Protocol

To overcome the security pitfalls and flaws of previous authentication protocols such as the ZZTL [22] adopted for e-health systems, we propose a secure and energy-efficient protocol called LACO. The proposed scheme provides authentication and key agreement, in addition to satisfying access control and preserving privacy. Furthermore, LACO scheme considers the ownership transfer of the users.

Our proposed protocol consists of five important phases: (1) Setup phase; (2) Registration phase; (3) Login phase; (4) Authentication and key agreement phase; (5) Ownership transfer phase. The details are provided below.

6.1. Setup phase

In this phase of the scheme, the medical server calculates $M_j = h(PID_j||s)$ for the sensor *j-th* belonging to the system, where PID_j is the sensor's identity and s is the master key of S. Finally, the sensor stores M_j in its memory.

6.2. Registration phase

When executing this phase of the protocol, the user U_i contacts with the medical server S and requests the smart-card. This phase of the scheme is run as follows

Step 1. The user U_i chooses an identity ID_i and sends it to the S as shown in the message Msg_1 in Fig. 6.

Step 2. Upon receipt of the registration request, the medical server S checks if ID_i is in its database. If so, it requests another identity. If not, the medical server generates the random number r_s, uses its master key s and smart-card identity IDS_i to compute $X_{1i} = h(IDS_i||ID_i||r_s)$ and $Y_{1i} = h(X_{1i}||s)$. Next, S calculates a value $HACO_j$ compatible with the access polices, computes $Z_{1j} = h(X_{1i}||Y_{1i}) \oplus HACO_j$ and stores values of X_{1i} and Z_{1j} in its database along with X_{0i} and Z_{0j} which are $NULL$ at the beginning. Then, S saves $\langle X_{1i}, Y_{1i}, Z_{1j}, h_{Bio}(\cdot)\rangle$ on the smart-card and hands it to the user U_i.

Step 3. Once the user receives the smart-card, s/he inserts ID_i and the password PW_i and then extracts her/his biometric data B_i from the terminal device and calculates $A_{1i} = h_{Bio}(B_i) \oplus h(PW_i||ID_i)$ and $B_{1i} = Y_{1i} \oplus h(ID_i||PW_i||h_{Bio}(B_i))$. It then sets the $flag = 0$ and writes $\langle A_{1i}, B_{1i}, flag\rangle$ on the smart-card and also deletes Y_{1i}. Therefore, the smart-card has the following values associated with it:$\langle A_{1i}, B_{1i}, flag, X_{1i}, Y_{1i}, Z_{1j}, h_{Bio}(\cdot)\rangle$.

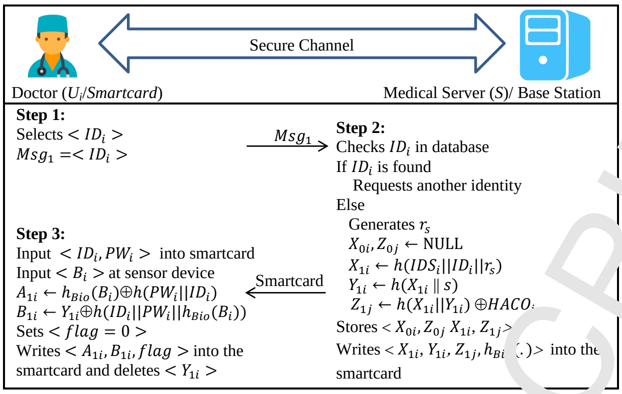

Figure 6: Registration phase of the proposed scheme

6.3. Login phase

When the user U_i decides to access the medical server's data, s/he inserts her/his smart-card into the terminal and does the login phase as the next step.

In detail, U_i inserts her/his ID'_i and PW'_i and also extracts her/his biometric information B'_i using the terminal's sensor device. Now, the smart-card computes $A'_{ni} = h_{Bio}(B'_i) \oplus h(PW'_i||ID'_i)$. If $A'_{ni} \neq A_{ni}$ the terminal rejects the smart-card. Otherwise it generates the new random numbers K_u and r_i, and a timestamp T_1. Using the information stored information on the smart-card, U_i calculates $Y'_{ni} = B'_{ni} \oplus h(ID'_i||PW'_i||h_{Bio}(B'_i))$ to compute messages $C_1 = K_u \oplus h(X_{ni}||Y'_{ni}||T_1)$, $C_2 = PID_j \oplus h(X_{ni}||Y'_{ni}||Z_{nj}||T_1)$, where PID_j is the identity of the sensor node to which the user wants to access to its data. Then, the smart-card checks the value of the $flag$. If it is equal to 0, it means that the previous session was successfully finished, therefore, it calcualtes $C_3 = X_{ni}||Z_{nj}$ and sets the $flag = 1$. Otherwise it means that the last session was not terminated and the smart-card did not do perform the update. Then, the smart-card computes $C_3 = h(r_j||X_{ni}||Y'_{ni})||h(r_i||Y'_{ni}||Z_{nj})$. Now, the smart-card calculates $C_4 = h(C_1||C_2||C_3||K_u||PID_j||T_1||r_i)$ and sends the message Msg_2, which includes the tuple $\langle C_1, C_2, C_3, C_4, r_1, T_1 \rangle$, to the medical server S through an insecure channel.

6.4. Authentication and key agreement phase

In this phase, the user U_i executes the following five authentication steps to prove her/his legitimacy to S (see Fig. 5). In addition, at the end of this session, U_i sets the session key with the other entities.

Step 1. When receiving the message Msg_2 transferred from login phase, S uses the current time T_2 and checks the timestamp condition. If $| T_1 - T_2 |$ is grater than ΔT, S aborts the connection. If not, for each tuple of $\langle X_{ni} = (X_{0i}, X_{1i}), Z_{nj} = (Z_{0j}, Z_{1j}) \rangle$ in its database it computes $Y'_{ni} = h(X_{ni}||s)$ and if $C_3 \neq h(r_j||X_{ni}||Y'_{ni})||h(r_j||Y'_{ni}||Z_{nj})$, and $C_3 \oplus (X_{ni}||Z_{nj}) \neq 0$, it rejects the connection. Otherwise, it concludes that X_{ni} and Z_{nj} are valid. Then, S calculates $K_u = C_1 \oplus h(X_{ni}||Y'_{ni}||T_1)$, $PID'_j = C_2 \oplus h(X_{ni}||Y'_{ni}||Z_{nj}||T_1)$ and $C'_4 = h(C_1||C_2||X_{ni}||Z_{nj}||K'_u||PID'_j||T_1||r_i)$. Eventually, S compares the value of C'_4 with the received C_4. If it is not equal, the connection ends. Otherwise,

