An End-to-End Authentication Scheme for Healthcare IoT Systems Using WMSN

Shadi Nashwan*

Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka, 42421, Saudi Arabia
*Corresponding Author: Shadi Nashwan. Email: shadi_nashwan@ju.edu.sa
Received: 29 November 2020; Accepted: 25 January 2021

Abstract: The healthcare internet of things (IoT) system has dramatically reshaped this important industry sector. This system employs the latest technology of IoT and wireless medical sensor networks to support the reliable connection of patients and healthcare providers. The goal is the remote monitoring of a patient’s physiological data by physicians. Moreover, this system can reduce the number and expenses of healthcare centers, make up for the shortage of healthcare centers in remote areas, enable consultation with expert physicians around the world, and increase the health awareness of communities. The major challenges that affect the rapid deployment and widespread acceptance of such a system are the weaknesses in the authentication process, which should maintain the privacy of patients, and the integrity of remote medical instructions. Current research results indicate the need of a flexible authentication scheme. This study proposes a scheme with enhanced security for healthcare IoT systems, called an end-to-end authentication scheme for healthcare IoT systems, that is, an E2EA. The proposed scheme supports security services such as a strong and flexible authentication process, simultaneous anonymity of the patient and physician, and perfect forward secrecy services. A security analysis based on formal and informal methods demonstrates that the proposed scheme can resist numerous security-related attacks. A comparison with related authentication schemes shows that the proposed scheme is efficient in terms of communication, computation, and storage, and therefore cannot only offer attractive security services but can reasonably be applied to healthcare IoT systems.

Keywords: Healthcare IoT systems; wireless medical sensor networks; mutual authentication service; anonymity service; perfect forward secrecy service; COVID-19

1 Introduction

The main goal of internet of things (IoT) healthcare systems is the remote monitoring of the physiological data of patients by physicians to make their lives safer and more comfortable [1–4]. A patient’s physiological data can be collected remotely using specific sensors when the patient is
away from a healthcare center, such as electrical activity of the brain, blood pressure, heartbeat, body temperature, blood sugar, pedometer readings, and respiratory signs [5]. These sensitive data are transmitted to physicians, who can provide immediate and continuous health advice, especially during an emergency, such as during the coronavirus disease 2019 (COVID-19) pandemic [6]. Furthermore, healthcare IoT systems can enable the reduction of the number of healthcare centers and cover shortages in remote areas. Healthcare IoT systems can increase the health awareness of communities at a lower cost. In an IoT healthcare system, communications between service providers and patients can be achieved through the internet [7–10]. The communication technology used in most such systems is a wireless medical sensor network (WMSN) [11–13], which enables a reliable wireless connection between provider communication devices and sensor nodes of patients [14,15].

There are two security challenges to the deployment and acceptance of such technology: data privacy of patients and integrity of medical instructions. An unauthorized party could access the sensitive data collected by sensor nodes, with consequences such as loss of jobs or health insurance.

An unauthorized party could modify messages to deliver the wrong order or advice to patients, such as to update the dose pumped by an electronic insulin device [16,17]. Unauthorized access to messages transferred between system nodes is the primary source of weakness, and unauthorized access to sensor nodes may lead to inconsistent or fabricated medical reports. Restricted capabilities of the sensor nodes themselves can cause other types of weaknesses.

Many security attacks exploit these weaknesses, such as smartcard loss attacks; patient and physician anonymity attacks; sensor node spoofing; patient and physician impersonation; and replay, insider, desynchronization, and man-in-the-middle attacks [18–30]. Therefore, WMSN security requires significant improvement.

1.1 Security Requirements of Healthcare IoT Systems

To determine the security requirements of a healthcare IoT system, authentication must be accomplished through an integral unit. Requirements include the following [18–29].

- A lightweight cryptographic function should be used at the patient node to achieve fast and secure mutual authentication with WMSN nodes;
- Mutual authentication is required not only between WMSN nodes and physician nodes but among all communication nodes using different cryptographic techniques;
- Authentication should detect both random and malicious alterations of authentication messages without effecting the communication data rate;
- With anonymity service becoming increasingly important, authentication should hide identities of physician nodes and all communication nodes;
- Authentication should support perfect forward secrecy for long-term keys of communication nodes such that an unauthorized party cannot disclose previous authentication session keys.

1.2 Architecture of Healthcare IoT Systems

A healthcare IoT system must prevent unauthorized access to sensitive patient data and medical instructions, so a high priority in the design of the authentication scheme should be given for the malicious modifications. We propose a model architecture to monitor patient medical status using WMSN. Fig. 1 shows its main components.
Figure 1: The healthcare IoT system architecture using WMSN

WMSN nodes are either sensor or actuator nodes. Sensor nodes can sense the physiological data of patients and send regular data reports to smart devices such as electroencephalogram, heartbeat, pulse rate, pedometer, breathing, vision, glucose level, and temperature sensors [24,30]. Actuator nodes receive medical instructions from a physician though a patient’s smart device to carry out actions such as for insulin pumps, drug delivery, and brain and muscle stimulators [13,31].

A patient’s smart device node should be able to store and transmit physiological data captured by sensor nodes, including on-demand and emergency sensor data [20]. Sensor nodes periodically send captured data to the smart device, which forwards it directly to the gateway node (GWN) though the internet. Therefore, the smart device must be able to compute the lightweight cryptographic functions to communicate with sensor nodes and GWN node.

The GWN node is the heart of the authentication process, providing registration stages to providers and patient smart devices. It coordinates authentication and key agreement (AKA) execution between all authentication nodes. The physician gathers a patient’s physiological data indirectly from the GWN node to analyze it and monitor the patient’s physical condition.

The physician node is represented by a physician’s monitoring device that collects the physiological data of a patient, either indirectly from the GWN node during periodic monitoring, or directly from the patient’s smart device node during an emergency. The physician can diagnose the medical state of the patient and send medical instructions to actuator nodes for treatment.
WMSN nodes suffer from restrictions such as memory space and computational capability. Moreover, the connection takes place through low frequencies with WMSN nodes. Thus, the communication cost is proportional to the distance between the WMSN node and other nodes in the system.

