Journal of
Sensor and Actuator Networks

MDPI

Article

Secure and Efficient WBAN Authentication Protocols for Intra-BAN Tier

Abdullah M. Almuhaideb [1,*] and Huda A. Alghamdi [2]

[1] SAUDI ARAMCO Cybersecurity Chair, Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[2] Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
* Correspondence: amalmuhaideb@iau.edu.sa

Abstract: Telecare medical information system (TMIS) is a technology used in a wireless body area network (WBAN), which has a crucial role in healthcare services. TMIS uses wearable devices with sensors to collect patients' data and transmit the data to the controller node via a public channel. Then, the medical server obtains the data from the controller node and stores it in the database to be analyzed. Unfortunately, an attacker can try to perform attacks via a public channel. Thus, establishing a secure mutual authentication protocol is essential for secure data transfer. Several authentication schemes have been presented to achieve mutual authentication, but there are performance limitations and security problems. Therefore, this study aimed to propose two secure and efficient WBAN authentication protocols between sensors and a mobile device/controller: authentication protocol-I for emergency medical reports and authentication protocol-II for periodic medical reports. To analyze the proposed authentication protocols, we conducted an informal security analysis, implemented BAN logic analysis, validated our proposed authentication protocol using the AVISPA simulation tool, and conducted a performance analysis. Consequently, we showed that our proposed protocols satisfy all security requirements in this study, attain mutual authentication, resist active and passive attacks, and have suitable computation and communication costs for a WBAN.

Keywords: WBAN; emergency authentication protocol; periodic authentication protocol; BAN logic; AVISPA simulation tool

Citation: Almuhaideb, A.M.; Alghamdi, H.A. Secure and Efficient WBAN Authentication Protocols for Intra-BAN Tier. *J. Sens. Actuator Netw.* **2022**, *11*, 44. https://doi.org/10.3390/jsan11030044

Academic Editor: Lei Shu

Received: 21 May 2022
Accepted: 1 August 2022
Published: 8 August 2022

1. Introduction

A WBAN is being utilized effectively in healthcare services remotely because of the fast progress of wireless communication technology. TMIS is one of the WBAN technologies that can provide a variety of healthcare services to patients remotely through telecare servers [1–3].

In the TMIS environment, patients can wear wearable devices with many sensors to continuously monitor patients' physical conditions and collect sensitive health data, such as the temperature of the body, heart rate, pressure, sugar of the blood, and other data [4,5]. The health data are transmitted to patients' mobile devices and then transferred to medical servers at any time and from any location. Thus, patients can save time and cost by utilizing numerous healthcare services remotely. Due to these advantages, TMIS offers better healthcare services compared to traditional healthcare services [6]. However, despite the advantages of TMIS, sensitive medical data concerning patients must be protected from malicious attacks as they are transmitted through unsecured channels. Thus, secure mutual authentication is essential for secure data transmission [7].

The transmitted messages include emergency medical reports and periodic medical reports. The emergency medical report occurs when a sensor detects an emergency in the body of a patient, which is needed to be sent as soon as the emergency is detected. The

periodic medical report occurs when the sensor nodes are requested to collect the patient's health data and send them to take an appropriate diagnosis at a specific time.

In this paper, we propose two WBAN authentication protocols for the intra-BAN tier: authentication protocol-I for emergency medical reports and authentication protocol-II for periodic medical reports. We conducted an informal security analysis to show that the proposed authentication protocols satisfy all security requirements in this study. Moreover, we implemented BAN logic to evaluate our proposed authentication protocols and ensure they attain mutual authentication. In addition, the AVISPA simulation tool was used to demonstrate that our proposed protocols resist active and passive attacks. Moreover, we conducted a performance analysis by comparing our proposed authentication protocols' computation and communication costs with related protocols.

The rest of the paper is organized as follows. Section 2 presents the related work, and Section 3 presents the problem statement and the proposed scheme, whereas Section 4 outlines the security analysis of the proposed authentication protocols. In Section 5, a performance analysis of the proposed protocols and related protocols is demonstrated. Finally, Section 6 presents the conclusion.

2. Related Works

WBANs deal with patients' sensitive health data, which must be safeguarded against cyberattacks. Many researchers have been interested in proposing WBAN authentication protocols to protect patients' sensitive data transmitted over insecure channels.

2.1. Overview of The System Model

The WBAN includes three tiers [8], as shown in Figure 1:

- The first tier is "Intra-BAN" The communication in this tier is between sensor nodes and a controller node. Sensors monitor and collect the patient's data, which are then transmitted via a public channel to the controller node/local server/mobile device.
- The second tier is "Inter-BAN". The communication in this tier is between a controller node/mobile device and a remote medical server. The controller node gathers data from sensor nodes and then sends them to the medical server via a public channel. The medical server stores the data in the database for later analysis.
- The third tier is "Beyond-BAN". The communication in this tier is between a medical server and a medical service provider (i.e., a doctor). The medical server can be over the cloud, and the doctor can access the server's data.

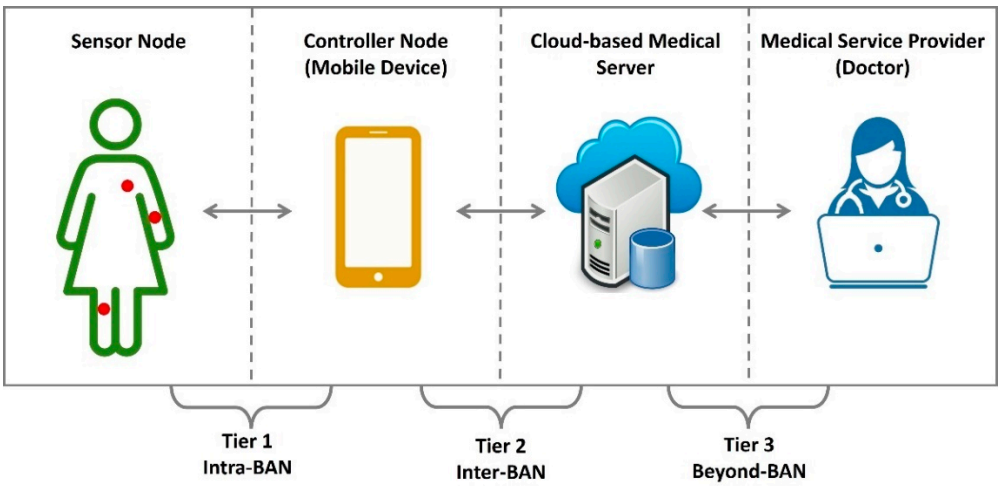

Figure 1. Overview of the WBAN system model.

2.2. Requirements of Authentication Schemes in WBAN

The authentication schemes in the WBAN must satisfy the following requirements:

- Emergency and periodic authentication protocols: The emergency authentication occurs when a sensor detects an emergency in the patient's body, and it needs to initiate the authentication request for sending the emergency report securely. Periodic authentication occurs when the controller node requests to collect the patient's data from a sensor node at a specific time, and the controller initiates the periodic authentication request to the sensor node for transmitting the data securely.
- Replay attack: An attacker can obtain messages when transmissions occur via unsecured channels. However, the attacker is unable to perform a replay attack if the message contains a timestamp.
- Session key disclosure attack: If an attacker tries to obtain the session key, the attacker cannot obtain secret values using messages sent via a public channel. Thus, the session key cannot be calculated by the attacker.
- Impersonation attack: An attacker cannot produce an authentication message to impersonate the legitimate entity.
- Controller node/mobile device stolen attack: If an attacker obtains a legitimate patient's mobile device, the attacker is unable to extract any information stored on it and is unable to generate a legitimate message.
- Off-line guessing attack: An attacker has the ability to guess either identity or a password, but not both at the same time.
- Perfect forward/backward secrecy: Future keys will not be attacked, and previous keys will not be misused (future/past key secrecy).
- Known session-specific temporary information attack: In case an attacker gets the secret values that are created randomly through the session, the session key cannot be calculated.
- Anonymity and unlinkability: This refers to an attacker being unable to obtain the identity of a legitimate entity through message eavesdropping and being unable to trace a legitimate entity using messages sent during previous sessions.
- Desynchronization attack: The solution should prevent the risk of a desynchronization attack that blocks communication between two parties and render them unable to proceed with authentication.
- Secure password change: This refers to an attacker being unable to arbitrarily change the password of a legitimate mobile device because the identity and password of the legitimate entity are unknown to the attacker.
- Performance: Authentication protocols must be cost-effective in terms of computation and communication.

2.3. Adversary Model

An adversary model's capabilities are as follows:

- The attacker has total control over all messages sent through unsecured channels. Thus, the attacker has the ability to eavesdrop, manipulate, insert, and remove messages [9].
- An attacker can steal a patient's mobile device/controller and access the data stored on it [10].
- An attacker could guess either a patient's identity (ID_i) or password (PW_i), but not both at the same time [2].
- An attacker can perform desynchronization, man-in-the-middle (MITM) attacks, impersonation attacks, replay attacks, and other possible attacks over public channels [11].
- An attacker is unable to compromise the trusted authority's private key [12].

2.4. The Existing Authentication Schemes in WBAN

The authors of [13] suggested a WBAN authentication protocol for the intra-BAN tier. Their scheme provides a group key generated by a controller node to many sensor nodes. The authentication protocol ensures forward secrecy only in the case of adding or deleting at least one sensor node where the group key is changed. However, it does not ensure forward secrecy when the sensor nodes are constant.

The scheme in [14] presented a WBAN authentication protocol for the interaction among sensor nodes and a controller device. It creates a group key between the controller device and many sensor nodes. The scheme ensures perfect forward secrecy where a new group key is generated for each session even if the sensor nodes are unchanged. However, the scheme has high communication and computation costs do not support node anonymity/unlinkability and are vulnerable to desynchronization attacks, stolen mobile device attacks, and a replay attack [15].

The authors of [16,17] suggested a lightweight WBAN authentication protocol to transmit data on a public channel securely. It relies on XOR operation and hash function to achieve low computation and communication costs. However, it presents security weaknesses such as a stolen mobile device/controller node attack, where an attacker can obtain the sensitive data within the controller device if the attacker can steal it. This allows for establishing the session key between the attacker and the sensor node.

The scheme in [18] presented an authentication protocol for the intra-BAN tier. It prevents node impersonation, MITM, and session key disclosure attacks, and it ensures forward secrecy, node anonymity, and node unlinkability. However, it has high computation and communication costs and does not prevent the risk of a desynchronization attack. The scheme adopts elliptic curve cryptography (ECC) with a point multiplication operation on the sensors and controller side, along with a hash function and XOR operation. However, the point multiplication operation is considered complex for the first tier given the resource constraints of the sensor nodes.

The authors of [19] suggested a lightweight WBAN authentication scheme to transmit sensitive data on a public channel securely. It relies on XOR operation and hash function to enhance performance. In addition, it creates biometric keys by extracting features from physiological signals, such as ECG signals. However, it presents security weaknesses such as a stolen controller device attack. If an attacker steals the controller device, the attacker can extract the secret key of the controller node, the secret key of the sensor node, and the identity of the sensor node, which represents the secret information. Thus, the session key between the attacker and a sensor node may be established.

The authors of [20] suggested a WBAN authentication protocol for the intra-BAN tier. The scheme has suitable computation and communication costs for a WBAN. Moreover, it provides some security features, such as protection from replay attacks, session key disclosure attacks, impersonation attacks, and desynchronization attacks and it ensures perfect forward/backward secrecy and node anonymity/unlinkability. However, it is prone to a stolen mobile device/controller node attack. If an attacker steals the controller device, the attacker can obtain the secret key of the controller device and the sensor node's identity and then compute the sensor node's secret key.

Ding et al. [15] and Abiramy and Sudha [21] (pp. 287–296) proposed a WBAN authentication protocol for interaction between sensor nodes and a controller node. The controller node can create a session key and distribute it to the sensor nodes in a group, which means all sensor nodes in the same group can use the same session key with the controller node. In addition, the schemes ensure perfect forward secrecy.

The scheme in [22] worked on establishing mutual authentication for the intra-BAN tier. The scheme prevents node impersonation, man-in-the-middle, and desynchronization attacks and it ensures forward/backward secrecy, node anonymity, and node unlinkability.

The authors of [23] suggested an authentication protocol for the intra-BAN tier. The scheme ensures forward/backward secrecy, node anonymity, and node unlinkability.

The scheme in [24] proposed an authentication protocol for the intra-BAN tier. Their scheme achieves integrity, confidentiality, authentication, and access control over sensitive data.

In summary, we concluded that the existing schemes did not satisfy all the solution requirements in this study, where most schemes focus on proposing secure authentication protocols but still present performance limitations or vice versa. Moreover, all existing schemes do not consider two types of authentication protocols. The first one occurs when a sensor detects an emergency in the patient's body and needs to initiate an authentication

to send the emergency medical report as soon as the emergency is detected. In contrast, the second one occurs when the controller node needs to initiate an authentication to collect the patient's data from sensor nodes at specific times. Furthermore, most schemes were designed with the assumption that a patient's mobile device/controller is trusted, but in reality, an attacker can steal the patient's mobile device and extract the sensitive information stored on it. As a result, they did not protect against the risk of a stolen mobile device/controller attack. Based on analyzing the previous WBAN authentication protocols, it was found that working on improving the existing schemes may lead to secure and efficient authentication protocols in a WBAN.