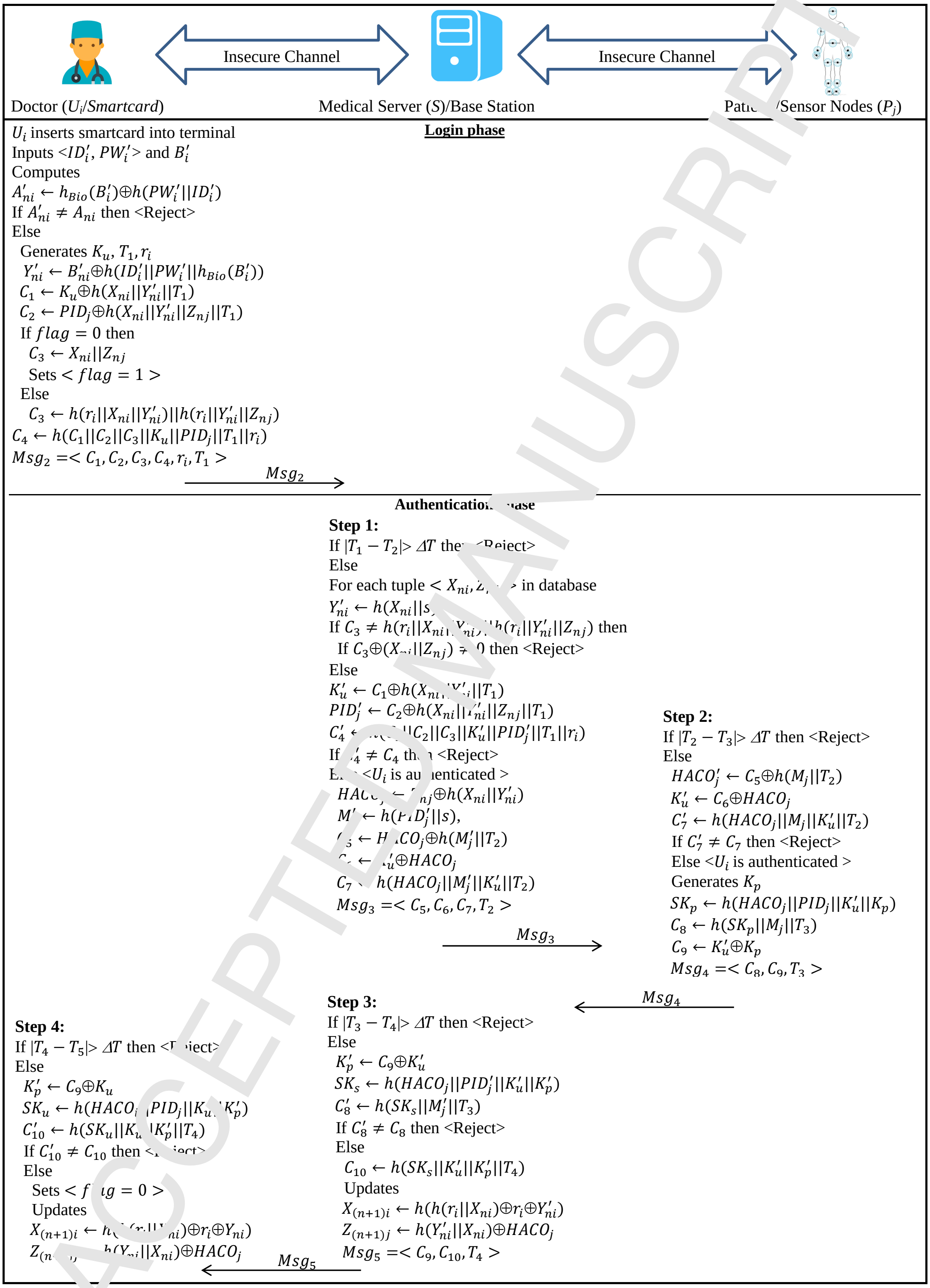

Figure 7: Login, authentication and key agreement phase of the proposed scheme

the user U_i is authenticated. After a successful authentication, S gets the access control string of the U_i as $HACO_j = Z_{nj} \oplus h(X_{ni}||Y_{ni})$. If this value is valid, it means that U_i can communicate with the sensor node with identity PID_j. Finally, S computes $M'_j = h(PID'_j||s)$, $C_5 = HACO_j \oplus h(M'_j||T_2)$, $C_6 = K'_u \oplus HACO_j$ and $C_7 = h(HACO_j||M'_j||K'_u||T_2)$ and then sends Msg_3, which consits of the tuple $\langle C_5, C_6, C_7, T_2\rangle$, to the sensor node P_j.

Step 2. Once P_j receives the Msg_3, it checks the validity of the timestamp T_2. If T_2 is not within the allowed margin, it aborts the connection. Otherwise, P_j uses its M_j value to obtain $HACO'_j = C_5 \oplus h(M_j||T_2)$. Then it extracts $K'_u = C_6 \oplus HACO_j$ and computes $C'_7 = h(HACO'_j||M_j||K'_u||T_2)$. If C'_7 is not equal to C_7, the session ends. If equal, P_j authenticates U_i, generates the random number K_p and calculates the session key $SK_p = h(HACO'_j||PID_j||K'_u||K_p)$. It also computes $C_8 = h(SK_p||M_j||T_3)$ and $C_9 = K'_u \oplus K_p$, where T_3 is the current timestamp of P_j. After that, P_j sends $\langle C_8, C_9, T_3\rangle$ to S as the response message Msg_4.

Step 3. After receiving Msg_4, S uses the current time T_4 and verifies the timestamp condition. If $| T_3 - T_4 | > \Delta T$, S terminates the connection. Otherwise, it extracts $K'_p = C_9 \oplus K'_u$ and the session key $SK_s = h(HACO_j||PID'_j||K'_u||K'_p)$. Then, it checks the validity of message recieved C_8 by comparing this value with $h(SK_s||M'_j||T_3)$. If these two values are not the same, S aborts the connection. Otherwise, it accepts the session key and also computes $C_{10} = h(SK_s||K'_u||K'_p||T_4)$ and updates $X_{(n+1)i} = h(h(r_i||X_{ni}) \oplus r_i \oplus Y'_{ni})$ and $Z_{(n+1)j} = h(Y'_{ni}||X_{ni}) \oplus HACO_j$. Finally it forwards the message $\langle C_9, C_{10}, T_4\rangle$ to U_i as message Msg_5.

Step 4. Once the message Msg_5 is received, U_i checks the validity of the T_4 timestamp. If the time T_4 is not within the threshold, it aborts the connection. Otherwise, it gets $K'_p = C_9 \oplus K_u$ and the session key $SK_u = h(HACO_j||PID_j||K_u||K'_p)$ and computes $C'_{10} = h(SK_u||K_u||K'_p||T_4)$. Then, it compares the value of the C_{10} received with C'_{10}. If it is not the same, it ends the connection. Otherwise it sets the $flag = 0$ and updates $X_{(n+1)i} = h(h(r_i||X_{ni}) \oplus r_i \oplus Y_{ni})$ and $Z_{(n+1)j} = h(Y_{ni}||X_{ni}) \oplus HACO_j$ and rewrites them into the memory of the smart-card.

At this point, the authentication phase is completed and the session key $SK_u = SK_s = SK_p$ is successfully established between the entities.

6.5. Ownership transfer phase

In this phase, the aim is to propose the mechanism that is in charge of lending the access permission to the data of the target sensor from one user to another. This phase is executed as follows. By executing these steps, the user's U_1 access permission is revoked and the permission is transferred to another user U_2.