The proposed architecture eliminates this obstacle. The physician node can connect to WMSN nodes through the patient’s smart device to collect physiological data in an emergency, allowing real-time gathering of data from the patient’s smart device [18].

1.3 Related Work

With increasing demand for healthcare IoT systems, numerous authentication schemes have been proposed to ensure data privacy and integrity of medical instructions. We highlight some schemes proposed for healthcare IoT systems using WMSN.

Kumar et al. [32] proposed an authentication scheme for WMSN to monitor the vital signs of a patient while preventing various security attacks. He et al. [33] claimed that this scheme was vulnerable to attacks such as password guessing, insider attacks, and cannot be achieved the user anonymity service, and suggested an improved scheme. This scheme cannot defeat attacks such as password guessing [34–36]. Li et al. [34] improved on this with an authentication scheme for WMSN applications to preserve user anonymity, using smartcard concepts and hash functions.

Das et al. [37] presented a secure and anonymous user authentication scheme based on smartcard concepts for healthcare applications using WMSN. Srinivas et al. [38] pointed out that the scheme of Li et al. [34] was vulnerable to stolen smartcard attack, insider attack, and user impersonation attack, and proposed a scheme claimed to resist all known attacks. Wu et al. [39] identified security weaknesses in the scheme of Srinivas et al. [38], such as offline password guessing attack.

Amin et al. [40] suggested an anonymity and robust mutual authentication scheme they claimed was more robust than other schemes. Ali et al. [41] showed that the scheme suffers from user offline password guessing, identity guessing, user impersonation, insider, and session key attacks.

Shuai et al. [42] noted that Wu et al. [39] and Ali et al. [41] cannot support perfect forward secrecy service nor resist a desynchronization attack, and proposed a scheme for healthcare systems using WMSN to solve these weaknesses. Fotouhi et al. [43] showed that Srinivas et al. [38] cannot support sensor anonymity and untraceability, nor resist an offline guessing attack, and claimed that Wu et al. [39] and Amin et al. [40] cannot support perfect forward secrecy or sensor anonymity and untraceability services.

It can be observed that none of the above schemes supports end-to-end authentication between all communication nodes of a system.

1.4 Motivations and Contributions

The healthcare IoT system is especially important in developing countries to achieve economic growth, as it can reduce the number and expense of healthcare centers, and enable patients to consult physicians from around the world. A healthcare IoT system can increase the health awareness of communities, especially during crises such as the COVID-19 pandemic. An authentication scheme by integrate the healthcare IoT system with WMSN technology can make it more secure and widely accepted.
The main contributions of this paper are as follows. An architecture of the healthcare IoT system using WMSN is introduced, including the main authentication nodes and the communication flow. An authentication scheme for healthcare IoT systems using WMSN is proposed. Security verification based on BAN logic is used to verify mutual authentication between nodes. An informal, comparative security analysis shows how the proposed scheme can resist all types of attacks. A comparative performance analysis demonstrates the scheme’s applicability.

1.5 Organization of This Paper

The remainder of this paper is organized as follows. Section 2 describes the proposed authentication scheme. Formal verification using BAN logic and an informal security analysis of the proposed scheme are described in Section 3. A performance analysis is presented in Section 4. We provide our conclusions in Section 5.

2 Proposed Authentication Scheme

An end-to-end authentication scheme for healthcare IoT systems using WMSN is proposed, which is based on the one-way hash function and symmetric cryptographic techniques.

2.1 Preliminaries

We address preliminaries such as scheme structure, notation, assumptions, and design requirements.

2.1.1 Scheme Structure

The proposed scheme has four types of authentication nodes; physician nodes (P_i), GWNs, WMSN nodes the physician must access (S_k), and smart device nodes (SD_j).

The scheme has 10 phases: physician node registration, smart device node registration, WMSN node registration, physician login authentication, patient login authentication, patient password change, physician password change, WMSN node authentication, and long- and short-term authentication.

2.1.2 Notation and Abbreviations

Notation and abbreviations are listed in Tab. 1.

2.1.3 Assumptions

We list the vulnerability assumptions used in the security analysis of the proposed authentication scheme.

- An adversary can recover the smartcard information of a physician node, and of the patient based on power consumption methods [44,45].
- An adversary can modify, intercept, capture, reroute, and retransmit authentication messages between all communication nodes where communication channels are considered unsecured and unreliable during authentication.
- An adversary can act as a legitimate smart device of a patient or physician node.
- The GWN node is considered a trusted communication node between the smart device of the physician node and the smart device node of the patient.
- Registration phases are accomplished directly through secure and reliable channels with the GWN node.
Table 1: Notation and abbreviations of proposed authentication scheme

| Notation | Description |
|----------|-------------|
| $P_i$    | Physician node |
| $PID_i$  | Identity number of $P_i$ |
| $PPW_i$  | Password of $P_i$ |
| $PSC_i$  | Security code of $P_i$ |
| $SN_i$   | Session number between $P_i$ and GWN node |
| $SC_i$   | Smartcard of $P_i$ |
| $ID_i$   | $P_i$ identity used in GWN side |
| $ID_{ip}$| Prefix identity for $P_i$ |
| $ID_{is}$| Suffix identity for $P_i$ |
| $X_i$    | Secret key of GWN node for $P_i$ |
| $SD_j$   | Patient smart device node |
| $SID_j$  | Identity number of $SD_j$ |
| $SPW_j$  | Password of $SD_j$ |
| $SC_j$   | Security code of $SD_j$ |
| $SN_j$   | Session number between $SD_j$ and GWN node |
| $SC_j$   | Smartcard of smart device $SD_j$ |
| $ID_j$   | $SD_j$ identity used in GWN side |
| $ID_{jp}$| Prefix identity for $SD_j$ |
| $ID_{js}$| Suffix identity for $SD_j$ |
| $X_j$    | Secret key of GWN node for $SD_j$ |
| $GWN$    | Gateway node/service provider |
| $S_k$    | WMSN node that physician node must access |
| $SID_k$  | Identity number of $S_k$ |
| $SS_{k0}$, $SS_{k1}$ | Sensor sequence number |
| $ST$     | Type of WMSN node |
| $PS_{ij}$| Subsequent authentication key |
| $R_0$, $R_5$, $R_9$ | Random numbers generated by $P_i$ side |
| $R_2$, $R_4$, $R_7$, $R_{10}$ | Random numbers generated by $SD_j$ side |
| $R_1$, $R_3$, $R_6$, $R_8$ | Random numbers generated by GWN node side |
| $h_0$, $h_1$, $h_2$, $h_3$ | Hash functions. |
| $TP$     | Timestamps of $P_i$ side |
| $T_{GWN0}$, $T_{GWN1}$ | Timestamps of GWN node side |
| $T_{SD}$ | Timestamp of $SD_j$ side |
| $\Delta T$ | Predefined threshold value |
| $||$     | String concatenation operation |
| $\oplus$ | XOR operation |
| $\Phi$   | Null value |