3. Problem Statement and Proposed Scheme

The public wireless network environment of a WBAN provides a significant security concern in terms of ensuring that only permitted entities have access to patients' sensitive health data. Unauthorized access may result in interception, interruption, or modification of sensitive data that may threaten a patient's life [25]. Several authentication schemes have been presented to achieve secure authentication and establish a session key, where the session key is utilized to encrypt the sensitive data transmitted through the insecure channel. Nonetheless, there are still performance limitations and security problems, such as impersonation attacks, desynchronization attacks, and other possible attacks over unsecured channels [26]. Our study will answer the following question:

- How can we achieve secure and efficient WBAN authentication protocols for the intra-BAN tier?

Therefore, we proposed WBAN authentication protocols for securing the communication between sensor nodes and a controller node. The sensor nodes (SN) work as data collectors for the patient body, and the controller node (CN) works as a local server for data collection from sensor nodes. The proposed scheme includes an initialization phase, registration phase, authentication protocol-I, authentication protocol-II, and changing password protocol. Table 1 presents the notations that are used in our proposed protocols.

Table 1. Notations that are used in the proposed protocols.

Notation	Description
P_i	i-th patient
CN	Controller node of P_i
SN	Sensor node-i of P_i
TA	Trusted authority
ID_i, PW_i	Identity and password of P_i
ID_{SN}	Identity of SN
HID_i	Masked identity of P_i
SID_i	Secret identity of P_i
S_{SN}	Secret key of SN
S_{TA}, PK_{TA}	Secret key and public key of TA
a_i, b_i, ${u_{SN}}^+$, r_i	CN-generated random numbers
x_{SN}	SN-generated random number
u_{SN}	TA-generated random number
V_{SN}, W_{SN}, ${V_{SN}}^+$, ${W_{SN}}^+$	Data to check message synchronization
RE_{SN}	Data used in protocol-II for ID_{SN} retrieval and SN authentication
HPW_i, AP_i, BP_i, CP_i, DP_i	Data used by CN to authenticate P_i
T_n	Timestamp n
${T_n}^*$	The time of message receipt
ΔT	The maximum transmission delay
X_{SN}	Data used to retrieve x_{SN} in protocol-I
L_{SN1}	Data used by CN to authenticate SN in protocol-I

Table 1. *Cont.*

Notation	Description
U_i	Data used to retrieve $u_{SN}{}^+$ in protocol-I
C	Data used to retrieve $V_{SN}{}^+$ in protocol-I
L_{i1}	Data used by *SN* to authenticate *CN* in protocol-I
R_i	Data used to retrieve r_i in protocol-II
L_{i2}	Data used by *SN* to authenticate *CN* in protocol-II
SK-I	Session key for protocol-I
SK-II	Session key for protocol-II
q	Large prime number
G_1	An additive group of order q
P	A generator of the group G_1
h	Hash function
$Z_q{}^*$	The nonzero positive integers' modulus q
$\|\|$	Concatenation operation
*	Scalar multiplication operation
$\oplus$	XOR operation
- - - ▶	Secure communication channel
⟶	Public communication channel

3.1. Initialization Phase

During this phase, *TA* creates the system's parameters as well as its private and public keys:

1. *TA* chooses an additive group G_1 of prime order q and a generator P of the group G_1.
2. *TA* selects a secure hash function h:{0,1} $\rightarrow Z_q$.
3. *TA* generates a secret random number $S_{TA} \in Z_q{}^*$ as its private key and calculates its public key $PK_{TA} = S_{TA} * P$ where $S_{TA} * P$ denotes the scalar multiplication operation of the point P in G_1.
4. *TA* publishes the system's parameters (G_1, PK_{TA}, P, q, h) and keeps S_{TA} as a private key.

3.2. Registration Phase

As shown in Figure 2, the *CN* and *SN* register with *TA* as follows:

1. P_i chooses ID_i and PW_i and then creates a number $a_i \in Z_q{}^*$. P_i calculates $HID_i = h\ (ID_i \ || \ a_i)$ and sends (HID_i) to *TA* securely. *TA* calculates $SID_i = (HID_i * S_{TA}) * PK_{TA}$ and then stores HID_i in secure memory.
2. *TA* assigns a unique ID_{SN} for each *SN* and then creates a number $u_{SN} \in Z_q{}^*$. *TA* calculates $S_{SN} = h(ID_{SN} \ || \ SID_i)$, $V_{SN} = u_{SN} \oplus h(SID_i)$, $W_{SN} = ID_{SN} \oplus h(u_{SN})$, and $RE_{SN} = ID_{SN} \oplus h(SID_i)$. *TA* sends ($ID_{SN}$, S_{SN}, V_{SN}) to the *SN* securely to store them in the *SN's* memory.
3. *TA* sends (SID_i, V_{SN}, W_{SN}, RE_{SN}) to the *CN* securely. The *CN* generates a random number $b_i \in Z_q{}^*$ and then calculates $HPW_i = h(ID_i\ PW_i \ || \ a_i)$, $AP_i = h(ID_i \ || \ PW_i) \oplus a_i$, $BP_i = HPW_i \oplus b_i$, $CP_i = SID_i \oplus b_i * P$, and $DP_i = h(a_i \ || \ b_i \ || \ HPW_i \ || \ SID_i)$. The *CN* stores ($AP_i$, BP_i, CP_i, DP_i, V_{SN}, W_{SN}, RE_{SN}) in its memory.

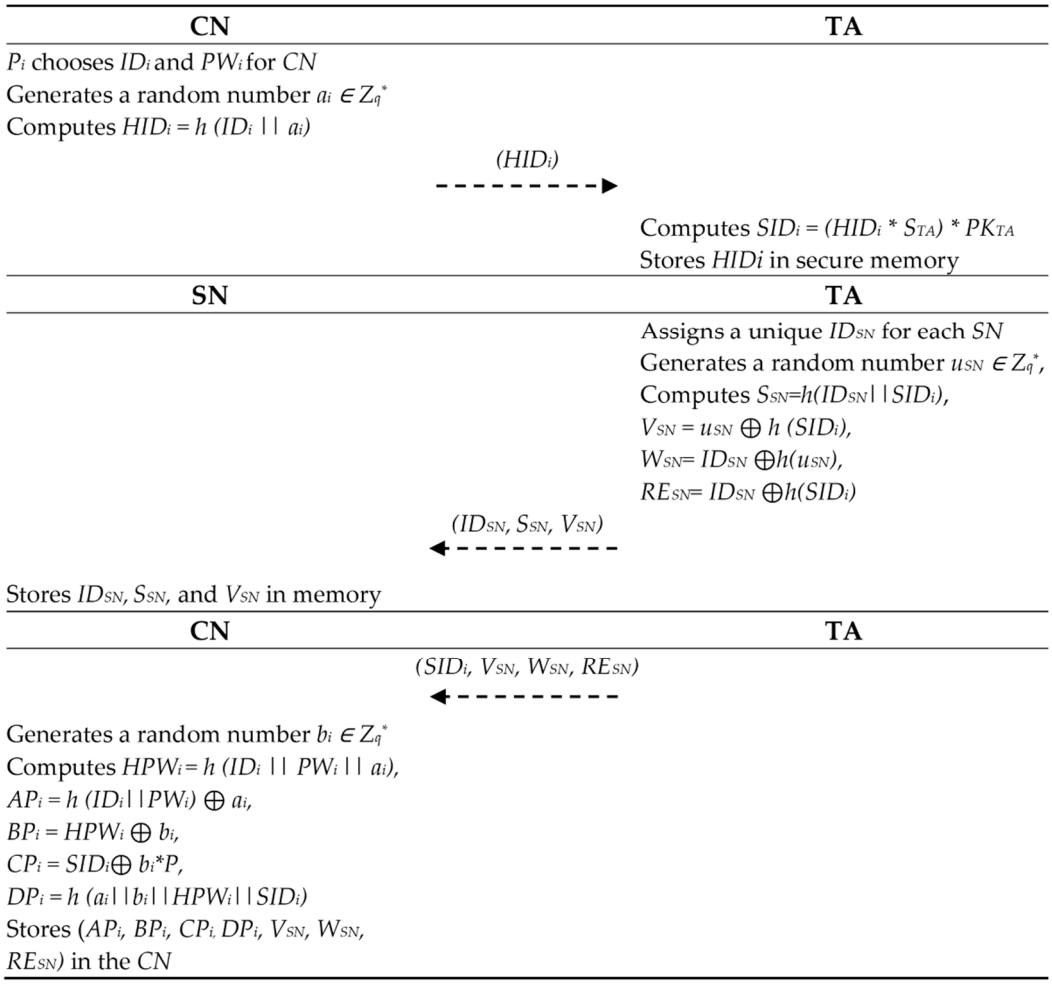

Figure 2. Registration phase.

3.3. Authentication Protocol-I

This authentication protocol is for emergency medical reports. When a sensor detects an emergency in the patient's body, the sensor node initiates the emergency authentication request to the controller node for sending the report securely. Figure 3 shows the authentication protocol between the *SN* and *CN* and contains the following steps:

1. *SN* creates a secret number $x_{SN} \in Zq^*$ and a current timestamp T_1. The *SN* computes $X_{SN} = h(S_{SN}) \oplus x_{SN}$ and $L_{SN1} = h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$. Afterwards, the *SN* transmits the message (V_{SN}, L_{SN1}, X_{SN}, T_1) to the *CN* via an unsecured channel.
2. When (V_{SN}, L_{SN1}, X_{SN}, T_1) is received, P_i enters ID_i and PW_i to the *CN*. Then, the *CN* computes $a_i = AP_i \oplus h(ID_i \,||\, PW_i)$, $HPW_i = h(ID_i \,||\, PW_i \,||\, a_i)$, $b_i = HPW_i \oplus BP_i$, and $SID_i = CP_i \oplus b_i * P$. Next, the *CN* checks to see if $DP_i \stackrel{?}{=} h(a_i \,||\, b_i \,||\, HPW_i \,||\, SID_i)$. If so, P_i is logged into the *CN* successfully.
3. The *CN* checks the validity of the timestamp, i.e., if $|T_1 - T_1^*| < \Delta T$, where T_1^* denotes the time of message receipt and ΔT denotes the longest possible transmission delay, then the *CN* retrieves W_{SN} of V_{SN} from its memory and then computes $u_{SN} = V_{SN} \oplus h(SID_i)$, $ID_{SN} = W_{SN} \oplus h(u_{SN})$, $S_{SN} = h(ID_{SN} \,||\, SID_i)$, $x_{SN} = h(S_{SN}) \oplus X_{SN}$, and $L_{SN1}^* = h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$. The *CN* checks whether $L_{SN1}^* \stackrel{?}{=} L_{SN1}$. If so, then the *SN* is authenticated. Next, the *CN* creates a secret random number $u_{SN}^+ \in Z_q^*$ and the current timestamp T_2. Afterwards, the *CN* computes $V_{SN}^+ = u_{SN}^+ \oplus h(SID_i)$, $W_{SN}^+ = ID_{SN} \oplus h(u_{SN}^+)$, $U_i = h(S_{SN}) \oplus u_{SN}^+$, $SK\text{-}I = h(ID_{SN} \,||\, S_{SN} \,||\, x_{SN} \,||\, u_{SN}^+ \,||\, V_{SN})$, and $C = V_{SN}^+ \oplus u_{SN}^+$. The *CN* replaces ($V_{SN}$, W_{SN})

with (V_{SN}, W_{SN}, V_{SN}^+, W_{SN}^+) and then computes $L_{i1} = h(S_{SN} || SK\text{-}I || ID_{SN} || V_{SN}^+ || T_2)$. The *CN* transmits the message (U_i, L_{i1}, C, T_2) to the *SN* through an unsecured channel.

4. When the *SN* received (U_i, L_{i1}, C, T_2), the *SN* checks the validity of the timestamps. If $|T_2 - T_2^*| < \Delta T$, where T_2^* denotes the time of message receipt, then the *SN* computes $u_{SN}^+ = U_i \oplus h(S_{SN})$, $SK\text{-}I = h(ID_{SN} || S_{SN} || x_{SN} || u_{SN}^+ || V_{SN})$, and $V_{SN}^+ = C \oplus u_{SN}^+$. The *SN* checks to see if $L_{i1} \stackrel{?}{=} h(S_{SN} || SK\text{-}I || ID_{SN} || V_{SN}^+ || T_2)$. If so, it replaces ($V_{SN}$) with ($V_{SN}^+$) in its memory, and the session key is established between the *SN* and *CN*.