1. A new user U_2 who wants to get the access permission, s/he inserts her/his smart-card into the terminal and enters ID'_2 and PW'_2. U_2 also extracts her/his biometric information B'_2 using the terminal's sensor device. Now, it calculates $A'_{n2} = h_{Bio}(B'_2) \oplus h(PW'_2||ID'_2)$ and checks whether $A'_{n2} = A_{n2}$. If not, the terminal rejects the smart-card. Otherwise, U_2, using the information stored on smart-card, computes $Y'_{n2} = B'_{n2} \oplus h(ID'_2||PW'_2||h_{Bio}(B'_2))$. It

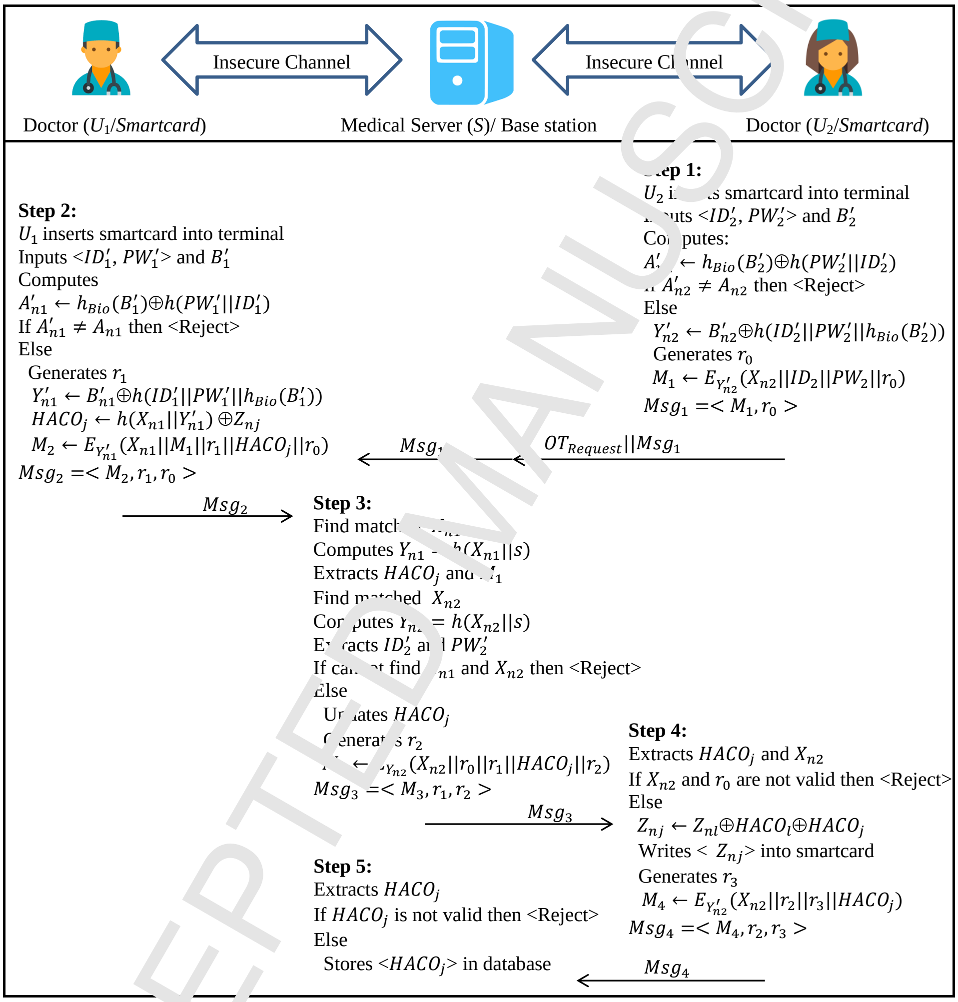

Figure 8: Ownership transfer phase of the proposed scheme

then generates the random number r_0 and calculates $M_1 = E_{Y'_{n2}}(X_{n2}\|ID_2\|PW_2\|r_0)$. Next U_2 sends the message $Msg_1 = M_1\|r_0$ along with the ownership transfer request to the current user U_1 who has the permission. This message is transferred through a medical server.

2. Once U_1 receives the message, s/he inputs her/his ID'_1 and PW'_1 and also retrieves her/his biometric information B'_1 using the terminal's sensor device. Now, it computes $A'_{n1} = h_{Bio}(B'_1) \oplus h(PW'_1\|ID'_1)$ and verifies whether $A'_{n1} = A_{n1}$. If not, the terminal rejects the smart-card. Otherwise, U_1 generates a random number r_1 and calculates $Y'_{n1} = B'_{n1} \oplus h(ID'_1\|PW'_1\|h_{Bio}(B'_1))$ using the information stored on the smart-card. Then it computes the access control string $HACO_j = h(X_{n1}\|Y'_{n1}) \oplus Z_{nj}$ and uses the encryption function $E_k(\cdot)$ to compute the message $M_2 = E_{Y'_{n1}}(X_{n1}\|M_1\|r_1\|HACO_j\|r_0)$. Finally, U_1 sends the message $Msg_2 = M_2\|r_1\|r_0$ to the medical server.
3. On receiving the message Msg_2 transferred from the current user U_1, the medical server finds the matched X_{n1} to calculate $Y_{n1} = h(X_{n1}\|s)$ for extracting $HACO_j$ and M_1 by decrypting the message M_2. Similarly it finds the matched X_{n2} to compute $Y_{n2} = h(X_{n2}\|s)$ for extracting ID_2 and PW_2 by decrypting the message M_1. If it cannot find X_{n1} and X_{n2} in its database and also cannot get r_0, it rejects the request. Otherwise it uses the new users U_2 identity ID_2 and password PW_2 to update $HACO_j$. In addition, it generates a random number r_2 and computes $M_3 = E_{Y_{n2}}(X_{n2}\|r_0\|r_1\|HACO_j\|r_2)$. Finally the medical server sends $Msg_3 = M_3\|r_1\|r_2$ to U_2.
4. Once U_2 received the message Msg_3 transferred from the medical server, s/he checks the validity of r_0 and X_{n2}. If these values are valid, s/he extracts $HACO_j$ by deciphering the message M_3 and uses Z_{nl} of the *l-th* sensor to compute $Z_{nj} = Z_{nl} \oplus HACO_l \oplus HACO_j = h(X_{n2}\|Y_{n2}) \oplus HACO_j \oplus HACO_l \oplus HACO_j = h(X_{n2}\|Y_{n2}) \oplus HACO_l$. Then s/he writes Z_{nj} on the smart-card. To inform the server that the ownership transfer was successfully, U_2 generates a random number r_3 and calculates $M_4 = E_{Y_{n2}}(X_{n2}\|r_2\|r_3\|HACO_j)$. Finally, s/he sends $Msg_4 = M_4\|r2\|r_3$ to the medical server.
5. When the message Msg_4 is received, the medical server extracts $HACO_j$ by decrypting the message M_4 and if it cannot find this string in its database, it cancels the request. Otherwise, it stores $HACO_j$ which is calculated for the access permission of the U_2 to *j-th* sensor.