2.1.4 Design Requirements

We introduce the security requirements used to design the proposed authentication scheme.

- AKA concepts are utilized in all authentication phases. Therefore, communication nodes will mutually and securely authenticate each other to set up a reliable channel and exchange patient data after each authentication session between WMSN and physician nodes.
• Dynamic anonymity is used in authentication to hide the actual identities of patient’s smart device and physician nodes. Therefore, communication nodes use a different identity in each authentication session, and an adversary cannot track or masquerade patients or service provider workers.

• A robust integrity mechanism is used in all authentication phases to detect modifications in authentication messages exchanged between communication nodes. Hence, an adversary cannot alter these messages.

• Lightweight symmetric cryptography is used in long- and short-term authentication to encrypt and decrypt authentication parameters with high entropy. Thus, an adversary cannot guess these parameters in polynomial time. Consequently, physiological data exchanged between communication nodes remain confidential, and only physician nodes can receive it.

• One-way hash functions are used in long- and short-term authentication to derive the long-term session keys. Therefore, an adversary cannot disclose the current session keys nor disclose previous session keys.

2.2 Proposed Scheme Description

The proposed authentication scheme deploys a set of hash and symmetric cryptographic functions; its steps are described using the notation and abbreviations in Tab. 1.

2.2.1 Physician Node Registration Phase

A new physician wanting to access the physiological data collected by the WMSN nodes through the smart device of a patient, whether for periodic monitoring or an emergency, must first register in the GWN node using his/her monitoring device. Fig. 2 shows the physician node registration phase, whose steps are as follows.

![Figure 2: Physician node registration phase](image)

**Step 1:** A new physician node (P_i) selects identity number (PID_i), password (PPW_i), and security code (PSC_i) according to the system specifications. P_i generates a random number (R_0), and computes C_i = h_2(PID_i ∥ PPW_i ∥ R_0). P_i sends a registration request message {PID_i, C_i, and PSC_i} to the GWN node through a secure communication channel.

**Step 2:** In response to the P_i request, the GWN node verifies the existence of the identity (PID_i) in the physicians table, which contains the data of physicians that have already registered.
If it exists, then the GWN node rejects the registration request message \{M1\}, and asks \(P_i\) to select an unrepeated identity (PID\(_i\)). Otherwise, the GWN node generates a random number (R\(_i\)) and secret key (X\(_i\)), whose value is saved securely and separately.

The GWN node initiates the session number \(SN_i = h_0(R_i)\), and computes \(PK_i = h_1(PID_i \| X_i)\), \(PF_i = (PK_i \oplus PSC_i)\) and \(PV_i = h_1((SN_i \| PSC_i) \oplus (C_i \| PK_i))\). The GWN node initiates the pseudonym identities \(ID_{ip} = h_1(PID_i \| SN_i)\) and \(ID_{is} = \Phi\), where \(\Phi\) has the null value. The GWN node inserts the record of \(P_i\) in the physician node table [PID\(_i\), ID\(_{ip}\), ID\(_{is}\), and SN\(_i\)]. The GWN node embeds the authentication parameters [SN\(_i\), PF\(_i\), and PV\(_i\)] in a new smartcard (SC\(_i\)), and connects the new physician with his/her patients through a specific table. The GWN node initiates the session counter (C\(_{0ij}\) = 0), and returns SC\(_i\) and his/her list of patients [SID\(_j\), and C\(_{0ij}\)] to \(P_i\) via a secure communication channel.

**Step 3:** \(P_i\) receives SC\(_i\) and inserts R\(_0\). \(P_i\) separately and securely stores the list of patients.

2.2.2 *Smart Device Registration Phase*

A new patient’s smart device (SD\(_j\)) receives physiological data from connected WMSN nodes and forwards it to a service provider for periodic monitoring. This device must be registered in the GWN node. Fig. 3 shows the smart device registration phase, whose steps are as follows.

![Figure 3: Patient’s smart device registration phase](image)

**Step 1:** A new smart device (SD\(_j\)) selects an identity number (SID\(_j\)), password (SPW\(_j\)), and security code (SSC\(_j\)), whose values are formulated according to the system specifications. SD\(_j\) generates a random number (R\(_2\)) and computes \(C_j = h_2(SID_j \| SPW_j \| R_2)\). SD\(_j\) transmits the registration request message \{M1: SID\(_j\), C\(_j\), and SSC\(_j\)\} to the GWN node through a secure communication channel.

**Step 2:** In response to the SD\(_j\) request, the GWN node verifies the existence of identity SID\(_j\) in the table of registered patients. If it exists, the GWN node rejects the request and asks SD\(_j\) to select another identity. Otherwise, the GWN node generates a random number (R\(_3\)) and a secret key (X\(_j\)), whose value is saved securely. The GWN node initiates SN\(_j = h_0(R_3)\), and computes SN = (SSC\(_j\) \oplus SN\(_j\)), SK\(_j = h_1(SID_j \| X_j)\), SF\(_j = (SK_j \oplus SSC_j)\), and SV\(_j = h_1((SN_j \| SSC_j) \oplus (C_j \| SK_j))\). The GWN node initiates the pseudonym identity ID\(_j = h_1(SID_j \| SN_j)\), and ID\(_{jp} = ID_{js} = \Phi\), assigns a specific \(P_i\) to patient SID\(_j\), and securely updates the list of patients
for $P_i$. The GWN node adds the $SD_j$ record to the patient node table [$\text{SID}_j$, $\text{ID}_j$, $\text{ID}_{ij}$, $\text{ID}_{js}$, and $\text{SN}_j$], and embeds the authentication parameters [$\text{SN}$, $\text{SF}_j$, and $\text{SV}_j$] in a new smartcard ($SC_j$).