SN		CN
Generates a random number $x_{SN} \in Z_q^*$, Current timestamp T_1 Computes: $X_{SN}=h(S_{SN}) \oplus x_{SN}$, $L_{SN1}=h(ID_{SN}\|\|x_{SN}\|\|S_{SN}\|\|V_{SN}\|\|T_1)$	$(V_{SN}, L_{SN1}, X_{SN}, T_1)$ →	
		Inputs ID_i and PW_i to *CN* *CN* computes $a_i = AP_i \oplus h(ID_i\|\|PW_i)$, $HPW_i = h(ID_i\|\|PW_i\|\|a_i)$, $b_i = HPW_i \oplus BP_i$, $SID_i = CP_i \oplus b_i * P$ Checks $DP_i \stackrel{?}{=} h(a_i\|\|b_i\|\|HPW_i\|\|SID_i)$ If yes, P_i is logged into *CN* successfully. Is $\|T_1 - T_1^*\| < \Delta T$? If so, retrieves W_{SN} of V_{SN} from its memory Computes: $u_{SN} = V_{SN} \oplus h(SID_i)$, $ID_{SN} = W_{SN} \oplus h(u_{SN})$, $S_{SN}=h(ID_{SN}\|\|SID_i)$, $x_{SN} = h(S_{SN}) \oplus X_{SN}$, and $L_{SN1}^*=h(ID_{SN}\|\|x_{SN}\|\|S_{SN}\|\|V_{SN}\|\|T_1)$ Checks if $L_{SN1}^* \stackrel{?}{=} L_{SN1}$, then *SN* is authenticated. Generates a random number $u_{SN}^+ \in Z_q^*$ Current timestamp T_2 Computes: $V_{SN}^+ = u_{SN}^+ \oplus h(SID_i)$, $W_{SN}^+ = ID_{SN} \oplus h(u_{SN}^+)$, $U_i = h(S_{SN}) \oplus u_{SN}^+$, $SK\text{-}I = h(ID_{SN}\|\|S_{SN}\|\|x_{SN}\|\|u_{SN}^+\|\|V_{SN})$, $C = V_{SN}^+ \oplus u_{SN}^+$, Replaces (V_{SN}, W_{SN}) with $(V_{SN}, W_{SN}, V_{SN}^+, W_{SN}^+)$ Computes: $L_{i1} = h(S_{SN} \|\| SK\text{-}I \|\| ID_{SN}\|\|V_{SN}^+\|\|T_2)$
	← (U_i, L_{i1}, C, T_2)	
Is $\|T_2 - T_2^*\| < \Delta T$? If so, computes: $u_{SN}^+ = U_i \oplus h(S_{SN})$, $SK\text{-}I = h(ID_{SN}\|\|S_{SN}\|\|x_{SN}\|\|u_{SN}^+\|\|V_{SN})$, $V_{SN}^+ = C \oplus u_{SN}^+$ Checks $L_{i1} \stackrel{?}{=} h(S_{SN}\|\|SK\text{-}I\|\|ID_{SN}\|\|V_{SN}^+\|\|T_2)$. If so, replaces (V_{SN}) with (V_{SN}^+), and the session key is established.		

Figure 3. Authentication protocol-I.

3.4. Authentication Protocol-II

This authentication protocol is for periodic medical reports. When the controller node requests to collect the patient's data from a sensor node at a specific time, the controller initiates the periodic authentication request to the sensor node for transmitting the data securely, as shown in Figure 4, and contains the following steps:

1. P_i inputs ID_i and PW_i to the CN. Then, the CN computes $a_i = AP_i \oplus h(ID_i \,||\, PW_i)$, $HPW_i = h(ID_i \,||\, PW_i \,||\, a_i)$, $b_i = HPW_i \oplus BP_i$, and $SID_i = CP_i \oplus b_i * P$. Next, the CN checks to see if $DP_i \stackrel{?}{=} h(a_i \,||\, b_i \,||\, HPW_i \,||\, SID_i)$. If so, P_i is logged into the CN successfully.
2. The CN retrieves ID_{SN} from its secure memory, where $ID_{SN} = RE_{SN} \oplus h(SID_i)$, and computes $S_{SN} = h(ID_{SN} \,||\, SID_i)$. The CN creates a secret number $r_i \in Zq^*$ and current timestamp T_1 and then computes $R_i = h(S_{SN}) \oplus r_i$, $SK\text{-}II = h(ID_{SN} \,||\, S_{SN} \,||\, r_i)$, and $L_{i2} = h(S_{SN} \,||\, SK\text{-}II \,||\, ID_{SN} \,||\, T_1)$. Afterwards, the CN transmits the message (R_i, L_{i2}, T_1) to the SN via a public channel.
3. When (R_i, L_{i2}, T_1) is received from the CN, the SN checks the validity of the timestamps. If $|T_1 - T_1^*| < \Delta T$, where T_1^* denotes the time of message receipt, then the SN computes $r_i = h(S_{SN}) \oplus R_i$ and $SK\text{-}II = h(ID_{SN} \,||\, S_{SN} \,||\, r_i)$. The SN checks to see if $L_{i2} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}II \,||\, ID_{SN} \,||\, T_1)$. If so, the session key is established between the CN and SN.

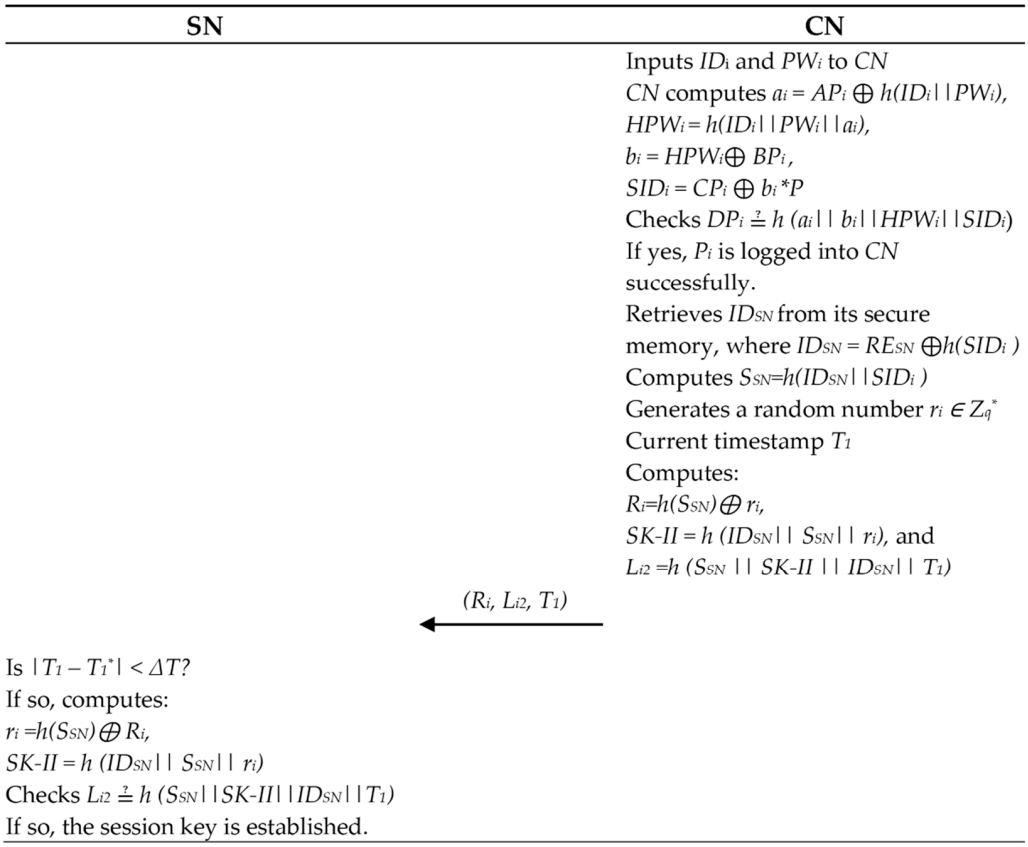

Figure 4. Authentication protocol-II.

3.5. Password Change Protocol

This protocol provides a secure changing of the password when P_i wants to change the old password of the CN, as shown in Figure 5, and contains the following steps:

1. P_i inputs ID_i and PW_i in the CN.

2. The *CN* computes $a_i = AP_i \oplus h(ID_i \,||\, PW_i)$, $HPW_i = h(ID_i \,||\, PW_i \,||\, a_i)$, $b_i = HPW_i \oplus BP_i$, and $SID_i = CP_i \oplus b_i * P$. Next, the *CN* checks to see if $DP_i \stackrel{?}{=} h(a_i \,||\, b_i \,||\, HPW_i \,||\, SID_i)$. If so, the *CN* asks P_i for a new password.
3. P_i inputs a new password PW_i^+.
4. The *CN* calculates $HPW_i^+ = h(ID_i \,||\, PW_i^+ \,||\, a_i)$, $AP_i^+ = h(ID_i \,||\, PW_i^+) \oplus a_i$, $BP_i^+ = HPW_i^+ \oplus b_i$, $CP_i = SID_i \oplus b_i * P$, and $DP_i^+ = h(a_i \,||\, b_i \,||\, HPW_i^+ \,||\, SID_i)$. Finally, the *CN* replaces (AP_i, BP_i, CP_i, DP_i, V_{SN}, W_{SN}, RE_{SN}) with (AP_i^+, BP_i^+, CP_i, DP_i^+, V_{SN}, W_{SN}, RE_{SN}) in the *CN*.

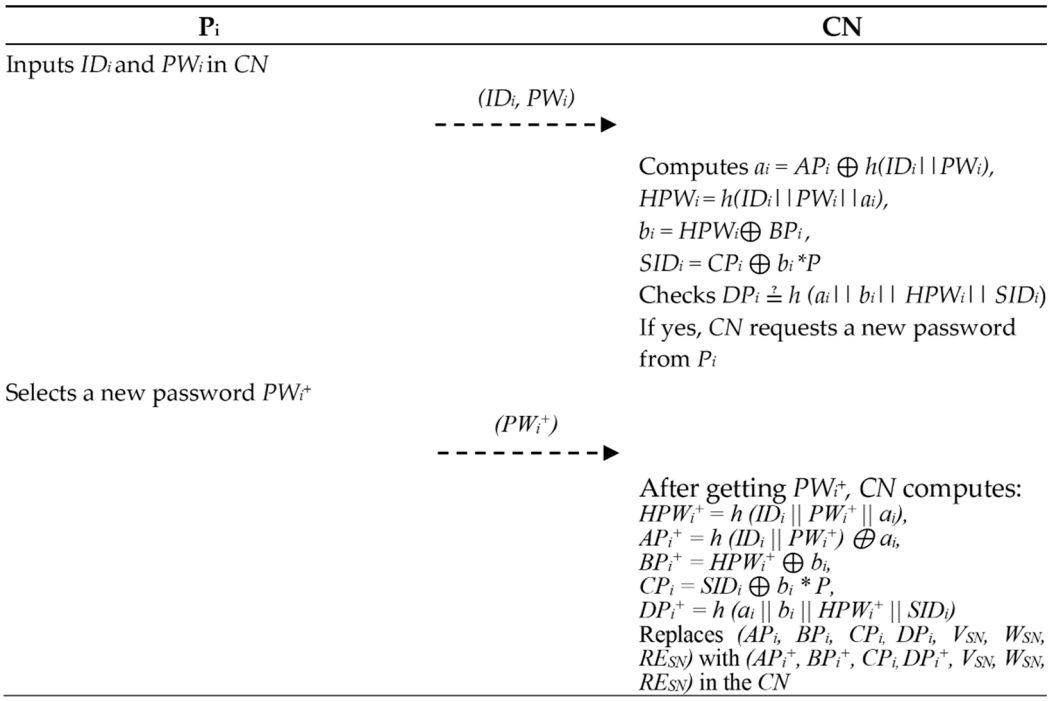

Figure 5. Change password protocol.

4. Security Analysis

Both informal and formal security analyses were conducted to show that our authentication protocols satisfy security requirements.

4.1. Informal Security Analysis

This section discusses the proposed protocols in connection to the aforementioned solution requirements in Section 2.2.

4.1.1. Emergency and Periodic Authentication Protocols

1. There are two proposed authentication protocols: the first protocol is for emergency reports, and the second protocol is for periodic reports. According to protocol-I, when a sensor detects an emergency in the patient's body, the sensor node initiates the emergency authentication request by sending the message $M1 = (V_{SN}, L_{SN1}, X_{SN}, T_1)$ to the CN. An attacker cannot generate a legal L_{SN1} because it is computed using the secret key S_{SN}. Thus, the CN authenticates the SN by checking $L_{SN1} \stackrel{?}{=} h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$. Then, the CN responds to the authentication by sending the message $M2 = (U_i, L_{i1}, C, T_2)$ to the SN. An attacker is unable to generate a valid L_{i1}, so the SN authenticates the CN by checking $L_{i1} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}I \,||\, ID_{SN} \,||\, V_{SN}^+ \,||\, T_2)$. Therefore, the SN and CN can authenticate each other.
2. The authentication protocol-II occurs when the controller node requests to collect the patient's data from a sensor node at a specific time. Thus, there is no need to initiate

the authentication by the SN, reducing costs and enhancing performance. In this case, the CN requests authentication periodically from the SN, whose identity ID_{SN} is stored in the CN. The CN initiates the authentication by sending the message $M1 = (R_i, L_{i2}, T_1)$ to the SN. An attacker cannot generate a legal L_{i2} because it is computed using the secret key S_{SN}. Thus, the SN authenticates the CN by checking $L_{i2} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}II \,||\, ID_{SN} \,||\, T_1)$. Therefore, the CN and SN can authenticate each other.