7. Security Analysis of the Proposed scheme

In this section, we analyze our proposed scheme LACO informally and formally. The security threats are based on the Dolev-Yao model [21] and formal verification is done with the ProVerif language [44, 25].

7.1. Informal security analysis

In this section, we discuss the robustness of our proposed scheme against the most common attacks in IoMT systems.

7.1.1. Insider attack

Supposed a privileged insider entity attempts to obtain user-related information from the entire table stored on the server. S/he can get $X_{ni} = h(IDS_i||ID_i||r_s)$, $Z_{nj} = h(X_{ni}||Y_{ni}) \oplus HACO_j$, and $HACO_j$ values and also eavesdrop messages from a full session. Nevertheless, s/he cannot disclose any vital information related to the user (e.g., ID_i, PW_i and B_i) by employing these three parameters, nor can calculate Msg_2 without knowing $Y_{ni} = h(X_{ni||s})$ to impersonate the user and establish a new session with the medical server. Therefore, the proposal is resistant to insider attacks.

7.1.2. Stolen smart-card attack

In this attack, the adversary needs to obtain important parameters using information stored in a non-tamper-resistant smart-card. In the LACO authentication protocol, the adversary can only obtain the information $\langle A_{ni}, B_{ni}, flag, X_{ni}, Z_{nj} \rangle$ stored in the smart-card. Due to the absence of some necessary values (ID_i, PW_i, B_i and PID_j), the adversary cannot calculate Msg_2 to establish a new session. Furthermore, the collision-resistance property of the one-way hash function provides additional robustness as an attacker cannot reveal the ID_i, PW_i and B_i associated with the user U_i. Thus, security against the stolen smart-card attack is provided successfully.

7.1.3. Off-line password guessing attack

If an adversary finds a message (e.g., transferred in the protocol flow or stored in the smart-card) in which all parameters are known except the password PW_i, s/he can perform a dictionary attack and guess the password. In our proposed scheme, all the messages involving PW_i are computed by using B_i and ID_i, so the adversary cannot find a message whose only unknown parameter in it is PW_i. Therefore, our proposed scheme is robust against this attack.

7.1.4. User impersonation attack

In this attack, the adversary attempts to provide the login messages either by eavesdropping or by computing these messages to deceive the server as a legitimate user. In LACO if the adversary replays the login message $Msg_2 = \langle C_1, C_2, C_3, C_4, r_i, T_1 \rangle$ of the previous sessions to the server, the server checks the validity of Msg_2 by verifying C_4. The adversary should forge C_4 by employing Y_{ni} and PID_j. Due to lack of any knowledge about the user's identity ID_i, the password PW_i and the biometric template B_i, the adversary cannot compute a valid C_4. Therefore, in LACO scheme user impersonation attacks are unsuccessful.

7.1.5. Medical server impersonation attack

To impersonate the medical server S, the adversary A has to send a valid message $Msg_3 = \langle C_5, C_6, C_7, T_2 \rangle$ to the patient (sensor node). The challenge for A is to calculate $C_7 = h(HACO_j||M_j||K_u||T_2)$ s/he needs to know M_j, K_u and $HACO_j$ which is impossible. Thus, providing or falsifying the message as mentioned above is impossible for A. On the other side, A cannot compute message $Msg_5 = \langle C_9, C_{10}, T_4 \rangle$ because s/he has no knowledge of K_u, K_p, and SK_s. So, A cannot fool the user either. Therefore, LACO scheme can resist the attack of medical server impersonation.

7.1.6. Sensor node impersonation attack

In LACO scheme, when the sensor node P_j authenticates a medical server S, as an acknowledgement, it computes $C_8 = h(SK_p||M_j||T_3)$ and $C_9 = K_u \oplus K_p$ and responds to S. To forge these two messages, the adversary A needs to know K_u and K_p. Moreover, due to lack of knowledge about $HACO_j$ and PID_j s/he cannot calculate $SK_p = h(HACO_j||PID_j||K_u||K_p)$. Therefore, A cannot falsify the messages of the sensor node to execute this attack.

7.1.7. Session key security

If the attacker tries to obtain a session key, s/he can do so either by eavesdropping the messages of the protocol or by computing it with the help of parameters extracted from smart-card memory. In LACO, the messages $C_8 = h(SK_p||M_j||T_3)$ and $C_{10} = h(SK_s||K_u||K_p||T_4)$ contain the session key (SK_p and SK_s). Nevertheless, in these two messages, the session key is protected by the one-way hash function $h(\cdot)$. In addition, the parameters the adversary gets from smart-card memory are M_j and PID_j which are not enough to compute the session key $SK_p = h(HACO_j||PID_j||K_u||K_p)$. For all this, our proposed scheme satisfies the session key security.

7.1.8. Entity privacy

In this attack, an adversary A tries to find any information related to a certain user U_i (e.g., user's identity ID_i, password PW_i and biometric template B_i) or related to a sensor nod P_j (e.g., sensor node's identity PID_i). As in LACO these parameters are never transferred in plain-text, and due to the collision-resistant property of the one-way hash function $h(\cdot)$, it is computationally impossible for A to derive these parameters. Therefore, LACO preserves the privacy of the user.

7.1.9. New user privacy

In the ownership transfer phase of LACO, the medical server S uses the identity ID_2 and password PW_2 of the new user U_2 and updates the string $HACO_j$ and then encrypts it with U_2's key Y_{n2} along with X_{n2}, r_1, and r_2 as the message M_3. Finally S sends this ciphertext to U_2, so the old user U_1 cannot decrypt M_3 without knowing the value of Y_{n2} and cannot get the updated $HACO_j$. Therefore, the old user can never again access to the patient information sensed by sensor nod P_j.

7.1.10. Old user privacy

In the LACO scheme, in both authentication and ownership transfer phases, the value of the $HACO_j$ is not transferred in plaintext but is transferred using a one-way hash function. So after transferring the patient ownership to the new user, the current user cannot get the value of previous $HACO_j$. Therefore, the new user will not be able to track past interactions between the patient and her/his previous user.

7.1.11. Windowing problem

In this attack, the adversary should not be able to find the any time interval in which the new user U_2 and the old user U_1 can access the current patient information. In the LACO scheme, the medical server sends $HACO_j$ to the new user, then the new owner uses it to computes Z_{nj} and stores it on the smart-card. Therefore so we cannot find a time period in which both the new user (U_2) and the old user (U_1) can access the patient information. In short, the windowing problem does not exist in LACO.