The GWN node returns $SC_j$ to $SD_j$ through a secure communication channel.

**Step 3:** $SD_j$ receives $SC_j$ and stores $R_2$ in $SC_j$. $SD_j$ initiates and securely stores the session counter ($C_{ij} = 0$).

### 2.2.3 WMSN Node Registration Phase

When a new WMSN node ($S_k$) is created as a sensor node to sense the physiological data of the patient or an actuator node to receive medical instructions from physician node $P_i$, the WMSN node must be registered in the patient’s smart device $SD_j$. This is a unique characteristic of the proposed authentication scheme. The stage can prevent the use of the sensor node by someone other than the patient. Fig. 4 shows WMSN node registration, which connects $S_k$ and $SD_j$. The steps are as follows.

**Figure 4:** WMSN node registration phase

**Step 1:** A new $S_k$ node sends a registration request message $M_1$: $\text{SID}_k$ to $SD_j$ through a secure communication channel, where the identity value ($\text{SID}_k$) of $S_k$ is initiated when created by the healthcare service provider.

**Step 2:** In response to the $S_k$ node request message $\{M_1\}$, $SD_j$ randomly generates the session number $\text{SN}_{k0} = (R_4)$ and initiates sensor sequence numbers $\text{SS}_{k0} = \text{SS}_{k1} = 0$. $SD_j$ adds the $S_k$ node record to the sensor nodes table [$\text{SID}_k$, $\text{SS}_{k0}$, and $\text{SN}_{k0}$]. $SD_j$ node securely sends $\{M_2: \text{SS}_{k1}, \text{SN}_{k0}\}$ to $S_k$.

**Step 3:** A new $S_k$ node securely stores [$\text{SS}_{k1}$, $\text{SN}_{k0}$].

### 2.2.4 Physician Login Authentication Phase

To monitor patients through WMSN services, the physician activates the monitoring device ($P_i$) by authentication to the smartcard ($SC_i$) obtained from the GWN node during physician node registration. Fig. 5 describes the physician login authentication phase between $P_i$ and $SC_i$. The main steps can be summarized as follows.

**Step 1:** $P_i$ inserts ($\text{PID}_i$, $\text{PPW}_i$, and $\text{PSC}_i$) as the login authentication request to the $SC_i$.

**Step 2:** In response to the $P_i$ request, $SC_i$ fetches ($R_0$) and computes $C_i = h_2(\text{PID}_i||\text{PPW}_i||R_0)$, $\text{PK}_i = (\text{PF}_i \oplus \text{PSC}_i)$, and $\text{XPV}_i = h_1((\text{SN}_i||\text{PSC}_i) \oplus (C_i||\text{PK}_i))$. $SC_i$ verifies whether ($\text{XPV}_i$) matches ($\text{PV}_i$) as stored in its memory by the GWN node. If not, then $SC_i$ rejects the login request.
and terminates the session. Otherwise, authentication will pass, and \(P_i\) is considered a legitimate node and will be used by an authorized physician. \(SC_i\) initiates the value of \(ID_i = h_1(PID_i \| SN_i)\).

![Figure 5: Physician login authentication phase](image)

### 2.2.5 Patient Login Authentication Phase

To use WMSN services, the patient activates his/her smart device (SD\(_j\)) to authenticate himself/herself to the smartcard (SC\(_j\)) obtained from the GWN node during smart device registration. Fig. 6 describes the patient login authentication phase between SD\(_j\) and SC\(_j\). The main steps are as follows.

**Step 1**: SD\(_j\) inserts (SID\(_j\)), (SPW\(_j\)), and (SSC\(_j\)) as the login authentication request to SC\(_j\).

**Step 2**: In response to the SD\(_j\) request, SC\(_j\) fetches (R\(_2\)) and computes 
\[
SN_j = (SSC_j \oplus SN), \\
C_j = h_2(SID_j \| SPW_j \| R_2), \\
SK_j = (SF_j \oplus SSC_j), \\
XSV_j = h_1((SN_j \| SSC_j) \oplus (C_j \| SK_j)).
\] SC\(_j\) verifies whether \((XSV_j)\) matches \((SV_j)\) as stored in its memory. If not, then SC\(_j\) terminates the login request and the session. Otherwise, authentication is passed, SD\(_j\) is considered a legitimate node, and it will be used by an authorized patient.

![Figure 6: Patient login authentication phase](image)

### 2.2.6 Smart Device Password Change Phase

This is accomplished between SD\(_j\) and SC\(_j\) when the patient wants to change a smart device (SD\(_j\)) password. Fig. 7 shows the smart device password change phase between SD\(_j\) and SC\(_j\) without going back to the GWN node. The patient must execute the following steps:

**Step 1**: The patient inserts (SID\(_j\)), (SPW\(_j\)), (SSC\(_j\)), and a new password \((\ast SPW_j)\) through SD\(_j\) as the request to change his/her password.

**Step 2**: SC\(_j\) computes 
\[
C_j = h_2(SID_j \| SPW_j \| R_2), \\
SK_j = (SF_j \oplus SSC_j), \\
XSV_j = h_1((SN_j \| SSC_j) \oplus (C_j \| SK_j)).
\] SC\(_j\) verifies whether \((XSV_j)\) matches \((SV_j)\) as stored in its memory.
by the GWN node. If not, then SC$_j$ rejects the request. Otherwise, SC$_j$ computes $C_j = h_2(SID_j∥SPW_j∥R_2)$ and a new verification code, $SV_j = h_1((SN_j∥SSC_j)⊕(C_j∥SK_j))$. SC$_j$ replaces the new code with the old one ($SV_j = SV_j^*$).