4.1.2. Replay and MITM Attacks

We presupposed in the model of the adversary that an attacker could obtain messages when transmissions occurred via unsecured channels. However, the attacker is unable to perform replay and MITM attacks on our protocols because each message sent through a public channel contains a timestamp. For protocol-I, timestamp 1 is created by the SN and contained in the hash value $L_{SN1} = h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$. Timestamp 2 is generated by the CN and contained in the hash value $L_{i1} = h(S_{SN} \,||\, SK\text{-}I \,||\, ID_{SN} \,||\, {V_{SN}}^+ \,||\, T_2)$. An attacker is unable to forge the values ID_{SN}, x_{SN}, and S_{SN} for L_{SN1} and S_{SN}, ID_{SN}, and ${V_{SN}}^+$ for L_{i1}.

For protocol-II, timestamp 1 is generated by the CN and included in the hash value $L_{i2} = h(S_{SN} \,||\, SK\text{-}II \,||\, ID_{SN} \,||\, T_1)$. An attacker cannot tamper with the values S_{SN} and ID_{SN} for L_{i2}. Therefore, the proposed protocols are resistant to such attacks.

4.1.3. Session Key Disclosure Attack

This security requirement is intended to ensure that if an attacker attempts to obtain the session key, the attacker cannot obtain secret values using messages sent via a public channel. For protocol-I, if an attacker wants to obtain the session key *SK-I*, the attacker must first obtain ID_{SN}, S_{SN}, x_{SN}, and ${u_{SN}}^+$. However, the attacker cannot obtain these values through messages sent over a public channel to compute SK-I.

For protocol-II, if an attacker tries to obtain the session key SK-II, the attacker must first obtain ID_{SN}, S_{SN}, and r_i. However, the attacker cannot obtain these values through the message sent via a public channel. Thus, the attacker cannot obtain the session keys.

4.1.4. Impersonation Attack

This security requirement aims to ensure that an attacker cannot produce an authentication message to impersonate a legitimate entity. For protocol-I, an attacker must generate an authentication message (V_{SN}, L_{SN1}, X_{SN}, T_1) to impersonate the genuine SN. However, the attacker is unable to calculate a legal L_{SN1} because it is computed using the secret key $S_{SN} = h(ID_{SN} \,||\, SID_i)$, and the attacker cannot compute a legal S_{SN} because it is computed using the secret identity $SID_i = (HID_i * S_{TA}) * PK_{TA}$, where we assumed in the adversary model that an attacker cannot compromise the TA's private key. Thus, the CN checks $L_{SN1} \stackrel{?}{=} h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$. If they do not match, then the attacker is locked out by the CN. Moreover, the attacker must generate an authentication message (U_i, L_{i1}, C, T_2) to impersonate the genuine CN. However, the attacker is unable to calculate a legal L_{i1}. Thus, the SN checks $L_{i1} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}I \,||\, ID_{SN} \,||\, {V_{SN}}^+ \,||\, T_2)$. If they do not match, then the attacker is locked out by the SN.

For protocol-II, an attacker must generate an authentication message (R_i, L_{i2}, T_1) to impersonate the legitimate CN. However, the attacker cannot calculate a legal L_{i2} because it is calculated using the secret key S_{SN}, and the secret key is computed using the secret identity SID_i. Thus, the SN checks $L_{i2} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}II \,||\, ID_{SN} \,||\, T_1)$. If they do not match, then the attacker is locked out by the SN. Therefore, our protocols are resistant to such attacks.

4.1.5. Mobile Device Stolen Attack

We assumed in the model of adversary that the mobile device/controller of the genuine P_i can be stolen by an attacker. However, for protocol-I, when the attacker receives an

authentication message from the SN, the session key cannot be established between the SN and the attacker because the attacker cannot extract P_i's ID_i and PW_i and cannot compute SID_i.

For protocol-II, the attacker cannot create a valid authentication message because the attacker cannot extract P_i's ID_i and PW_i and cannot compute SID_i. As a result, the proposed protocols are resistant to such attacks.

4.1.6. Off-Line Guessing Attack

We presupposed in the model of the adversary that an attacker could guess either P_i's ID_i or PW_i but not both at the same time. For protocol-I and protocol-II, the attacker cannot compute $a_i = AP_i \oplus h(ID_i \,||\, PW_i)$ without correctly guessing both ID_i and PW_i at the same time. Thus, the attacker cannot compute HPW_i, b_i, and SID_i. Therefore, the proposed protocols are resistant to such attacks.

4.1.7. Perfect Forward/Backward Secrecy

This security service aims to guarantee that if an attacker obtains any session key, this should not impact the secrecy of future/past session keys. For authentication protocol-I, the session key cannot be calculated by the attacker $SK\text{-}I = h(ID_{SN} \,||\, S_{SN} \,||\, x_{SN} \,||\, {u_{SN}}^{+} \,||\, V_{SN})$ because the attacker cannot obtain x_{SN} and ${u_{SN}}^{+}$, which are secret random numbers.

For authentication protocol-II, the session key cannot be calculated by the attacker $SK\text{-}II = h(ID_{SN} \,||\, S_{SN} \,||\, r_i)$ because it is a dynamic that involves a secret random number r_i. Thus, our protocols guarantee forward/backward secrecy, as a new session key is created for each session.

4.1.8. Known Session-Specific Temporary Information Attack

This security requirement aims to ensure that if an attacker gets the secret values that are created randomly through the session, the session key cannot be calculated. For protocol-I, if an attacker obtains the secret values x_{SN} and ${u_{SN}}^{+}$, which are created randomly during the session between the SN and CN, the attacker still cannot compute $S_{SN} = h(ID_{SN} \,||\, SID_i)$ without obtaining $SID_i = (HID_i * S_{TA}) * PK_{TA}$. Thus, the session key cannot be calculated by the attacker $SK\text{-}I = h(ID_{SN} \,||\, S_{SN} \,||\, x_{SN} \,||\, {u_{SN}}^{+} \,||\, V_{SN})$.

For protocol-II, if an attacker obtains the random number r_i generated during the session between the CN and SN, the attacker still cannot compute S_{SN} without obtaining SID_i. Thus, the session key cannot be calculated by the attacker $SK\text{-}II = h(ID_{SN} \,||\, S_{SN} \,||\, r_i)$. Therefore, the proposed protocols are resistant to such attacks.

4.1.9. Node Anonymity and Untraceability

For protocol-I, the messages transmitted in the authentication, i.e., $M1 = (V_{SN}, L_{SN1}, X_{SN}, T_1)$ and $M2 = (U_i, L_{i1}, C, T_2)$, are updated during each session because the authentication depends on secret random numbers x_{SN} and ${u_{SN}}^{+}$. Likewise, for protocol-II, the messages transmitted by the CN, i.e., $M3 = (R_i, L_{i2}, T_1)$, depend on a secret random number r_i, making the messages transmitted during the session independently. Thus, the attacker cannot obtain ID_i and ID_{SN} through eavesdropping on these messages and the attacker is unable to track a node using the messages sent during previous sessions. As a result, our protocols preserve these security features.

4.1.10. Desynchronization Attack

A desynchronization attack blocks communication between parties at a particular stage during the authentication, rendering both parties unable to update some data synchronously and proceed with authentication. The proposed authentication protocol-I prevents desynchronization attacks by updating the data (V_{SN}, W_{SN}) stored by the CN and (V_{SN}) stored by the SN during the authentication. If the attacker blocks the communication from the SN to CN, the SN needs to restart a new authentication round. If the attacker blocks the communication from the CN to SN, there are data stored in the CN (V_{SN}, W_{SN},

$V_{SN}{}^{+}$, $W_{SN}{}^{+}$) indicated as the nonupdated and updated data. When the SN sends a new authentication request to the CN using the nonupdated data (V_{SN}), the CN can still use the nonupdated data (V_{SN}, W_{SN}).

In other words, the CN and SN, respectively, can verify that the received message is synchronized by checking the equality of $L_{SN1}{}^{*} \stackrel{?}{=} h(ID_{SN} \,||\, x_{SN} \,||\, S_{SN} \,||\, V_{SN} \,||\, T_1)$ and $L_{i1} \stackrel{?}{=} h(S_{SN} \,||\, SK\text{-}I \,||\, ID_{SN} \,||\, V_{SN}{}^{+} \,||\, T_2)$ [27].

For protocol-II, if the attacker blocks the communication from the CN to SN, the CN only needs to restart a new authentication round [28,29]. Therefore, our protocols prevent the risk of a desynchronization attack.

4.1.11. Secure Password Change

Our protocols provide a secure password change when a P_i wants to change the old password. First, the P_i must input the current ID_i and PW_i in the CN to ensure that the user is the legitimate owner of the CN, where the CN checks whether $DP_i \stackrel{?}{=} h(a_i \,||\, b_i \,||\, HPW_i \,||\, SID_i)$. If so, the CN requests a new password from the P_i and then computes $HPW_i{}^{+}$, $AP_i{}^{+}$, $BP_i{}^{+}$, and $DP_i{}^{+}$. After that, the CN replaces (AP_i, BP_i, CP_i, DP_i, V_{SN}, W_{SN}, RE_{SN}) with ($AP_i{}^{+}$, $BP_i{}^{+}$, CP_i, $DP_i{}^{+}$, V_{SN}, W_{SN}, RE_{SN}) in the CN for future purposes. Thus, an attacker cannot arbitrarily change the password because the attacker does not know ID_i and PW_i. Therefore, our protocols provide a secure password change.

In summary, Table 2 shows the security features' comparison of our authentication protocols with related protocols [13,14,16–20]. As shown in Table 2, our protocols met all security requirements.

Table 2. Security features comparison.

Feature	[13]	[14]	[16]	[17]	[18]	[19]	[20]	Proposed
Emergency and periodic authentication protocols	×	×	×	×	×	×	×	✓
Replay and MITM attacks	N/A	N/A	✓	✓	✓	✓	✓	✓
Session key disclosure attack	✓	✓	N/A	N/A	✓	N/A	✓	✓
Impersonation attack	N/A	N/A	✓	✓	✓	✓	✓	✓
Mobile device stolen attack	×	×	×	×	×	×	×	✓
Off-line guessing attack	N/A	N/A	N/A	✓	N/A	N/A	N/A	✓
Perfect forward/backward secrecy	×	✓	✓	✓	✓	✓	✓	✓
Known session-specific temporary information	N/A	N/A	N/A	N/A	✓	N/A	✓	✓
Node anonymity and unlinkability	×	×	✓	✓	✓	✓	✓	✓
Desynchronization attacks	×	×	✓	✓	×	✓	✓	✓
Secure password change	×	×	×	×	×	×	×	✓

N/A = Information not available.

4.2. BAN Logic Proof

We implemented BAN logic to evaluate our proposed authentication protocols and ensure they attain mutual authentication [30].

4.2.1. Basic Notation

We used the following fundamental notation in both authentication protocol-I and protocol-II:

- P, Q: two principals.
- X_1, X_2: two statements.
- SK: the session key.
- $P \mid \equiv X_1$: P believes X_1, if X_1 is true.
- $P \lhd X_1$: P sees X_1, i.e., P receives X_1 contained within a message, but P does not necessarily believe X_1.

- P $|\sim X_1$: P once says X_1, i.e., P transmits a message including X_1. It is unclear if P sent the message lately or a long time ago, but P believes X_1 when P sent it.
- P $|\Rightarrow X_1$: P controls X_1, and P should trust X_1.
- #(X_1): X_1 is fresh, i.e., X_1 has never been sent before.
- $(X_1)_K$: X_1 is combined with K.
- P $\overset{K}{\leftrightarrow}$ Q: P and Q have the same key K.
- $\frac{P}{Q}$: if P is true, then Q is also true.