7.2. Formal security analysis

This section presents the formal security verification of the LACO authentication protocol. Various methods are used for formal verification of security protocols in the literature (e.g., the BAN-logic [45], AVISPA [46], ProVerif [25]). The well-known ProVerif language is used in this work. The ProVerif uses the Dolev-Yao cryptography model [21] to evaluate the security level of the protocol. ProVerif supports cryptographic operations such as symmetric encryption/decryption and hash functions. Some basic terms and process grammars of the ProVerif language are presented in Table 3. The premises, which are our assumptions for the scheme channels, session keys, secret keys, constants, functions, equations, queries and events in the analysis, are defined in Fig. 9. The processes linked to the user U_i, the medical server S, and the sensor node P_j are illustrated in Fig. 10. In the box on the left, we first encoded the user registration phase and the rest corresponds to the encoding of the login, the authentication and key agreement phases on the user side. In the same way, in the central box, we encoded the setup and registration phases as well as the authentication and key agreement phases on the medical server side. Finally, in the box of the right, we encoded the setup phase and the authentication and key agreement phases on the patient/sensor side. Eventually, the results of the ProVerif verification are shown in Fig. 11. The results show that all the events result in "true" and also demonstrate that LACO is secure.

In Table 2, we compare the security and functionality features of our LACO authentication protocol with other schemes presented in the literature for IoMT systems. As for the table notation, Y and N indicate to "provide" and "not to provide" the property of security and functionality, respectively.

8. Performance comparison

In this section, we evaluate the computation cost and communication cost of the LACO authentication and key agreement protocol. We remind that LACO scheme has two main phases: 1) authentication and key agreement phase; and 2) ownership transfer phase. The ownership transfer phase is executed when it is necessary to change the proprietorship of the user/doctor. To the best of our knowledge, we are the first work to address the above task. Therefore, in this section we only evaluate the authentication and key agreement phase.

8.1. Computation cost evaluation

To evaluate efficiency of LACO and compare it with previous work, we use the most common cryptographic techniques for secure communications, such as AES cipher and SHA-1 hash algorithm. In [47] and [48], the execution time and the length required for AES, SHA-1 and biohash are T_s = 0.1303 ms, T_h = 0.0004 ms, and T_{bh} = 0.01 ms, respectively. Therefore, the estimated computation cost for the proposed LACO scheme is 0.0212 ms, while for ZZTL [22],

Table 2: Security/functionality features comparison

Attributes	ZZTL [22]	[23]	[30]	[24]	[27]	LACO
User untraceability preservation	*N*	*Y*	*Y*	*Y*	*Y*	*Y*
Security against replay attack	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*
Security against user impersonation attack	*Y*	*N*	*N*	*Y*	*Y*	*Y*
Security against server impersonation attack	*Y*	*N*	*N*	[illegible]	*Y*	*Y*
Security against sensor node impersonation attack	*Y*	*N*	*Y*	*Y*	*Y*	*Y*
Security against de-synchronization attack	*N*	*Y*	*Y*	*N*	*Y*	*Y*
Security against DoS attack	*N*	*Y*	*Y*	*Y*	*Y*	*Y*
Immunity against insider attack	*N*	*Y*	*Y*	[illegible]	*N*	*Y*
Immunity against stolen smart-card attack	*Y*	*Y*	*N*	*Y*	*Y*	*Y*
Immunity against session key disclosure attack	*Y*	*Y*	*N*	*Y*	*Y*	*Y*
Immunity against off-line password guessing attack	*Y*	*N*	*N*	*N*	*N*	*Y*
Anonymity of the user	*Y*	*Y*	*N*	*Y*	*N*	*Y*
Support of three-factor security	*Y*	[illegible]	*Y*	*N*	*N*	*Y*
Support of access control	*N*	*N*	*N*	*N*	*N*	*Y*
Support of ownership transfer	*N*	*N*	*N*	*N*	*N*	*Y*

Table 3: Notations of the ProVerif language

Notation	Description
free x : channel	x is a public channel
free x : channel [private]	x is a private channel
free y : bitstring [private]	y is a global bit-string that is not known by the attacker
free y : bitstring	y is a global bit-string that is known by the attacker
const y : bitstring	y is a constant bit-string
new y : bitstring	y is created as a fresh bit-string
table T(bitstring, bitstring, bitstring)	T is the table which takes three records of bit-strings
insert T(a, b, c)	Inserting the records a, b and c into the table
get T(= a, b, c)	retrieving a record in accordance with parameters a, b and c
in(x, y)	y is the input message received through channel x
out(x, y)	y is the output message sent through channel x
fun	defining the function
let y = a in	Evaluating a y by a value a
if M then N else P	If condition M is satisfied then do N else do P
query attacker(y)	Evaluating the secrecy of the term y against the simulated threat model
event e(y)	Event e can occur if an evaluation of y is succesfull
inj-event(d(y)) ==> inj-event(e(z))	For each occurrence of the event $d(y)$, at least there is an earlier occurrence of the event $e(z)$.

```
(*-LACO channels-*)
free c: channel.
free sc0: channel [private].
free sc1: channel [private].
(*-LACO session keys-*)
free SKu: bitstring [private].
free SKp: bitstring [private].
free SKs: bitstring [private].
(*-Server's secret key-*)
free s:bitstring [private].
(*-LACO constants-*)
free IDi: bitstring [private].
free PWi: bitstring [private].
free Bi: bitstring [private].
const IDSi: bitstring.
const SIDi: bitstring.
const PIDj: bitstring.
const HACOj: bitstring.
const f0: bitstring.
const f1: bitstring.
table T(bitstring,bitstring,bitstring).

(*-LACO functions-*)
fun h(bitstring):bitstring.
fun hBio(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.
(* Scheme equations *)
equation forall x:bitstring,y:bitstring;
xor(xor(x,y),y)=x.

(*-LACO queries-*)
query attacker(SKu).
query attacker(SKp).
query attacker(SKs).
query id:bitstring; inj-event(UserAuth(id))
==> inj-event(UserLogin(id)).

(*—LACO events-*)
event UserLogin(bitstring).
event UserAuth(bitstring).
```