**Figure 7:** Smart device password change phase

2.2.7 Physician Password Change Phase

This is accomplished between P$_i$ and SC$_i$ when the physician (P$_i$) wants to change his/her password. Fig. 8 shows the details of the physician password change phase between P$_i$ and SC$_i$ without going back to the GWN node. The steps are as follows.

**Step 1:** The physician inputs (PID$_i$), (PPW$_i$), (PSC$_i$), and a new password ($^*$PPW$_i$) though P$_i$ to request a password change.

**Step 2:** SC$_i$ computes $C_i = h_2(PID_i∥PPW_i∥R_0)$, $PK_i = (PF_i⊕PSC_i)$, and $XPV_i = h_1((SN_i∥PSC_i)⊕(C_i∥PK_i))$. SC$_i$ verifies whether ($XPV_i$) matches ($PV_i$) as stored in memory by the GWN node. If not, SC$_i$ rejects the request. Otherwise, SC$_i$ computes $C_i = h_2(PID_i∥^*PPW_i∥R_0)$, and a new verification code $PV_i = h_1((SN_i∥PSC_i)⊕(C_i∥PK_i))$, and replaces the verification code with the new one ($PV_i = PV_i^*$).

**Figure 8:** Physician password change phase

2.2.8 Long-Term Authentication Phase

A physician can monitor a patient’s medical state by gathering physiological data indirectly from the patient’s smart device through the GWN node. Therefore, the physician, through the monitoring device, must achieve mutual authentication with the GWN node and the patient’s smart device SD$_j$, and to establish the subsequence session key with SD$_j$. Fig. 9 shows the
long-term authentication phase between the physician’s monitoring device $P_i$, the patient’s smart device $SD_j$, and the GWN node as a service provider. The following steps are carried out.

**Step 1:** $P_i$ initiates the authentication request message through $SC_i$ by inserting a patient identity ($SID_j$). $P_i$ generates a random number ($R_5$) and computes $TPK_i = (ID_i \oplus PK_i)$, where $ID_i$ was computed and $PK_i$ extracted during physician login authentication. $P_i$ computes $CT_{i0} = E_{TPK_i}(TP0 \| R_5 \| SID_j)$ and $V_{i0} = h_3(TP0 \| TPK_i \| SN_i \| ID_i \| R_5)$, where $TP0$ is a current timestamp of $P_i$. $P_i$ sends an authentication request message $\{M1: ID_i, CT_{i0}, V_{i0}\}$ to the GWN node through a public communication channel.

**Step 2:** Upon receiving $M1$ from $P_i$, the GWN node searches the table of physician nodes to find $(ID_{ip})$ and $(ID_{is})$ based on $ID_i$ as received from $P_i$. One of the following cases will occur [18,26]:

![Figure 9: Long-term authentication phase](image)

**Case 1:** $(ID_i \neq ID_{ip})$ and $(ID_i \neq ID_{is})$. The GWN node rejects $M1$ and terminates the session.

**Case 2:** $(ID_i = ID_{ip})$ and $(ID_{is} \neq \Phi)$. The GWN node computes new values for $SN_i = h_0(SN_i)$, $PK_i = h_1(PID_i \| X_i)$, and $TPK_i = (ID_i \oplus PK_i)$. The GWN node computes $< TP0 \| R_5 \| SID_j > = D_{TPK_i}(CT_{i0})$ and checks whether $P_i$ can monitor the medical state of $SID_j$. If not, then the GWN
node rejects M1 and terminates the session. Otherwise, the GWN node verifies the value of (TP0).
If it does not hold, then the GWN node rejects M1 and terminates the session. Otherwise, the
GWN node computes $X_{V_{\theta}} = h_3(TP_0 || TPK_i || SN_j || ID_j || R_5)$ to verify whether $(X_{V_{\theta}})$ matches $V_{\theta}$. If so, then the GWN node renews $ID_{is} = ID_{ip}$, and $ID_{ip} = h_1(ID || R_5)$. Otherwise, the GWN node rejects M1 and terminates the session.

**Case 3:** $(ID_i = ID_{ip})$ and $(ID_{is} = \Phi)$. The GWN node computes new values for $PK_i = h_1(PID_i || X_i)$ and $TPK_i = (ID_i \bigoplus PK_i)$, and computes $<TP_0 || R_5 || SID_j> = D_{TPK_i} (CT_{\theta})$. The GWN node checks whether $P_i$ can monitor the medical state of $SID_j$. If not, then the GWN node rejects M1 and terminates the session. Otherwise, the GWN node verifies the value of (TP0). If it does not hold, then the GWN node rejects M1 and terminates the session. Otherwise, the GWN node computes $X_{V_{\theta}} = h_3(TP_0 || TPK_i || SN_j || ID_j || R_5)$ to verify whether $X_{V_{\theta}}$ matches $V_{\theta}$. If so, then the GWN node renews $ID_{is} = ID_{ip}$ and $ID_{ip} = h_1(ID || R_5)$. Otherwise, the GWN node rejects M1 and terminates the session.

**Case 4:** $ID_i = ID_{is}$. The GWN node computes $PK_i = h_1(PID_i || X_i)$, $TPK_i = (ID_i \bigoplus PK_i)$, and $<TP_0 || R_5 || SID_j> = D_{TPK_i} (CT_{\theta})$, and checks whether $P_i$ can monitor the medical status of $SID_j$. If not, then the GWN node rejects M1 and terminates the session. Otherwise, the GWN node verifies the value of (TP0). If it does not hold, then the GWN node rejects M1 and terminates the session. Otherwise, the GWN node computes $X_{V_{\theta}} = h_3(TP_0 || TPK_i || SN_j || ID_j || R_5)$ to verify whether $X_{V_{\theta}}$ matches $V_{\theta}$. If so, then the GWN node renews $ID_{is} = ID_{ip}$. Otherwise, the GWN node computes $PK_i = h_1(PID_i || X_i)$ and $TPK_i = (ID_i \bigoplus PK_i)$, and checks whether $P_i$ can monitor the medical status of $SID_j$. If not, then the GWN node rejects M1 and terminates the session. Otherwise, the GWN node computes $X_{V_{\theta}} = h_3(TP_0 || TPK_i || SN_j || ID_j || R_5)$ to verify whether $X_{V_{\theta}}$ matches $V_{\theta}$. If so, then the GWN node renews $ID_{is} = ID_{ip}$ and $ID_{ip} = h_1(ID || R_5)$. Otherwise, the GWN node rejects M1 and terminates the session.