4.2.2. Inference Rules

Rule 1 (Message meaning rule)

$$MMR = \frac{P \mid \equiv P \overset{K}{\leftrightarrow} Q,\ P \lhd (X_1)_K}{P \mid \equiv Q \mid \sim X_1}$$

Rule 2 (Nonce verification rule)

$$NVR = \frac{P \mid \equiv \#(X_1), P \mid \equiv Q \mid \sim X_1}{P \mid \equiv Q \mid \equiv X_1}$$

Rule 3 (Jurisdiction rule)

$$JR = \frac{P \mid \equiv Q \mid \Rightarrow X_1, P \mid \equiv Q \mid \equiv X_1}{P \mid \equiv X_1}$$

Rule 4 (Belief rule)

$$BR = \frac{P \mid \equiv (X_1, X_2)}{P \mid \equiv X_1}$$

Rule 5 (Freshness rule)

$$FR = \frac{P \mid \equiv \#(X_1)}{P \mid \equiv \#(X_1, X_2)}$$

Rule 6 (Session key rule)

$$SKR = \frac{P \mid \equiv \#(X_1), P \mid \equiv Q \mid \equiv X_1}{P \mid \equiv P \overset{K}{\leftrightarrow} Q}$$

4.2.3. Protocol-I Goals

G1: $SN \mid \equiv (SN \overset{SK-I}{\leftrightarrow} CN)$

G2: $SN \mid \equiv CN \mid \equiv (SN \overset{SK-I}{\leftrightarrow} CN)$

G3: $CN \mid \equiv (SN \overset{SK-I}{\leftrightarrow} CN)$

G4: $CN \mid \equiv SN \mid \equiv (SN \overset{SK-I}{\leftrightarrow} CN)$

4.2.4. Protocol-I Assumptions

A_1: $CN \mid \equiv \#(T_1)$

A_2: $SN \mid \equiv \#(T_2)$

A_3: $SN \mid \equiv CN \Rightarrow (SN \overset{SK-I}{\leftrightarrow} CN)$

A_4: $CN \mid \equiv SN \Rightarrow (SN \overset{SK-I}{\leftrightarrow} CN)$

A_5: $SN \mid \equiv SN \overset{S_{SN}}{\leftrightarrow} CN$

A_6: $CN \mid \equiv SN \overset{S_{SN}}{\leftrightarrow} CN$

4.2.5. Protocol Idealized Forms

Msg_1: $SN \rightarrow CN : (V_{SN}, x_{SN},\ ID_{SN},\ T_1)_{S_{SN}}$

$$\text{Msg}_2\text{: } CN \rightarrow SN : (V_{SN}^{+}, u_{SN}^{+}, ID_{SN}, T_2)_{S_{SN}}$$

4.2.6. Protocol-I Formal Analysis

We implemented the BAN logic analysis of the proposed authentication protocol-I as below:

Step 1: D_1 is obtained from Msg_1.

$$D_1 : CN \lhd (V_{SN}, x_{SN},\ ID_{SN},\ T_1)_{S_{SN}}$$

Step 2: CN verifies the transmitted message is from SN. Applying MMR with D_1 and A_6 yields D_2.

$$\frac{CN \mid \equiv\ SN \overset{S_{SN}}{\leftrightarrow} CN,\ CN \lhd (V_{SN}, x_{SN},\ ID_{SN},\ T_1)_{S_{SN}}}{CN \mid \equiv\ SN \mid \sim\ (V_{SN}, x_{SN},\ ID_{SN},\ T_1)}$$

$$D_2 : CN \mid \equiv\ SN \mid \sim\ (V_{SN}, x_{SN},\ ID_{SN},\ T_1)$$

Step 3: CN checks whether the SN request is fresh. Applying FR with A_1 and D_2 yields D_3.

$$\frac{CN \mid \equiv\ \#(\text{T}_1)}{CN \mid \equiv\ \#(V_{SN}, x_{SN},\ ID_{SN},\ T_1)}$$

$$D_3 :\ CN \mid \equiv\ \#(V_{SN}, x_{SN},\ ID_{SN},\ T_1)$$

Step 4: CN checks whether the SN request is valid. Applying NVR with D_2 and D_3 yields D_4.

$$\frac{CN \mid \equiv\ \#(V_{SN}, x_{SN},\ ID_{SN},\ T_1), CN \mid \equiv\ SN \mid \sim\ (V_{SN}, x_{SN},\ ID_{SN},\ T_1)}{CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN},\ T_1)}$$

$$D_4 :\ CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN},\ T_1)$$

Step 5: The CN now trusts the SN and all its transmitted parameters. D_5 is obtained by applying BR using D_4.

$$\frac{CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN},\ T_1)}{CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN})}$$

$$D_5 :\ CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN})$$

Step 6: D_6 is obtained by applying SKR using D_3 and D_5 to accomplish G4.

$$\frac{CN \mid \equiv\ \#(V_{SN}, x_{SN},\ ID_{SN},\ T_1), CN \mid \equiv SN \mid \equiv (V_{SN}, x_{SN},\ ID_{SN})}{CN \mid \equiv\ SN \mid \equiv\ \left(SN \overset{SK-I}{\leftrightarrow} CN\right)}$$

$$D_6 :\ CN \mid \equiv\ SN \mid \equiv\ \left(SN \overset{SK-I}{\leftrightarrow} CN\right)$$

Step 7: CN has full control over the transmitted SN parameters. D_7 is obtained by applying JR using A_4 and D_6 to accomplish G3

$$\frac{CN \mid \equiv SN \Rightarrow (SN \overset{SK-I}{\leftrightarrow} CN), CN \mid \equiv\ SN \mid \equiv\ (SN \overset{SK-I}{\leftrightarrow} CN)}{CN \mid \equiv\ (SN \overset{SK-I}{\leftrightarrow} CN)}$$

$$D_7 :\ CN \mid \equiv\ (SN \overset{SK-I}{\leftrightarrow} CN)$$

Step 8: D_8 is obtained from Msg_2

$$D_8 : \ SN \lhd (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)_{S_{SN}}$$

Step 9: SN verifies the transmitted message is from CN. Applying MMR with D_8 and A_5 yields D_9.

$$\frac{SN \mid \equiv \ SN \overset{S_{SN}}{\leftrightarrow} CN \ , \ SN \lhd (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)_{S_{SN}}}{SN \mid \equiv \ CN \mid \ \sim \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)}$$

$$D_9 : SN \mid \equiv \ CN \mid \ \sim \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)$$

Step 10: SN checks whether CN request is fresh. Applying FR with A_2 and D_9 yields D_{10}.

$$\frac{SN \mid \equiv \#(T_2)}{SN \mid \equiv \#(V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)}$$

$$D_{10} : \ SN \mid \equiv \#(V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)$$

Step 11: SN checks whether CN request is valid. Applying NVR with D_9 and D_{10} yields D_{11}.

$$\frac{SN \mid \equiv \#(V_{SN}^+, u_{SN}^+, ID_{SN}, T_2), \ SN \mid \equiv \ CN \mid \ \sim \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)}{SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)}$$

$$D_{11} : \ SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)$$

Step 12: SN now trusts CN and all its transmitted parameters. Applying BR with D_{11} yields D_{12}.

$$\frac{SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN}, T_2)}{SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN})}$$

$$D_{12} : \ SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN})$$

Step 13: D_{13} is obtained by applying SKR using D_{10} and D_{12} to accomplish G2.

$$\frac{SN \mid \equiv \#(V_{SN}^+, u_{SN}^+, ID_{SN}, T_2), \ SN \mid \equiv \ CN \mid \equiv \ (V_{SN}^+, u_{SN}^+, ID_{SN})}{SN \mid \equiv \ CN| \ \equiv \ (SN \overset{SK-I}{\leftrightarrow} CN)}$$

$$D_{13} : \ SN \mid \equiv \ CN| \ \equiv \ (SN \overset{SK-I}{\leftrightarrow} CN)$$

Step 14: SN has the new session key's parameters from transmitted CN parameters. D_{14} is obtained by applying JR to A_3 and D_{13} to accomplish G_1.

$$\frac{SN \mid \equiv CN \Rightarrow (SN \overset{SK-I}{\leftrightarrow} CN), SN \mid \equiv \ CN| \ \equiv \ (SN \overset{SK-I}{\leftrightarrow} CN)}{SN \mid \ \equiv \ (SN \overset{SK-I}{\leftrightarrow} CN)}$$

$$D_{14} : \ SN \mid \ \equiv \ (SN \overset{SK-I}{\leftrightarrow} CN)$$

In summary of protocol-I, the SN and CN attained mutual authentication. Furthermore, the session key SK-I was established in a secure manner based on G1, G2, G3, and G4. In the following, we analyze the authentication protocol-II.

4.2.7. Protocol-II Goals

G1: $SN \ \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$

G2: $SN \ \mid \equiv CN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$

G3: $CN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$

G4: $CN \mid \equiv SN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$

4.2.8. Protocol-II Assumptions

A_1: $SN \mid \equiv \#(T_1)$

A_2: $SN \mid \equiv CN \Rightarrow (CN \overset{SK-II}{\leftrightarrow} SN)$

A_3: $CN \mid \equiv SN \Rightarrow (CN \overset{SK-II}{\leftrightarrow} SN)$

A_4: $SN \mid \equiv CN \overset{S_{SN}}{\leftrightarrow} SN$

A_5: $CN \mid \equiv SN \mid \equiv (ID_{SN})$

4.2.9. Protocol-II Idealized Forms

Msg_1: $TA \rightarrow CN : (ID_{SN})$

Msg_2: $CN \rightarrow SN : (r_i, ID_{SN}, T_1)_{S_{SN}}$

4.2.10. Protocol-II Formal Analysis

We implemented the BAN logic to analyze our proposed authentication protocol-II, as below:

Step 1: D_1 is obtained from Msg_1.

$$D_1 : CN \lhd (ID_{SN})$$

Step 2: CN trusts the transmitted parameters from TA and trusts the SN is legitimate. D_2 is obtained from A_5, $S_{SN} = h\ (ID_{SN} \mid\mid SID_i)$, $r_i = h\ (S_{SN}) \oplus R_i$, and the session key $SK\text{-}II = h\ (ID_{SN} \mid\mid S_{SN} \mid\mid r_i)$ to accomplish G4.

$$D_2 : CN \mid \equiv\ SN \mid\ \equiv\ (CN \overset{SK-II}{\leftrightarrow} SN)$$

Step 3: D_3 is obtained by applying JR using A_3 and D_2 to accomplish G3.

$$\frac{CN \mid \equiv SN \Rightarrow (CN \overset{SK-II}{\leftrightarrow} SN), CN \mid \equiv\ SN \mid\ \equiv\ (CN \overset{SK-II}{\leftrightarrow} SN)}{CN \mid\ \equiv\ (CN \overset{SK-II}{\leftrightarrow} SN)}$$

$$D_3 :\ CN \mid\ \equiv\ (CN \overset{SK-II}{\leftrightarrow} SN)$$

Step 4: D_4 is obtained from Msg_2

$$D_4 :\ SN \lhd (r_i, ID_{SN}, T_1)_{S_{SN}}$$

Step 5: SN verifies the transmitted message is from CN. Applying MMR with D_4 and A_4 yields D_5.

$$\frac{SN \mid\ \equiv\ CN \overset{S_{SN}}{\leftrightarrow} SN,\ SN \lhd (r_i, ID_{SN}, T_1)_{S_{SN}}}{SN \mid \equiv\ CN \mid\ \sim\ (r_i, ID_{SN}, T_1)}$$

$$D_5 :\ SN \mid \equiv\ CN \mid\ \sim\ (r_i, ID_{SN}, T_1)$$

Step 6: SN checks whether the CN request is fresh. Applying FR with A_1 and D_5 yields D_6.

$$\frac{SN \mid \equiv \#(T_1)}{SN \mid \equiv \#(r_i, ID_{SN}, T_1)}$$

$$D_6 :\ SN \mid \equiv \#(r_i, ID_{SN}, T_1)$$

Step 7: SN checks whether the CN request is valid. Applying NVR with D_5 and D_6 yields D_7.

$$\frac{SN \mid \equiv \#(r_i, ID_{SN}, T_1), SN \mid \equiv\ CN \mid\ \sim\ (r_i, ID_{SN}, T_1)}{SN \mid \equiv\ CN \mid \equiv\ (r_i, ID_{SN}, T_1)}$$

$$D_7 : SN \mid \equiv CN \mid \equiv (r_i, ID_{SN}, T_1)$$

Step 8: SN now trusts CN and all its transmitted parameters. Applying BR with D_7 yields D_8.

$$\frac{SN \mid \equiv CN \mid \equiv (r_i, ID_{SN}, T_1)}{SN \mid \equiv CN \mid \equiv (r_i, ID_{SN})}$$

$$D_8 : SN \mid \equiv CN \mid \equiv (r_i, ID_{SN})$$

Step 9: D_9 is obtained by applying SKR, using D_6 and D_8 to accomplish G2.

$$\frac{SN \mid \equiv \#(r_i, ID_{SN}, T_1),\ SN \mid \equiv CN \mid \equiv (r_i, ID_{SN})}{SN \mid \equiv CN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)}$$

$$D_9 : SN \mid \equiv CN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$$

Step 10: SN has the new session key's parameters from transmitted CN parameters. D_{10} is obtained by applying JR, using A_2 and D_9 to accomplish G1.

$$\frac{SN \mid \equiv CN \Rightarrow (CN \overset{SK-II}{\leftrightarrow} SN), SN \mid \equiv CN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)}{SN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)}$$

$$D_{10} : SN \mid \equiv (CN \overset{SK-II}{\leftrightarrow} SN)$$

In summary of protocol-II, the CN and SN attained mutual authentication. Furthermore, the session key *SK-II* was established in a secure manner based on G1, G2, G3, and G4.

4.3. AVISPA Simulation Tool

We used the Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool and the Security Protocol ANimator for AVISPA (SPAN) to analyze the security of the proposed authentication protocols. We showed that the simulator executed our authentication protocols entirely, proving the validity of the authentication and privacy reports generated by AVISPA's security checker module. First, we wrote the protocol-I and protocol-II codes in High-Level Protocol Specification Language (HLPSL). After that, we ran the SPAN animator to confirm that protocol-I and protocol-II were entirely executable. Finally, we ran the On-the-Fly Model Checker (OFMC) and the Constraint Logic-Based Attack Searcher (CL-AtSe) to determine whether our protocols' security goals were SAFE or UNSAFE. If the results of the OFMC and CL-AtSe models were SAFE, it meant that the protocols were secure.