Figure 9: Premises of the code for LACO

```
let User=
out(sc0,IDi);
in(sc0,(X:bitstring,Y:bitstring,Z:bitstring));
let A=xor(hBio(Bi),h(con(PWi,IDi))) in
let B=xor(Y,h(con(IDi,con(PWi,hBio(Bi)))))
in
let F=f0 in

!
(
event UserLogin(IDi);
new uku:bitstring;
new uri:bitstring;
new uT1:bitstring;
if A = xor(hBio(Bi),h(con(PWi,IDi))) then
let uY =
xor(B,h(con(IDi,con(PWi,hBio(Bi))))) in
let uC1 = xor(uku,h(con(X,con(uY,uT1)))) in
let uC2 =
xor(PIDj,h(con(X,con(uY,con(Z,uT1))))) in
if F = f0 then let uC3 = con(X,Z) else let uC3
=
con(h(con(uri,con(X,uY))),h(con(uri,con(uY,Z
)))) in
let F=f1 in
let uC4 =
h(con(con(con(con(con(con(uC1,uC2),uC3),uk
u),PIDj),uT1),uri))in
let Msg2 = (uC1,uC2,uC3,uC4,uri,uT1) in
out(c,Msg2);
in(c,(uC9:bitstring,uC10:bitstring,uT4:bitstring
));
let ukp = xor(uC9,uku) in
let SKu =
h(con(con(con(HACOj,PIDj),uku),ukp)) in
if uC10 = h(con(con(con(SKu,uku),ukp),uT4))
then
let F = f0 in
let Xnew = h(xor(xor(h(con(uri,X)),uri),uY))
in
let Znew = xor(h(con(uY,X)),HACOj) in
let X = Xnew in
let Z = Znew in
0
).
```

```
let SRegU =
in(sc0,SIDi:bitstring);
new Srs:bitstring;
let SX = h(con(con(IDSi,SIDi),Srs)) in
let SY = h(con(SX,s)) in
let SZ = xor(h(con(SX,SY)),HACOj) in
insert T(SIDi,SX,SZ);
out (sc0,(SX,SY,SZ)).

let SRegP =
let SMj = h(con(PIDj,s)) in
out(sc1,SMj).

let SAuth =
in(c,(SC1:bitstring,SC2:bitstring,SC3:bitstring,
SC4:bitstring,Sri:bitstring,ST1:bitstring));
new ST2: bitstring;
get T(=SIDi,SX,SZ) in
let SY = h(con(SX,s)) in
if SC3 = con(SX,SZ) || SC3 =
con(h(con(Sri,con(SX,SY))),h(con(Sri,con(SY
,SZ)))) then
let Sku = xor(SC1,h(con(con(SX,SY),ST1)))
in
let SPIDj =
xor(SC2,h(con(con(con(SX,SY),SZ),ST1))) in
if SC4 =
h(con(con(con(con(con(con(SC1,SC2),SC3),S
ku),SPIDj),ST1),Sri)) then
event UserAuth(SIDi);
let SHACOj = xor(SZ,h(con(SX,SY))) in
let SMj = h(con(PIDj,s)) in
let C5 = xor(SHACOj,h(con(SMj,ST2))) in
let C6 = xor(Sku,SHACOj) in
let C7 =
h(con(con(con(SHACOj,SMj),Sku),ST2)) in
let Msg3 = (C5,C6,C7,ST2) in
out(c,Msg3);
in
in(c,(SC8:bitstring,SC9:bitstring,ST3:bitstring));
new ST4:bitstring;
let Skp = xor(SC9,Sku) in
let SKs =
h(con(con(con(SHACOj,SPIDj),Sku),Skp)) in
if SC8 = h(con(con(SKs,SMj),ST3)) then
let C10 = h(con(con(con(SKs,Sku),Skp),ST4))
in
let SXnew =
h(xor(xor(h(con(Sri,SX)),Sri),SY)) in
let SZnew = xor(h(con(SY,SX)),SHACOj) in
let SX = SXnew in
let SZ = SZnew in
let Msg5 = (SC9,C10,ST4) in
out(c,Msg5).

let S = SRegU | SRegP | SAuth.
process !User |!S |!Patient
```

```
let Patient =
in(sc1,pMj:bitstring);
!
(
in(c,(pC5:bitstring,pC6:bitstring,pC7:bitstring,
pT2:bitstring));
new pkp:bitstring;
new pT3:bitstring;
let pHACOj = xor(pC5,h(con(pMj,pT2))) in
let pku = xor(pC6,pHACOj) in
if pC7 =
h(con(con(con(pHACOj,pMj),pku),pT2)) then
let SKp =
h(con(con(con(pHACOj,PIDj),pku),pkp)) in
let C8 = h(con(con(SKp,pMj),pT3)) in
let C9 = xor(pku,pkp) in
let Msg4 = (C8,C9,pT3) in
out(c,Msg4);
0
).
```

Figure 10: ProVerif scripts of LACO

```
Query not attacker(SKu[])
RESULT not attacker(SKu[]) is true.
Query not attacker(SKp[])
RESULT not attacker(SKp[]) is true.
Query not attacker(SKs[])
RESULT not attacker(SKs[]) is true.
Query inj-event(UserAuth(id)) ==> inj-event(UserLogin(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserLogin(id)) is true.
```

Figure 11: ProVerif results of LACO

He *et al.*'s protocol [23], Das *et al.*'s scheme [30], Amin *et al.*' protocol [24] and Kumar *et al.*'s scheme [27] is 0.0476 ms, 1.1755 ms, 0.0072 ms, 0.0148 ms, and 0.9141 ms, respectively. It is clear from Table 4 that the computation cost for the proposed scheme is lower than that of all other existing schemes, with the exception of the protocols [30] and [24]. In terms of communication cost, LACO transmits a slightly lower number of bits than [24] and double than [30]. Although [30] in numbers is more efficient than LACO, note as you can see in the Table 2 that this solution is much more insecure, which makes the LACO schema a more appropriate solution from the point of view of security and sensor resources.

As for the sensor point of view, the cost on this side is shown in Table 5. From these results, it is clear that the LACO scheme is more efficient than the other schemes for this perspective. Note that because the authors did not consider the sensor node in the ZZTL scheme, no value could be provided for this protocol in the Table mentioned above.

From the foregoing We conclude that the proposed scheme offers additional functionality features (like access control, and three-factor security) and provides better security than the predecessor schemes (see Table 2). At the same time, it is very efficient in terms of resource consumption which allows it to be implemented in sensors with constrained resources.

Table 4: Overall computational and communication cost of the IoMT authentication schemes

Scheme	Total computation cost	Communication cost (bits)	Estimated time (ms)
ZZTL [22]	$19T_h + 4T_{bh}$	1120	0.0476
He *et al.* [23]	$7T_h + 9T_s$	1216	1.1715
Das *et al.* [30]	$18T_h$	1280	0.0072
Amin *et al.* [24]	$37T_h$	2720	0.0148
Kumar *et al.* [27]	$5T_h + 7T_s$	2592	0.9141
LACO	$\mathbf{28T_h + 1T_{bh}}$	**2208**	**0.0212**

Table 5: Sensor node computational cost of the IoMT authentication schemes

Scheme	Computation cost	Estimated time (ms)
ZZTL [22]	–	–
He *et al.* [23]	$1T_h + 2T_s$	0.261
Das *et al.* [30]	$8T_h$	0.0032
Amin *et al.* [24]	$6T_h$	0.0024
Kumar *et al.* [27]	$1T_h + 2T_s$	0.261
LACO	$\mathbf{4T_h}$	**0.0016**

8.2. Communication cost evaluation

In Table 4, we also provide a communication comparison between our proposed LACO protocol and the predecessors presented for IoMT systems. In our experiments, the timestamp is 32 bits, the output of the hash function is 160 bits, the random numbers length is 160 bits, and AES cipher outputs 256 bits. Although the communication cost of ZZTL, [23] and [30] is less than LACO, our scheme offers additional functionality features (like access control, and three-factor security) and provides a security level higher than ZZTL, [23] and [30] (see Table 2).