**Step 3:** According to the values of $PID_i$ and $SID_j$ determined through M1, the GWN node computes the authentication session key $PS_{ij} = h_2((PID_i \bigoplus X_i) || (SID_j \bigoplus X_j) || SQ_{ij})$, where $SQ_{ij}$ is a sequence number of the current execution for long-term authentication. The GWN node fetches the $SD_j$ node record from the patient table and computes $SK_j = h_1(SID_j || X_j)$ and $TSK_j = (ID_j \bigoplus SK_j)$. The GWN node initiates session counter $C_{0j} = (C_{0j} + 1)$ and computes pseudonym identity $ID_{ip} = h_1(SID_j || ID_j)$, $SN_j = h_0(SN_j)$, and $ID_j = h_1(SID_j || SN_j)$. The GWN node generates random number $R_6$ and computes $CT_{\theta j} = E_{TSK_j} (T_{GWN0} || R_6 || PS_{ij})$ and $V_{\theta j} = h_3(T_{GWN0} || R_6 || PS_{ij} || ID_j || SID_j || R_6)$, where $T_{GWN0}$ is the current timestamp. The GWN node sends an authentication request message $\{M2: C_{0j}, CT_{\theta j}, V_{\theta j}\}$ to $SD_j$ through an unsecure public communication channel.

**Step 4:** When $M2$ is received from the $GWN$ node, the $SD_j$ node through the $SC_j$ computes $\Delta C_j = (C_{0j} - C_{1j})$. $SD_j$ checks whether $1 \leq \Delta C_j \leq \mu_2$, where $\mu_2$ is assigned based on system requirements. If not, then $SD_j$ rejects $M2$ and terminates the session. Otherwise, it retrieves $SN_j = (SSC_j \bigoplus SN)$, computes $SN_j = h_0(SN_j)$ function for $\Delta C_j - 1$ times until $\Delta C_j - 1 = 1$. $SD_j$ updates $SN = (SSC_j \bigoplus SN_j)$, and computes $ID_j = h_1(SID_j || SN_j)$ and $TSK_j = (ID_j \bigoplus SK_j)$, where $SK_j$ was computed during patient login authentication. $SD_j$ computes $<T_{GWN0} || R_6 || PS_{ij}> = D_{TSK_j} (CT_{\theta j})$. $SD_j$ checks the value of $T_{GWN0}$. If it does not hold, then $SD_j$ rejects $M2$ and terminates the session. Otherwise, $SD_j$ sets $ID_{is} = ID_{ip}$ and computes $ID_{ip} = h_1(SID_j || ID_j)$ function for $(\Delta C_j - 1)$ times until $(\Delta C_j - 1) = 1$. $SD_j$ computes $X_{V_{\theta j}} = h_3(T_{GWN0} || PS_{ij} || ID_{ip} || SID || C_0)$ to verify whether $X_{V_{\theta j}}$ matches $V_{\theta j}$. If not, then $SD_j$ rejects $M2$ and terminates the session. Otherwise, $SD_j$ believes the GWN node is legitimate. $SD_j$ generates random number $R_7$, and computes $CT_{\theta j} = E_{TSK_j} (T_{SD} || R_7 || C_{1j})$ and $V_{\theta j} = h_3(T_{SD} || TSK_j || PS_{ij} || ID_{ip} || R_7)$, where $T_{SD}$ is the current timestamp of $SD_j$. Then $SD_j$ sets $C_{1j} = C_{0j}$, and sends the response authentication message $\{M3: ID_{ip}, CT_{\theta j}, V_{\theta j}\}$ to the GWN node through a public communication channel.
Step 5: Upon receiving M3 from SDj, the GWN node fetches TSKj again to compute \( \langle TSD || R7 || CT1 || C0j \rangle = D_{TSKj}(CT_{ij}) \), where the pseudonym identity \( ID_{ij} = ID_{ip} \). The GWN node verifies the value of TSD. If it does not satisfy, the GWN node rejects M3 and terminates the session. Otherwise, the GWN node computes \( XV_{ij} = h_3(TSD || TSKj || PS_{ij} || ID_{ij} || R7) \) to verify whether \( XV_{ij} \) matches \( V_{ij} \). If not, then the GWN node rejects M3 and terminates the session. Otherwise, the GWN node believes SDj is legitimate. The GWN node generates random number \( R8 \) and computes \( CT_{ij} = E_{TPKj}(R8 || PS_{ij} || TGWN1) \), where TGWN1 is the current timestamp. The GWN node computes \( V_{ij} = h_3(PID_j || PS_{ij} || R8 || SN_j || TGWN1) \), and sends the response authentication message \{M4: CT_{ij}, and V_{ij}\} to Pj.

Step 6: When M4 is received from the GWN node, Pj computes \( \langle R7 || PS_{ij} || TGWN1 \rangle = D_{TPKj}(CT_{ij}) \) and checks the value of TGWN1. If it does not hold, then Pj rejects M4 and terminates the session. Otherwise, Pj computes \( XV_{ij} = h_3(PID_j || PS_{ij} || R8 || SN_j || TGWN1) \) and verifies whether \( XV_{ij} \) matches \( V_{ij} \). If not, then Pj rejects M4 and terminates the session. Otherwise, Pj believes the GWN node is legitimate. Pj computes \( V_{ij} = h_3(PID_j || PS_{ij} || R8 || SN_j || (TP1 - TGWN1)) \) and \( V_{ix} = (TP1 \oplus TGWN1) \oplus V_{ij} \), where TP1 is the current timestamp of Pj. Then, Pj updates SNj = \( h_0(SN_j) \) and sets IDj = IDjp = \( h_1(ID_j || R5) \). Pj sends an acknowledgment message \{M5: IDj, and V_{ix}\} to the GWN node.