Figures 6 and 7 show the HLPSL code for authentication protocol-I. Figures 8 and 9 show the HLPSL code for protocol-II. We had three agents' roles, a session role, and an environment role. The *role controller* was played by agent *CN*, the *role sensor* was played by agent *SN*, and the *role trustedauthority* was played by agent *TA*. The *role controller* header contained *SN*, *CN*, and *TA* as agents, hash function, *SKcnta* as the symmetric key used to create a secure channel, and *SND/RCV* channels of type Dolev–Yao (*dy*). The *role sensor* header contained *SN*, *CN*, and *TA* as agents, hash function, *SKsnta* as the symmetric key to generate a secure channel, and *SND/RCV* channels. The *role trustedauthority* header contained *SN*, *CN*, and *TA* as agents, hash function, *SKsnta/SKcnta* as the symmetric keys, and *SND/RCV* channels. The SN was not permitted to know *SKcnta*, and the CN did not know *SKsnta*.

File: /home/span/Desktop/authenticationSNCN1.hlpsl Page 1 of 2

```
%% Authentication Protocol-I between SN and CN
role controller (SN,CN,TA:   agent,
              SKcnta:   symmetric_key,
              H, Mul:        hash_func,
              SND,RCV:  channel(dy))
played_by CN
def=
  local State: nat,
         IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
         Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_xsn,sn_cn_t1,cn_sn_usn,cn_sn_t2   : protocol_id
  init State := 0
  transition
%%%CN registration phase
     1. State = 0 /\ RCV(start)=|>
%%%CN sends registration request to TA securely
         State':= 1
        /\ Aii':=new()/\ IDi':=new() /\ PWi':=new() /\ SND({H(IDi'.Aii')}_SKcnta) /\ secret
({IDi',PWi',Aii'},sp1,{CN})
%%%CN receives registration response from TA securely
     2. State = 1 /\  RCV({Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P)).xor(Usn',H(Mul(Mul(H(IDi.Aii).Sta').Mul
(Sta'.P)))).xor(IDsn',H(Usn')).xor(IDsn',H(Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P))))}_SKcnta)=|>
         State':= 2 /\ secret ({Sta'},sp4,{TA})
%%%CN receives authentication request from the SN via public channel
     3. State = 2 /\ RCV(xor(Usn,H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))).H(IDsn.Xxsn'.H(IDsn.Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P))).xor(Usn,H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))).T1').xor(H(H(IDsn.Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P)))),Xxsn').T1') =|> State':= 3 /\ Lsn1':=H(IDsn.Xxsn'.H(IDsn.Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P))).xor(Usn,H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))).T1')
%% CN generates SKI
      /\ Usnnew':= new() /\ T2' :=new()/\ Vsnnew':=xor(Usnnew',H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))))
      /\ Wsnnew':=xor(IDsn,H(Usnnew'))/\ Ui':= xor(H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))),Usnnew'))
      /\ SKI':=H(IDsn.H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))).Xxsn'.Usnnew'.xor(Usn,H(Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P)))))/\ Li1':=H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul
(Sta.P))).SKI'.IDsn.Vsnnew'.T2')/\C':=xor(Vsnnew',Usnnew')
      /\ SND(Ui'.Li1'.C'.T2')/\ secret({SKI'},sp5,{SN,CN})
%%%CN has freshly generated the values T2 and Usnnew for SN
      /\ witness(CN,SN,cn_sn_usn,Usnnew')/\ witness(CN,SN,cn_sn_t2,T2')
%%% CN checks that SN is the emitter of Xxsn and T1
      /\ request(CN,SN,sn_cn_xsn,Xxsn')  /\ request(CN,SN,sn_cn_t1,T1')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role sensor (SN,CN,TA:   agent,
              SKsnta:   symmetric_key,
              H, Mul:        hash_func,
              SND,RCV:  channel(dy))
played_by SN
def=
  local State: nat,
         IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
         Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_xsn,sn_cn_t1,cn_sn_usn,cn_sn_t2   : protocol_id
  init   State := 0
  transition
%%%SN registration phase
%%%SN receives registration from TA securely
     1. State = 0 /\ RCV ({IDsn'.H(IDsn'.Mul(Mul(H(IDi'.Aii').Sta').Mul(Sta'.P))).xor(Usn',H(Mul(Mul(H
(IDi'.Aii').Sta').Mul(Sta'.P))))}_SKsnta) =|> State':= 1 /\ secret ({Sta'},sp4,{TA})
%%% Authentication phase
%%% SN sends authentication request and  SKI to CN via public channel
       /\ Xxsn':=new()/\ T1':=new()/\ SND(xor(Usn',H(Mul(Mul(H(IDi'.Aii').Sta').Mul(Sta'.P)))).H
(IDsn'.Xxsn'.H(IDsn'.Mul(Mul(H(IDi'.Aii').Sta').Mul(Sta'.P))).xor(Usn',H(Mul(Mul(H(IDi'.Aii').Sta').Mul
(Sta'.P)))).T1')
.xor(H(H(IDsn'.Mul(Mul(H(IDi'.Aii').Sta').Mul(Sta'.P)))),Xxsn').T1')
%%%SN has freshly generated the values T1 and Xxsn for CN
       /\ witness(SN,CN,sn_cn_xsn,Xxsn') /\ witness(SN,CN,sn_cn_t1,T1')
```

Figure 6. HLPSL code for protocol-I.

File: /home/span/Desktop/authenticationSNCN1.hlpsl Page 2 of 2

```
%% SN recieves authentication response from CN via public channel
     2. State = 1 /\ RCV (xor(H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))),Usnnew')).H(H(IDsn.Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P))).H(IDsn.H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))).Xxsn.Usnnew'.xor(Usn,H(Mul
(Mul(H(IDi.Aii).Sta).Mul(Sta.P))))).IDsn.xor(Usnnew',H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))).T2').xor(xor
(Usnnew',H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))),Usnnew').T2') =|> State':= 2
       /\ SKI':=H(IDsn.H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))).Xxsn.Usnnew'.xor(Usn,H(Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P))))/\ Li1':=H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))).SKI'.IDsn.xor
(Usnnew',H(Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))).T2') /\ secret({SKI'},sp5,{SN,CN})
%%% SN checks that CN is the emitter of Usnnew and T2
       /\request(SN,CN,cn_sn_usn,Usnnew')/\request(SN,CN,cn_sn_t2,T2')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role trustedauthority (SN,CN,TA:agent,
            SKsnta , SKcnta :symmetric_key,
             H, Mul:hash_func,SND,RCV:channel(dy))
played_by TA
def=
  local State: nat,
        IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
        Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_xsn,sn_cn_t1,cn_sn_usn,cn_sn_t2   : protocol_id
  init   State := 0
  transition
%%%CN and SN registration phase
%%%TA receives registration request from CN securely
    1. State = 0 /\ RCV({H(IDi'.Aii')}_SKcnta)=|> State':= 1
       /\ Sta':=new()/\ PKta':=Mul(Sta'.P)/\ SIDi':= Mul(Mul(H(IDi'.Aii').Sta').PKta')
%%%TA sends registration to SN securely
       /\ IDsn' :=new()/\ Usn':=new()/\ Ssn':=H(IDsn'.SIDi')/\ Vsn':= xor(Usn',H(SIDi'))
       /\ Wsn':=xor(IDsn',H(Usn'))/\ SND({IDsn'.Ssn'.Vsn'}_SKsnta)/\ secret ({IDsn',Ssn'},sp2,
{TA,SN,CN})
%%%TA sends registration response to CN securely
       /\ REsn':=xor(IDsn',H(SIDi'))/\ SND({SIDi'.Vsn'.Wsn'.REsn'}_SKcnta)/\ secret ({SIDi'},sp3,{CN,TA})/
\ secret ({Sta'},sp4,{TA})
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session (SN,CN,TA        :agent,
              SKsnta ,SKcnta  :symmetric_key,
              H, Mul          :hash_func)
def=
  local SN1,SN2,SN3,RV1,RV2,RV3 : channel(dy)
  composition
           controller(SN,CN,TA,SKcnta,H,Mul,SN1,RV1)
        /\ trustedauthority(SN,CN,TA,SKsnta ,SKcnta,H,Mul,SN2,RV2)
        /\ sensor(SN,CN,TA,SKsnta,H,Mul,SN3,RV3)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=
  const sn,cn,ta                                                          : agent,
        sksnta , skcnta                                                   : symmetric_key,
        h,mul                                                             : hash_func,
        sn_cn_xsn, sn_cn_t1, cn_sn_usn, cn_sn_t2,sp1, sp2, sp3, sp4, sp5, sp6   : protocol_id,
        p,pkta                                                            : text
  intruder_knowledge = {sn,cn,ta,p,pkta,h,mul}
  composition
     session(sn,cn,ta,sksnta ,skcnta,h,mul)
end role
goal
  secrecy_of sp1,sp2,sp3,sp4,sp5
  authentication_on sn_cn_xsn
  authentication_on sn_cn_t1
  authentication_on cn_sn_usn
  authentication_on cn_sn_t2
end goal
environment()
```

Figure 7. HLPSL code for protocol-I.

```
File: /home/span/Desktop/authenticationSNCN2.hlpsl                Page 1 of 2

%% Authentication Protocol-II between CN and SN
role controller (SN,CN,TA:   agent,
              SKcnta:   symmetric_key,
              H, Mul:        hash_func,
              SND,RCV:  channel(dy))
played_by CN
def=
  local State: nat,
       IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
       Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_idsn,cn_sn_ri,cn_sn_t1   : protocol_id
  init State := 0
  transition
%%%CN registration phase
     1. State = 0 /\ RCV(start)=|>
%%%CN sends registration request to TA securely
    State':= 1/\Aii':=new()/\ IDi':=new()/\ PWi':=new()/\ SND({H(IDi'.Aii')}_SKcnta)/\ secret
({IDi',PWi',Aii'},sp1,{CN})
%%%CN receives registration response from TA securely
     2. State = 1 /\  RCV({Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P)).xor(Usn',H(Mul(Mul(H(IDi.Aii).Sta').Mul
(Sta'.P)))).xor(IDsn',H(Usn')).xor(IDsn',H(Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P))))}_SKcnta)=|>
     State':= 2 /\ secret ({Sta'},sp4,{TA})
%%%Authentication phase and generate SKII
         /\ Rri':=new()/\ T1':=new()/\ Ri':=xor(H(H(IDsn'.Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P)))),Rri')
         /\ SKII' := H(IDsn'. H(IDsn'.Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P))). Rri')
         /\ Li2':=H(H(IDsn'.Mul(Mul(H(IDi.Aii).Sta').Mul(Sta'.P))).SKII'.IDsn'.T1')
%%CN sends authentiocation requests to SN
          /\ SND(Ri'.Li2'.T1') /\ secret({SKII'},sp5,{SN,CN})
%%%CN has freshly generated the values T1 and Rri for SN
          /\ witness(CN,SN,cn_sn_ri,Rri')/\ witness(CN,SN,cn_sn_t1,T1')
%%% CN checks IDsn to authenticate SN
          /\ request(CN,SN,sn_cn_idsn,IDsn')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role sensor (SN,CN,TA:   agent,
              SKsnta:   symmetric_key,
              H, Mul:        hash_func,
              SND,RCV:  channel(dy))
played_by SN
def=
  local State: nat,
       IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
       Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_idsn,cn_sn_ri,cn_sn_t1    : protocol_id
  init   State := 0
  transition
%%%SN registration phase
%%%SN receives registration  from TA securely
     1. State = 0 /\ RCV ({IDsn'.H(IDsn'.Mul(Mul(H(IDi'.Aii').Sta').Mul(Sta'.P))).xor(Usn',H(Mul(Mul(H
(IDi'.Aii').Sta').Mul(Sta'.P))))}_SKsnta) =|>State':= 1/\ secret ({Sta'},sp4,{TA})
%% SN recieves authentication request from CN via public channel
     2. State = 1 /\ RCV (xor(H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P)))),Rri').H(H(IDsn.Mul(Mul(H
(IDi.Aii).Sta).Mul(Sta.P))).H(IDsn. H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))). Rri').IDsn.T1').T1') =|>
        State':= 2
        /\ SKII':=H(IDsn. H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))). Rri')/\ secret({SKII'},sp5,{SN,CN})
        /\ Li2':=H(H(IDsn.Mul(Mul(H(IDi.Aii).Sta).Mul(Sta.P))).SKII'.IDsn.T1')
%%% SN checks that CN is the emitter of Rri and T1
        /\request(SN,CN,cn_sn_ri,Rri') /\request(SN,CN,cn_sn_t1,T1')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role trustedauthority (SN,CN,TA:   agent,
             SKsnta , SKcnta :   symmetric_key,
              H, Mul:        hash_func,
              SND,RCV:  channel(dy))
played_by TA
```