9. Conclusion and Future works

The e-health management systems integrated by IoT faced several challenges, such as secure communications and authentication and key agreement protocols. The most important limitation in these systems is the limited resources of IoT sensors, which makes it difficult to provide an adequate security level for the system. In this work, we present a new authentication and key agreement protocol that preserves anonymity and provides an access control mechanism for the user. Our proposed protocol, called LACO, can also cover the transfer of user/doctor ownership. In the LACO scheme, when it is necessary to change the proprietorship of the user/doctor, the ownership transfer phase is executed with the help of the medical server. To the best of our knowledge, LACO is the first contribution that addresses the ownership transfer of the user/doctor in IoMT systems. We evaluated both the security and efficiency of LACO and demonstrated that our proposed scheme is secure and practical for being employed in IoMT systems. As future work, we would like to implement LACO on a low-cost hardware platform and demonstrate that it can be used in the real world. In addition, a key aspect to study also on the proposed solution is its impact on the quality of service offered to patients, which could be studied with a pilot project in the hospital with a small group of patients. Note that in healthcare there is always a balance between the patient safety and the security of the scheme supported on-board by the medical device. Finally, the integration of the proposed scheme with existing standards and regulations in the medical field is very relevant and should be studied in the future as well.

10. References

[1] H. Hamidi, An approach to develop the smart health using internet of things and authentication based on biometric technology, Future Generation Computer Systems 91 (2019) 434 – 449.

[2] M. Jayaratne, D. Nallaperuma, D. D. Silva, D. Alahakoon, B. Devitt, K. E. Webster, N. Chilamkurti, A data integration platform for patient-centered e-healthcare and clinical decision support, Future Generation Computer Systems 92 (2019) 996 – 1008. doi:https://doi.org/10.1016/j.future.2018.07.061.
URL http://www.sciencedirect.com/science/article/pii/S0167739X17308142

[3] A. A. Mutlag, M. K. A. Ghani, N. Arunkumar, M. A. Mohammed, O. Mohd, Enabling technologies for fog computing in healthcare iot systems, Future Generation Computer Systems 90 (2019) 62 – 78. doi:https://doi.org/10.1016/j.future.2018.07.049.
URL http://www.sciencedirect.com/science/article/pii/S0167739X18314006

[4] K. E. Deligiannidis, Primary care issues in rural populations, Physician Assistant Clinics 4 (1) (2019) 11 – 19, primary Care of the Medically Underserved. doi:https://doi.org/10.1016/j.cpha.2018.08.001.
URL http://www.sciencedirect.com/science/article/pii/S2405799118300793

[5] B. Afzal, M. Umair, G. A. Shah, E. Ahmed, Enabling iot platforms for social iot applications: Vision, feature mapping, and challenges, Future Generation Computer Systems 92 (2019) 718 – 731. doi:https://doi.org/10.1016/j.future.2017.12.002.
URL http://www.sciencedirect.com/science/article/pii/S0167739X17312724

[6] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach, Future Generation Computer Systems 78 (2018) 641 – 658. doi:https://doi.org/10.1016/j.future.2017.02.014.
URL http://www.sciencedirect.com/science/article/pii/S0167739X17302121

[7] B. Alexander, S. Haseeb, A. Baranchuk, Are implanted electronic devices hackable?, Trends in Cardiovascular Medicine-doi:https://doi.org/10.1016/j.tcm.2018.11.011.

[8] A. Baranchuk, M. M. Refaat, K. K. Patton, M. K. Chung, K. Krishnan, V. Kutyifa, G. Upadhyay, J. D. Fisher, D. R. Lakkireddy, A. C. of Cardiology, et al., Cybersecurity for cardiac implantable electronic devices: what should you know?, Journal of the American College of Cardiology.

[9] J. Kim, Energy-efficient dynamic packet downloading for medical iot platforms, IEEE Transactions on Industrial Informatics 11 (6) (2015) 1653–1659.

[10] C. Camara, P. Peris-Lopez, J. E. Tapiador, Security and privacy issues in implantable medical devices: A comprehensive survey, Journal of Biomedical Informatics 55 (2015) 272 – 289. doi:https://doi.org/10.1016/j.jbi.2015.04.007.
URL http://www.sciencedirect.com/science/article/pii/S153204641500074X

[11] N. Ellouze, S. Rekhis, N. Boudriga, M. Allouche, Powerless security for cardiac implantable medical devices: Use of wireless identification and sensing platform, Journal of Network and Computer Applications 107 (2018) 1 – 21. doi:https://doi.org/10.1016/j.jnca.2018.01.009.
URL http://www.sciencedirect.com/science/article/pii/S1084804518300237

[12] M. G. Chvez-ngeles, The ecological semantics of the iomt: Modelling cyborgs networks for health policy, Informatics in Medicine Unlocked 12 (2018) 138 – 142. doi:https://doi.org/10.1016/j.imu.2018.04.005.
URL http://www.sciencedirect.com/science/article/pii/S2352914818300431

[13] L. Haoyu, L. Jianxing, N. Arunkumar, A. F. Hussein, M. M. Jaber, An iomt cloud-based real time sleep apnea detection scheme by using the spo2 estimation supported by heart rate variability, Future Generation Computer Systemsdoi:https://doi.org/10.1016/j.future.2018.12.001.
URL http://www.sciencedirect.com/science/article/pii/S0167739X18326980

[14] A. H. Sodhro, S. Pirbhulal, A. K. Sangaiah, Convergence of iot and product lifecycle management in medical health care, Future Generation Computer Systems 86 (2018) 380 – 391.

[15] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, K.-S. Kwak, The internet of things for health care: a comprehensive survey, IEEE Access 3 (2015) 678–708.

[16] S. B. Baker, W. Xiang, I. Atkinson, Internet of things for smart healthcare: Technologies, challenges, and opportunities, IEEE Access 5 (2017) 26521–26544.

[17] D. Kim, K. Park, Y. Park, J.-H. Ahn, Willingness to provide personal information: Perspective of privacy calculus in iot services, Computers

in Human Behavior 92 (2019) 273 – 281. doi:https://doi.org/10.1016/j.chb.2018.11.022.
URL `http://www.sciencedirect.com/science/article/pii/S0747563218305570`

[18] W. Wu, S. Pirbhulal, A. K. Sangaiah, S. C. Mukhopadhyay, G. Li, Optimization of signal quality over comfortability of textile electrodes for ecg monitoring in fog computing based medical applications, Future Generation Computer Systems 86 (2018) 515 – 526.

[19] L. Gonzlez-Manzano, J. M. de Fuentes, P. Peris-Lopez, C. Camara, Encryption by heart (ebh)using ecg for time-invariant symmetric key generation, Future Generation Computer Systems 77 (2017) 136 – 148. doi:https://doi.org/10.1016/j.future.2017.07.018.
URL `http://www.sciencedirect.com/science/article/pii/S0167739X16307798`

[20] M. Wazid, A. K. Das, N. Kumar, M. Conti, A. V. Vasilakos, A novel authentication and key agreement scheme for implantable medical devices deployment, IEEE Journal of Biomedical and Health Informatics 22 (4) (2018) 1299–1309. doi:10.1109/JBHI.2017.2721545.

[21] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on information theory 29 (2) (1983) 198–208.

[22] L. Zhang, Y. Zhang, S. Tang, H. Luo, Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement, IEEE Transactions on Industrial Electronics 65 (3) (2018) 2795–2805.