Step 7: Upon receiving M5 from Pj, the GWN node computes \( TP1 = (TP1 \oplus TGWN1) \oplus TGWN1 \) and \( \Delta TP = (TP1 - TGWN1) \), and checks whether \( \Delta TP \) exceeds the threshold \( \mu_3 \), which is assigned based on system requirements. If not, then the GWN node resends M4, with a fresh value of TGWN1, to Pj. Otherwise, the GWN node computes \( XV_{ij} = h_3(ID_{ip} || PS_{ij} || R7 || SN_j || \Delta TP) \) to verify whether \( XV_{ij} \) matches \( V_{ij} \). If not, then the GWN node rejects M5 and terminates the session. Otherwise, the GWN node believes Pj node is legitimate, and it updates SNj = \( h_0(SN_j) \), ID_{ix} = \( \Phi \), and SQ_{ij} = (SQ_{ij} + 1).

2.2.9 Short-Term Authentication Phase

When a physician wants to monitor a patient’s medical status based on real-time data through a direct communication channel, physiological data must be received from the patient’s smart device without returning to the GWN node. In this case, the physician achieves mutual authentication with the patient’s smart device to prevent unauthorized access to the direct unsecured connection.

Fig. 10 shows the short-term authentication phase between the Pj and SDj devices. The following steps are carried out after long-term authentication:

Step 1: Pj initiates an authentication request message through SCj by inserting a patient identity (SIDj). SCj retrieves the authentication session key (PS_{ij}) generated during the last long-term authentication phase with SDj through the GWN node. Pj generates random number \( R9 \) and initiates a session counter, \( C0_{ij} = (C0_j + 1) \). Pj computes pseudonym identity \( ID0_{ij} = h_1(SID_j \parallel ID0_{ij}) \), \( PS_{ij} = h_1(PS_{ij} \parallel ID0_{ij}) \), \( CT_{ij} = E_{PS_{ij}}(TP_j || R9 || C0_{ij}) \), and \( V_{ij} = h_3(TP_j \parallel SID_j \parallel PS_{ij} \parallel ID0_{ij} \parallel R9) \), where TPj is the current timestamp of Pj. Then, Pj sends authentication request message \{M1: C0_{ij}, CT_{ij}, V_{ij}\} to SDj.

Step 2: Upon receiving M1, SDj computes \( \Delta C_{ij} = (C0_{ij} - C1_{ij}) \) and checks whether \( 1 \leq \Delta C_{ij} \leq \mu_1 \), where \( \mu_1 \) is assigned based on system requirements. If not, then SDj rejects M1 and terminates the session. Otherwise, SDj sets ID1_{ij} = ID0_{ij}, computes \( ID1_{ij} = h_1(SID_j \parallel ID1_{ij}) \) function for \( (\Delta C_{ij} - 1) \) times until \( \Delta C_{ij} - 1 = 1 \). SDj calculates \( PS_{ij} = h_1(PS_{ij} \parallel ID0_{ij}) \) and \( < TP_j || R9 || C0_{ij} >= D_{PS_{ij}}(CT_{ij}) \). SDj verifies the value of TPj. If it does not satisfy, then SDj rejects M1 and
terminates the session. SD\textsubscript{j} computes \( XV_{ij} = h_3(T\textsubscript{P}\|SID\textsubscript{j}\|PS\textsubscript{j}\|ID1\textsubscript{ij}\|R\textsubscript{ij}) \) to verify whether \( XV_{ij} \) matches \( V_{ij} \). If not, then \( SD\textsubscript{j} \) rejects M1 and terminates the session. Otherwise, \( SD\textsubscript{j} \) believes \( P_i \) is legitimate. \( SD\textsubscript{j} \) generates random number \( R\textsubscript{10} \) and computes \( CT\textsubscript{j2} = E_{PS\textsubscript{ij}}(TP\|R\textsubscript{10}\|C1\textsubscript{ij}) \), where \( TP\textsubscript{j} \) is the current timestamp of \( SD\textsubscript{j} \). SD\textsubscript{j} computes \( V_{j2} = h_3(T\textsubscript{P}\|SID\textsubscript{j}\|PS\textsubscript{j}\|ID1\textsubscript{ij}\|R\textsubscript{10}) \), sets \( C1\textsubscript{ij} = C0\textsubscript{ij} \), and sends the response authentication message M2: \( ID1\textsubscript{ij} \), \( CT\textsubscript{j2} \), \( V_{j3} \) to \( P_i \).

**Step 3:** Upon receiving M2 from \( SD\textsubscript{j} \), \( P_i \) retrieves \( PS\textsubscript{ij} \), where the pseudonym identity \( ID1\textsubscript{ij} = ID0\textsubscript{ij} \). \( SD\textsubscript{j} \) computes \( <TP\|R\textsubscript{10}\|C1\textsubscript{ij} >= D_{PS\textsubscript{ij}}(CT\textsubscript{j2}) \), and \( P_i \) verifies \( TP\textsubscript{j} \). If it does not satisfy, then \( P_i \) rejects M2 and terminates the session. Otherwise, \( P_i \) computes \( XV_{j3} = h_3(T\textsubscript{P}\|SID\textsubscript{j}\|PS\textsubscript{j}\|ID1\textsubscript{ij}\|R\textsubscript{10}) \) to verify whether \( XV_{j3} \) matches \( V_{j3} \). If not, then \( P_i \) rejects M2 and terminates the session. Otherwise, \( P_i \) believes \( SD\textsubscript{j} \) is legitimate.

**Figure 10:** Short-term authentication phase

### 2.2.10 WMSN Node Authentication Phase

To exchange physiological data and medical instructions between smart device \( SD\textsubscript{j} \) and connected WMSN node \( S_k \), mutual authentication between both is achieved in all authentication sessions. Fig. 11 shows the WMSN node authentication phase between \( S_k \) and \( SD\textsubscript{j} \). The steps are as follows.