Figure 8. HLPSL code for protocol-II.

```
File: /home/span/Desktop/authenticationSNCN2.hlpsl                                   Page 2 of 2

def=
  local State: nat,
         IDi,
PWi,Aii,SIDi,IDsn,Usn,Ssn,Vsn,Wsn,REsn,P,Sta,PKta,Xxsn,T1,Lsn1,Usnnew,Vsnnew,Wsnnew,SKI,Li1,T2,C,Ui:text,
         Rri, Ri, SKII, Li2 : text
  const sp1,sp2,sp3,sp4,sn_cn_idsn,cn_sn_ri,cn_sn_t1    : protocol_id
  init   State := 0
  transition
%%%CN and SN registration phase
%%%TA receives registration request from CN securely
    1. State = 0 /\ RCV({H(IDi'.Aii')}_SKcnta)=|>
       State':= 1 /\ Sta':=new()/\ PKta':=Mul(Sta'.P)/\ SIDi':= Mul(Mul(H(IDi'.Aii').Sta').PKta')
%%%TA sends registration to SN securely
       /\ IDsn' :=new()/\ Usn':=new()/\ Ssn':=H(IDsn'.SIDi')/\ Vsn':= xor(Usn',H(SIDi'))/\ Wsn':=xor
(IDsn',H(Usn'))
       /\ SND({IDsn'.Ssn'.Vsn'}_SKsnta)/\ secret ({IDsn',Ssn'},sp2,{TA,SN,CN})
%%%TA sends registration response to CN securely
       /\ REsn':=xor(IDsn',H(SIDi'))/\ SND({SIDi'.Vsn'.Wsn'.REsn'}_SKcnta)
       /\ secret ({SIDi'},sp3,{CN,TA})/\ secret ({Sta'},sp4,{TA})/\ witness(SN,CN,sn_cn_idsn,IDsn')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session (SN,CN,TA:   agent,
              SKsnta ,SKcnta :   symmetric_key,
              H, Mul:        hash_func)
def=
  local
        SN1,SN2,SN3,RV1,RV2,RV3 : channel(dy)
  composition
           controller(SN,CN,TA,SKcnta,H,Mul,SN1,RV1)
        /\ trustedauthority(SN,CN,TA,SKsnta ,SKcnta,H,Mul,SN2,RV2)
        /\ sensor(SN,CN,TA,SKsnta,H,Mul,SN3,RV3)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=
  const sn,cn,ta                                         : agent,
        skcnta, sksnta                                   : symmetric_key,
        h,mul                                            : hash_func,
        sn_cn_idsn,cn_sn_ri,cn_sn_t1                     : protocol_id,
        sp1, sp2, sp3, sp4, sp5, sp6                     : protocol_id,
        p,pkta                                           : text

  intruder_knowledge = {sn,cn,ta,p,pkta,h,mul}

  composition
     session(sn,cn,ta,sksnta,skcnta,h,mul)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
goal
  secrecy_of sp1,sp2,sp3,sp4, sp5
  authentication_on sn_cn_idsn
  authentication_on cn_sn_ri
  authentication_on cn_sn_t1
end goal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
environment()
```

Figure 9. HLPSL code for protocol-II.

Moreover, HLPSL should specify the session and environment roles. The *role session* defines the interactions between the three agents' roles, which describes a protocol session. The *role environment* includes the *intruder knowledge*, specifies the *goal*, and expresses a *com-*

position of one or more sessions. The parameters (*sp1*, *sp2*, *sp3*, *sp4*, *sp5*, *sn_cn_xsn*, *sn_cn_t1*, *cn_sn_usn*, *cn_sn_t2*) of protocol-I and parameters (*sp1*, *sp2*, *sp3*, *sp4*, *sp5*, *sn_cn_idsn*, *cn_sn_ri*, *cn_sn_t1*) of protocol-II are declared as *protocol_id* and are used as privacy and authentication checkers. In the *goal* section, the goal *secrecy_of sp1*, *sp2*, *sp3*, *sp4*, *sp5* indicates that the values of specific variables are kept secret to the specific agents. The goals *authentication_on sn_cn_xsn* and *authentication_on sn_cn_idsn* express that the *CN* authenticates the *SN* after receiving messages containing these values. The goal *authentication_on sn_cn_t1* means that the *SN* selects a timestamp *t1* and the *CN* authenticates the *SN* after receiving a message from the *SN* containing *t1*. The goals *authentication_on cn_sn_usn* and *authentication_on cn_sn_ri* mean that the *CN* generates random values and the *SN* checks that *CN* is the emitter of these values and authenticates *CN* after receiving messages from *CN* containing these values. The goals *authentication_on cn_sn_t1* and *authentication_on cn_sn_t2* indicate that the *CN* selects timestamps and the *SN* authenticates the *CN* after receiving the timestamps from the messages of the *CN*.

We used the SPAN animator to demonstrate the correct execution of our proposed protocols. Figures 10 and 11 show snapshots of the SPAN animator of protocol-I and protocol-II, respectively. As shown in Figures 10 and 11, all messages were executed and exchanged by the simulator. No message was left unexecuted.

The simulation results of our protocols using AVISPA with OFMC and CL-AtSe are shown in Figure 12. The summary reports demonstrate that the proposed protocol-I and protocol-II were SAFE and met all the security goals stated in the *role environment*.

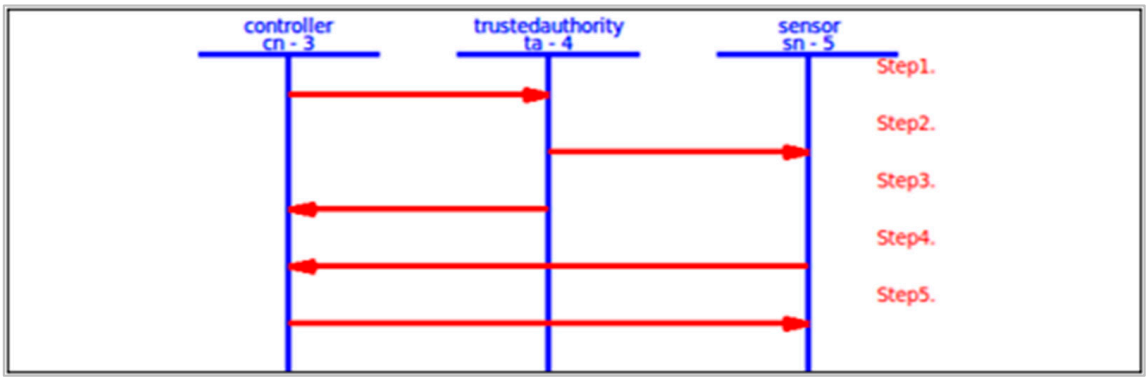

Figure 10. SPAN animator of protocol-I.

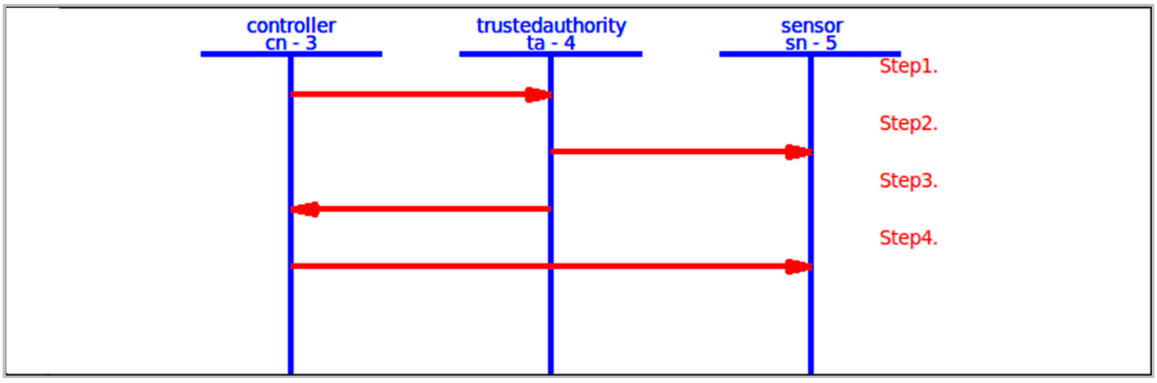

Figure 11. SPAN animator of protocol-II.

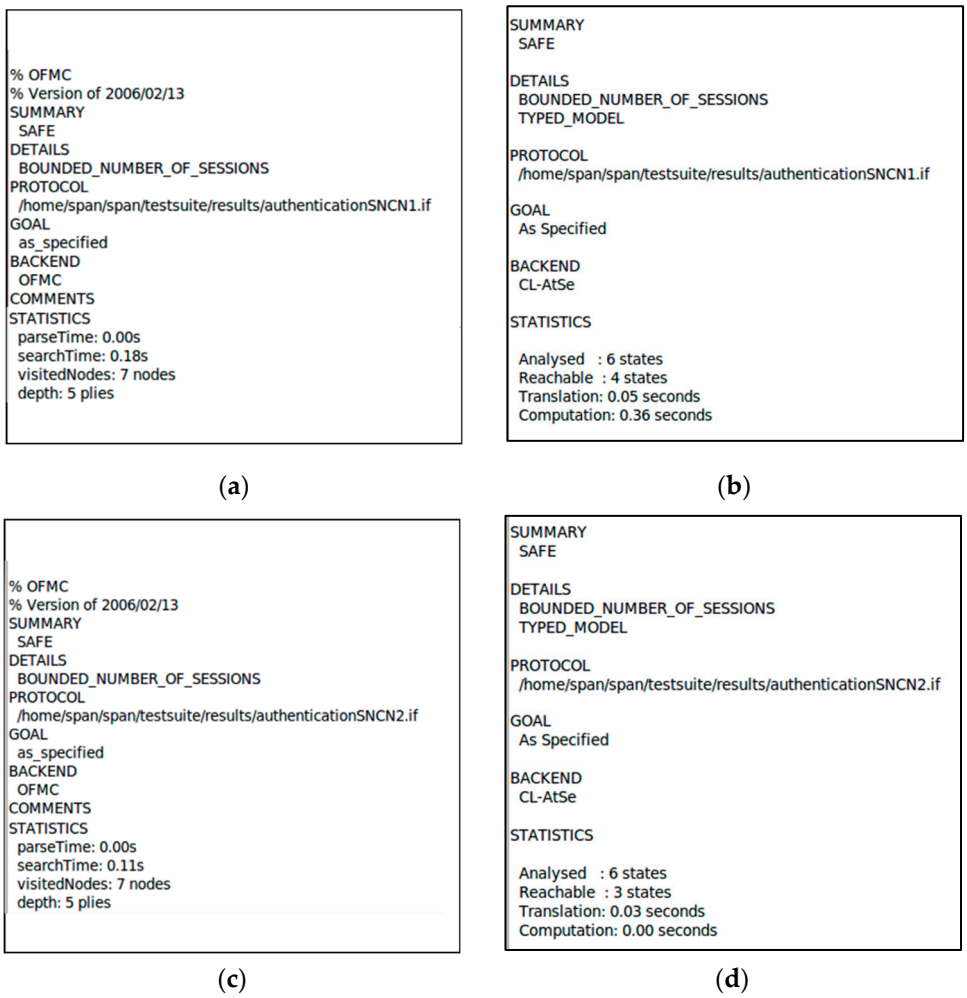

Figure 12. AVISPA with OFMC and CL-AtSe summary reports: (**a**) protocol-I OFMC report; (**b**) protocol-I CL-AtSe report; (**c**) protocol-II OFMC report; (**d**) protocol-II CL-AtSe report.

5. Performance Analysis

In this section, we compare the computation and communication costs of our proposed authentication protocols with those of related protocols [13,14,16–20].

5.1. Computation Costs

To calculate the computation costs, we considered the time complexity of each operation based on the experiments conducted in [2] and [31], where the time needed for scalar multiplication (T_{mul}) was 2.226 ms, the time needed for random number generation (T_{rng}) was 0.539 ms, the time needed for symmetric encryption and decryption (T_{ed}) was 0.0046 ms, the time needed for point addition (T_{add}) was 0.0288 ms, and the time needed for the one-way hash function (T_h) was 0.0023 ms.

The computation cost for Shen et al.'s scheme in [13] amounted to 6 T_{mul} + 2 T_{ed} + 2 T_{add} + 1, $T_h \approx$ 13.4251 ms. The computation cost of Liu et al. [14] was 4 T_{mul} + 2 T_{ed} + 2 $T_h \approx$ 8.9178 ms. The computation cost of Ur Rehman et al.'s scheme in [16] was 3 T_{rng} + 6 $T_h \approx$ 1.6308 ms. Chen et al.'s scheme in [17] had a cost of 3 T_{rng}+ 7 $T_h \approx$ 1.6331 ms. Wan et al.'s scheme in [18] required 2 T_{rng}+ 11 T_h + 6 $T_{mul} \approx$ 14.4593 ms. Moreover, the computation cost of Rehman et al.'s scheme in [19] was 1 T_{rng}+ 5 $T_h \approx$ 0.5505 ms. The computation cost for Li et al [20] was 2 T_{rng}+ 8 $T_h \approx$ 1.0964 ms. Our authentication protocol-I required 1 T_{mul} + 2 T_{rng} + 11, $T_h \approx$ 3.3293 ms. The cost of our authentication protocol-II was 1 T_{mul} + 1 T_{rng} + 8 $T_h \approx$ 2.7834 ms.