[23] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, S.-S. Yeo, Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks, Multimedia Systems 21 (1) (2015) 49–60.

[24] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, N. Kumar, A robust and anonymous patient monitoring system using wireless medical sensor networks, Future Generation Computer Systems 80 (2018) 483–495.

[25] B. Blanchet, B. Smyth, V. Cheval, M. Sylvestre, Proverif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial.

[26] X. H. Le, M. Khalid, R. Sankar, S. Lee, An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare, Journal of Networks 6 (3) (2011) 355.

[27] P. Kumar, S.-G. Lee, H.-J. Lee, E-sap: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks, Sensors 12 (2) (2012) 1625–1647.

[28] Y.-F. Chang, S.-H. Yu, D.-R. Shiao, A uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care, Journal of Medical Systems 37 (2) (2013) 9902. doi:10.1007/s10916-012-9902-7.
URL `https://doi.org/10.1007/s10916-012-9902-7`

[29] D. He, S. Zeadally, An analysis of rfid authentication schemes for internet of things in healthcare environment using elliptic curve cryptography, IEEE internet of things journal 2 (1) (2015) 72–83.

[30] A. K. Das, A. Goswami, A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care, Journal of medical systems 37 (3) (2013) 9948.

[31] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, X. Li, Cryptanalysis and enhancement of anonymity preserving remote user mutual authentication and session key agreement scheme for e-health care systems, Journal of medical systems 39 (11) (2015) 140.

[32] D. Wang, P. Wang, Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks, Ad Hoc Networks 20 (2014) 1–15.

[33] F. Wu, L. Xu, S. Kumari, X. Li, An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks, Multimedia Systems 23 (2) (2017) 195–205.

[34] M. S. Farash, M. Turkanović, S. Kumari, M. Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment, Ad Hoc Networks 36 (2016) 152–176.

[35] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng, N. Kumar, Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks, Computer Networks 101 (2016) 42–62.

[36] S. Arasteh, S. F. Aghili, H. Mala, A new lightweight authentication and key agreement protocol for internet of things, in: Information Security and Cryptology (ISCISC), 2016 13th International Iranian Society of Cryptology Conference on, IEEE, 2016, pp. 52–59.

[37] Q. Jiang, S. Zeadally, J. Ma, D. He, Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks, IEEE Access 5 (2017) 3376–3392.

[38] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, J. Ma, Security analysis and improvement of bio-hashing based three-factor authentication scheme

for telecare medical information systems, Journal of Ambient Intelligence and Humanized Computing 9 (4) (2018) 1061–1073.

[39] Y. Lu, L. Li, H. Peng, Y. Yang, An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem, Journal of medical systems 39 (3) (2015) 32.

[40] C.-H. Liu, Y.-F. Chung, Secure user authentication scheme for wireless healthcare sensor networks, Computers & Electrical Engineering 59 (2017) 250–261.

[41] S. Challa, A. K. Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan, A. V. Vasilakos, An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Computers & Electrical Engineering 69 (2018) 534–554.

[42] A. Lumini, L. Nanni, An improved biohashing for human authentication, Pattern recognition 40 (3) (2007) 1057–1065.

[43] Y. J. Chin, T. S. Ong, A. B. J. Teoh, K. Goh, Integrated biometrics template protection technique based on fingerprint and palmprint feature-level fusion, Information Fusion 18 (2014) 161–174.

[44] B. Blanchet, Modeling and verifying security protocols with the applied pi calculus and proverif, Foundations and Trends in Privacy and Security 1 (1-2) (2016) 1–135. doi:10.1561/3300000004.

[45] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Transactions on Computer Systems (TOCS) 8 (1) (1990) 18–36.

[46] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, et al., The AVISPA tool for the automated validation of internet security protocols and applications, in: International conference on computer aided verification, Springer, 2005, pp. 281–285.

[47] L. Xu, F. Wu, Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care, Journal of medical systems 39 (2) (2015) 10.

[48] L. Nanni, A. Lumini, Random subspace for an improved biohashing for face authentication, Pattern Recognition Letters 29 (3) (2008) 295–300.

- We present several serious security attacks against Zhang et al. scheme (called ZZTL). Our proposed attacks include user traceability, de-synchronization, DoS and insider attacks.

- In order to increase the security level offered by ZZTL protocol, we fix all security faults found in this scheme.

- We propose a new architecture involving three main entities. We also provide the access control mechanism during the authentication phase.

- We also consider the situation where the current doctor of the patient wants to transfer her/his privileges to a new doctor (ownership transfer).

- The security of the proposed scheme is examined from a formal (ProVerif language) and informal point of view.

- The efficiency of our proposal is higher than the predecessor schemes. Therefore our scheme can be used for resource-constrained sensors in IoT systems.

Seyed Farhad Aghili received his M.S. degree in Electrical Engineering from Shahid Rajaee Teacher Training University (SRTTU) in 2013. He is currently a Ph.D. candidate at the Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan. His current research interest includes RFID and IoT systems security.
e-mail: sf.aghili@eng.ui.ac.ir;
Address: Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran

Hamid Mala received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 2003, 2006 and 2011, respectively. He joined University of Isfahan (UI) in September 2011 as an Assistant Professor in the Department of Information Technology Engineering. Currently, he is with the Faculty of Computer Engineering at UI. His Research interests include design and cryptanalysis of block ciphers, digital signatures, cryptographic protocols and secure multiparty computation.

e-mail: h.mala@eng.ui.ac.ir;

Address: Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran.

Mohammad Shojafar received his PhD in Information technology from Sapienza University of Rome, Italy in 2016. He received the MSc and BSc in QIAU and Iran University Science and Technology, Tehran, Iran in 2010 and 2006, respectively. He is an Intel Innovator and senior researcher in SPRITZ Security and Privacy Research Group at the University of Padua, Italy since 2018. He was participated on European H2020 “SUPERFLUIDITY”, and some Italian projects named “SAMMClouds”, “V-FoG”, “PRIN15” aim to address some of the open network and security issues in Cloud and Fog networking. His main research interest is in the area of Network and network security and privacy. In this area, he published more than 90 papers in topmost international peer-reviewed journals and conference, e.g., IEEE TCC, IEEE TNSM, IEEE TGCN, and IEEE ICC/GLOBECOM (h-index=24, 1800+ citations). He is Associate Editor for several journals, including Cluster Computing, Ad Hoc & Sensor Wireless Networks, and editor in TJCA and TIIS. He is a Member of the IEEE.
For additional information: http://mshojafar.com

ACCEPTED MANUSCRIPT

Pedro Peris-Lopez is Visiting Lecturer at the Department of Computer Science, Universidad Carlos III de Madrid, Spain. He holds a M.Sc. in Telecommunications Engineering and Ph.D. in Computer Science. His research interests are in the field of protocols design, primitives design, lightweight cryptography, cryptanalysis etc. Nowadays, his research is focused on Radio Frequency Identification Systems (RFID) and Implantable Medical Devices (IMD). In these fields, he has published a great number of papers in specialized journals and conference proceedings. For additional information see: www.lightweightcryptography.com