As shown in Table 3, the proposed authentication protocols had a better computation cost compared to the computation costs in [13,14,18] and had a higher computation cost than the computation costs in [16,17,19,20]. However, the existing schemes presented security concerns, where they did not satisfy all security requirements in this study. Thus, our proposed authentication protocols achieved better security than other schemes and had acceptable computation costs. Therefore, the proposed authentication protocols were suitable for the WBAN.

Table 3. Computation cost comparison.

Scheme	Computation Cost (ms)
[18]	14.4593
[13]	13.4251
[14]	8.9178
[17]	1.6331
[16]	1.6308
[20]	1.0964
[19]	0.5505
Proposed Protocol-I	3.3293
Proposed Protocol-II	2.7834

5.2. Communication Costs

To calculate the communication costs, we considered the bit sizes and communication overhead. Bit sizes were calculated based on the schemes in [31] and [32], where the one-way hash function had 160 bits, the group element G1 had 1024 bits, the identity had 128 bits, and the timestamp required 32 bits.

In Shen et al.'s [13] scheme, the first authentication message (Q_i) required 1024 bits, the second authentication message (MAC_{PDA}, Q_{PDA}) needed 1184 bits, and the third message (M) needed 1024 bits. Therefore, the total cost was 3232 bits. In Liu et al. [14], the first message (MAC_C) needed 160 bits, the second message (MAC_i^+) required 160 bits, and the third message (M) needed 1024 bits. Thus, the total cost was 1344 bits. In Ur Rehman et al. [16], the first message (tid_N, a_N, b_N, t_N) needed 192 bits and the second message (β, μ, η) required 160 bits. Thus, the total cost was 352 bits. In Chen et al. [17], the first message (tid_N,y_N,a_N,b_N,t_N) needed 352 bits and the second message (α, β, η, μ) required 480 bits. Thus, the total cost was 832 bits. In Wan et al. [18], the first message (D_1, G_{SN}, Z_{SN}, W_1, T_1) needed 1536 bits, the second message (M_1, W_2, T_2) needed 1216 bits, and the third message (W_3, T_3) needed 192 bits. Thus, the total cost was 2944 bits. In Rehman et al. [19], the first message (tid_N,a_N,b_N,t_N) needed 192 bits, whereas the second message (β, μ, η) required 160 bits. Therefore, the total cost was 352 bits. In Li et al. [20], the first message (tid_{SN}, H_{SN}, x) needed 448 bits and the second message (tid_{SN}, z, H_{CN}) required 448 bits. Thus, the total cost was 896 bits. In our proposed authentication protocol-I, the first message (V_{SN}, L_{SN1}, X_{SN}, T_1) needed 512 bits and the second message (U_i, L_{i1}, C, T_2) needed 352 bits. Therefore, the total cost was 864 bits. In our proposed authentication protocol-II, the message (R_i, L_{i2}, T_1) needed 352 bits.

The result of the communication cost comparison is shown in Table 4. In terms of communication costs in bits, authentication protocol-I had a better communication cost than the communication costs in [13,14,18,20] and had a higher communication cost than the communication costs in [16,17,19]. However, protocol-I achieved better security than these other schemes. Thus, protocol-I had an acceptable communication cost. On the other hand, authentication protocol-II had a better communication cost than the communication costs in [13,14,17,18,20] and had the same communication cost as [16,19]. Regarding communication overhead, the schemes in [13,14,18] needed three messages, which added overhead to the communication channel. On the other hand, the authentication protocol-II required one message. Therefore, the proposed authentication protocols are suitable for the WBAN.

Table 4. Communication cost comparison.

Scheme	Communication Cost (bits)	Communication Overhead
[13]	3232	3
[18]	2944	3
[14]	1344	3
[20]	896	2
[17]	832	2
[16]	352	2
[19]	352	2
Proposed Protocol-I	864	2
Proposed Protocol-II	352	1

6. Conclusions

We proposed two secure and efficient WBAN authentication protocols between sensor nodes and a controller node: authentication protocol-I for emergency medical reports and authentication protocol-II for periodic medical reports. The proposed scheme included an initialization phase, registration phase, authentication protocol-I, authentication protocol-II, and password change protocol. We conducted an informal security analysis and found that our proposed authentication protocols enhanced the security of the existing schemes and satisfied all security requirements in this study. We also implemented the BAN logic and found that our proposed authentication protocols attained mutual authentication. We also utilized the AVISPA simulation tool and found that our proposed protocols were secure against active and passive attacks. Moreover, we conducted a performance analysis and found that the proposed authentication protocols had suitable computation and communication costs for a WBAN.

Author Contributions: Conceptualization, A.M.A. and H.A.A.; methodology, A.M.A. and H.A.A.; validation, A.M.A. and H.A.A.; writing—original draft preparation, H.A.A.; writing—review and editing, A.M.A.; supervision, A.M.A.; funding acquisition, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to express their appreciation to the Journal Editor, an Associate Editor, and the four anonymous reviewers for their insightful comments. We also would like to thank Imam Abdulrahman Bin Faisal University for facilitating access to the resources used in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hsu, C.L.; Le, T.V.; Hsieh, M.C.; Tsai, K.Y.; Lu, C.F.; Lin, T.W. Three-Factor UCSSO Scheme with Fast Authentication and Privacy Protection for Telecare Medicine Information Systems. *IEEE Access* **2020**, *8*, 196553–196566. [CrossRef]
2. Son, S.; Lee, J.; Kim, M.; Yu, S.; Das, A.K.; Park, Y. Design of Secure Authentication Protocol for Cloud-Assisted Telecare Medical Information System Using Blockchain. *IEEE Access* **2020**, *8*, 192177–192191. [CrossRef]
3. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-Based Three-Factor Mutual Authentication Protocol for Telecare Medical Information System. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]
4. Poongodi, T.; Rathee, A.; Indrakumari, R.; Suresh, P. IoT Sensing Capabilities: Sensor Deployment and Node Discovery, Wearable Sensors, Wireless Body Area Network (WBAN), Data Acquisition. *Intell. Syst. Ref. Libr.* **2020**, *174*, 127–151. [CrossRef]
5. Taleb, H.; Nasser, A.; Andrieux, G.; Charara, N.; Motta Cruz, E. Wireless Technologies, Medical Applications and Future Challenges in WBAN: A Survey. *Wirel. Netw.* **2021**, *27*, 5271–5295. [CrossRef]
6. Deebak, B.D.; Al-Turjman, F. Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 346–360. [CrossRef]

7. Wazid, M.; Das, A.K.; Vasilakos, A.V. Authenticated Key Management Protocol for Cloud-Assisted Body Area Sensor Networks. *J. Netw. Comput. Appl.* **2018**, *123*, 112–126. [CrossRef]
8. Alzahrani, B.A.; Irshad, A.; Albeshri, A.; Alsubhi, K.; Shafiq, M. An Improved Lightweight Authentication Protocol for Wireless Body Area Networks. *IEEE Access* **2020**, *8*, 190855–190872. [CrossRef]
9. Zhang, J.; Zhang, Q.; Li, Z.; Lu, X.; Gan, Y. A Lightweight and Secure Anonymous User Authentication Protocol for Wireless Body Area Networks. *Secur. Commun. Netw.* **2021**, *2021*, 4939589. [CrossRef]
10. Yu, S.J.; Lee, J.Y.; Park, Y.H.; Park, Y.H.; Lee, S.W.; Chung, B.H. A Secure and Efficient Three-Factor Authentication Protocol in Global Mobility Networks. *Appl. Sci.* **2020**, *10*, 3565. [CrossRef]
11. Yang, X.; Yi, X.; Nepal, S.; Khalil, I.; Huang, X.; Shen, J. Efficient and Anonymous Authentication for Healthcare Service with Cloud Based WBANs. *IEEE Trans. Serv. Comput.* **2021**, 1. [CrossRef]
12. Ali, Z.; Ghani, A.; Khan, I.; Ashraf, S.; Hafizul, S.K. A Robust Authentication and Access Control Protocol for Securing Wireless Healthcare Sensor Networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502. [CrossRef]
13. Shen, J.; Chang, S.; Shen, J.; Liu, Q.; Sun, X. A Lightweight Multi-Layer Authentication Protocol for Wireless Body Area Networks. *Futur. Gener. Comput. Syst.* **2018**, *78*, 956–963. [CrossRef]
14. Liu, X.; Jin, C.; Li, F. An Improved Two-Layer Authentication Scheme for Wireless Body Area Networks. *J. Med. Syst.* **2018**, *42*, 1–14. [CrossRef]
15. Ding, Y.; Xu, H.; Zhao, M.; Liang, H.; Wang, Y. Group Authentication and Key Distribution for Sensors in Wireless Body Area Network. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 15501477211044338. [CrossRef]
16. Ur Rehman, Z.; Altaf, S.; Iqbal, S. An Efficient Lightweight Key Agreement and Authentication Scheme for WBAN. *IEEE Access* **2020**, *8*, 175385–175397. [CrossRef]
17. Chen, C.M.; Xiang, B.; Wu, T.Y.; Wang, K.H. An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks. *Appl. Sci.* **2018**, *8*, 1074. [CrossRef]
18. Wan, T.; Wang, L.; Liao, W.; Yue, S. A Lightweight Continuous Authentication Scheme for Medical Wireless Body Area Networks. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3473–3487. [CrossRef]
19. Rehman, Z.U.; Altaf, S.; Ahmad, S.; Huda, S.; Al-Shayea, A.M.; Iqbal, S. An Efficient, Hybrid Authentication Using Ecg and Lightweight Cryptographic Scheme for Wban. *IEEE Access* **2021**, *9*, 133809–133819. [CrossRef]
20. Li, X.; Ibrahim, M.H.; Kumari, S.; Kumar, R. Secure and Efficient Anonymous Authentication Scheme for Three-Tier Mobile Healthcare Systems with Wearable Sensors. *Telecommun. Syst.* **2018**, *67*, 323–348. [CrossRef]
21. Abiramy, N.V.; Sudha, S.V. A secure and lightweight authentication protocol for multiple layers in wireless body area network. *Smart Intell. Comput. Appl.* **2019**, *104*, 287–296. [CrossRef]
22. Koya, A.M.; Deepthi, P.P. Deepthi. Anonymous Hybrid Mutual Authentication and Key Agreement Scheme for Wireless Body Area Network. *Comput. Netw.* **2018**, *140*, 138–151. [CrossRef]
23. Arfaoui, A.; ben Letaifa, A.; Kribeche, A.; Senouci, S.M.; Hamdi, M. Adaptive Anonymous Authentication for Wearable Sensors in Wireless Body Area Networks. In Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 606–611. [CrossRef]
24. Morales-Sandoval, M.; De-La-Parra-Aguirre, R.; Galeana-Zapien, H.; Galaviz-Mosqueda, A. A Three-Tier Approach for Lightweight Data Security of Body Area Networks in E-Health Applications. *IEEE Access* **2021**, *9*, 146350–146365. [CrossRef]
25. Azees, M.; Vijayakumar, P.; Karuppiah, M.; Nayyar, A. An Efficient Anonymous Authentication and Confidentiality Preservation Schemes for Secure Communications in Wireless Body Area Networks. *Wirel. Netw.* **2021**, *27*, 2119–2130. [CrossRef]
26. Almuhaideb, A.M.; Alqudaihi, K.S. A Lightweight and Secure Anonymity Preserving Protocol for WBAN. *IEEE Access* **2020**, *8*, 178183–178194. [CrossRef]
27. Ostad-Sharif, A.; Nikooghadam, M.; Abbasinezhad-Mood, D. Design of a Lightweight and Anonymous Authenticated Key Agreement Protocol for Wireless Body Area Networks. *Int. J. Commun. Syst.* **2019**, *32*, e3974. [CrossRef]
28. Shuai, M.; Xiong, L.; Wang, C.; Yu, N. Lightweight and Privacy-Preserving Authentication Scheme with the Resilience of Desynchronisation Attacks for WBANs. *IET Inf. Secur.* **2020**, *14*, 380–390. [CrossRef]
29. Xu, Z.; Xu, C.; Liang, W.; Xu, J.; Chen, H. And Key Agreement Scheme for Medical Internet of Things. *IEEE Access* **2019**, *7*, 53922–53931. [CrossRef]
30. Almuhaideb, A.M.; Alqudaihi, K.S. Authentication in Wireless Body Area Network: Taxonomy and Open Challenges. *J. Internet Things* **2021**, *3*, 159–182. [CrossRef]
31. Kilinc, H.H.; Yanik, T. A Survey of SIP Authentication and Key Agreement Schemes. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1005–1023. [CrossRef]
32. Kim, M.; Yu, S.; Lee, J.; Park, Y.; Park, Y. Design of Secure Protocol for Cloud-Assisted Electronic Health Record System Using Blockchain. *Sensors* **2020**, *20*, 2913. [CrossRef] [PubMed]