**Step 1:** To achieve mutual authentication with \( S_k \), \( SD\textsubscript{j} \) determines its identity (\( SID\textsubscript{k} \)) of \( S_k \). \( SD\textsubscript{j} \) randomly generates a secret key (\( SK\textsubscript{k} \)), updates \( SN\textsubscript{k0} = h_1(SN\textsubscript{k0}\|SID\textsubscript{k}) \), and computes \( CT\textsubscript{k} = ((SK\textsubscript{k}\|ST) \oplus h_2(SN\textsubscript{k0}\|SID\textsubscript{k}\|SS\textsubscript{k0}) \), where the value of \( ST \) is used to determine whether \( SD\textsubscript{j} \) needs to receive physiological data or forward medical instructions. \( SD\textsubscript{j} \) computes the pseudonym identity \( ID\textsubscript{k} = h_1(SK\textsubscript{k}\|SID\textsubscript{k}) \) and \( V_{k0} = h_3(ST\|SID\textsubscript{k}\|SK\textsubscript{k}\|SN\textsubscript{k0}\|SS\textsubscript{k0}) \), and renews \( SS\textsubscript{k0} = SS\textsubscript{k0} + 1 \). \( SD\textsubscript{j} \) sends an authentication request message \{M1: \( CT\textsubscript{k} \), \( V_{k0} \), \( SS\textsubscript{k0} \)\} to \( S_k \) through an unsecure communication channel.

**Step 2:** Upon receiving M1 from \( SD\textsubscript{j} \), \( S_k \) computes \( \Delta SS\textsubscript{k} = (SS\textsubscript{k0} - SS\textsubscript{k1}) \) and verifies whether \( 1 \leq \Delta SS\textsubscript{k} \leq \mu 0 \), where \( \mu 0 \) is assigned based on the system requirements. If not, then \( S_k \) rejects M1 and terminates the session. Otherwise, \( S_k \) sets \( SN\textsubscript{k1} = SN\textsubscript{k0} \), computes \( SN\textsubscript{k1} = h_1(SN\textsubscript{k1}\|SID\textsubscript{k}) \) function for \( \Delta SS\textsubscript{k} \) times until \( \Delta SS\textsubscript{k} = 1 \).

\( S_k \) determines \( (SK\textsubscript{k}\|ST) = CT\textsubscript{k} \oplus h_2(SN\textsubscript{k0}\|SID\textsubscript{k}\|SS\textsubscript{k0}) \) and computes \( V_{k1} = h_3(ST\|SID\textsubscript{k}\|SK\textsubscript{k}\|SN\textsubscript{k1}\|SS\textsubscript{k0} - 1) \). \( S_k \) verifies whether \( V_{k1} \) matches \( V_{k0} \). If not, then \( S_k \) rejects M1 and terminates the session. Otherwise, \( SD\textsubscript{j} \) is considered a legitimate smart device for \( S_k \). Then \( S_k \) computes
SN_{k0} = h_1(SN_{k1} \parallel SID_k), \ V_{k2} = h_3(ST \parallel SID_k \parallel SK_k \parallel SN_{k0} \parallel SS_{k0}), \text{ and } ID_k = h_1(SK_k \parallel SID_k); \text{ renews } SS_{k1} = SS_{k0}; \text{ and computes } SN_{k0} = h_1(SN_{k1} \parallel SID_k). S_k \text{ sends response authentication } \{M2: ID_k, \text{ and } V_{k2}\} \text{ to } SD_j \text{ through an unsecure communication channel.}

Step 3: When SD_j receives M2 from S_k, SD_j computes SN_{k0} = h_1(SN_{k0} \parallel SID_k) \text{ and } V_{k3} = h_3(ST \parallel SID_k \parallel SK_k \parallel SN_{k0} \parallel SS_{k0}), \text{ and verifies whether } V_{k3} \text{ matches } V_{k2}. \text{ If so, then } S_k \text{ is considered a legitimate WMSN node for } SD_j. \text{ Otherwise, } SD_j \text{ rejects } M2 \text{ and terminates the session.}

Figure 11: WMSN node authentication phase

3 Security Analysis

We discuss the security of the proposed authentication scheme. First, the BAN logic model is used to illustrate the validity of the mutual authentication service and secure session key [39]. Further analysis demonstrates that the scheme can resist all common attacks.

3.1 Formal Security Validation Using BAN Logic Model

The BAN logic model is used to validate the freshness, trustfulness and originality of the authentication messages exchanged between authentication nodes [41,42,46].

The login authentication and password change phases are not used frequently, and the registration phases are executed through secure communication channels. We concentrate on the soundness of the long-term, short-term, and WMSN node authentication phases. The basic notation and believing rules of the BAN logic model are summarized in Tabs. 2 and 3, respectively.

The lists the authentication phase goals, the idealized form of the authentication messages for the long-term, the short-term and WMSN node authentication phases, and the assumptions used in the verification process for the long-term, short-term, and WMSN node authentication phases are illustrated in Tabs. 4–6, respectively.

The physician node (P_i), GWN node (GWN), patient’s smart device (SD_j), and sensor node (S_k) are considered the main involved principles in the security verification of the proposed authentication scheme.

In the long-term authentication phase, TPK_i and TSK_j are the secret keys used to symmetrically encrypt authentication messages, while sets of unrepeated timestamps (TP_0, TP_1, TGWN_0, TGWN_1, and TS_D) and random numbers (R_5, R_6, R_7, and R_8) are used to guarantee the freshness
of an authentication session. In the short-term authentication phase, $PS_{ij}$ is a secret key used to symmetrically encrypt the authentication messages, while unrepeated timestamps $TP_i$ and $TP_j$ and random numbers $R_9$ and $R_{10}$ are used to guarantee the freshness of the authentication session. $SK_k$ is the secret key used to symmetrically encrypt the authentication messages in the WMSN node authentication phase, while serial numbers $SS_{k0}$ and $SS_{k1}$ are used to guarantee the freshness of authentication sessions.

Table 2: Notation of BAN logic model

| Notation | Description |
|----------|-------------|
| $X \equiv F$ | Principle F can consider X as a true statement or F is entitled to believe X. |
| $F \triangleleft X$ | Principle F sees X statement. So, F can receive, read, and repeat it. |
| $F \implies X$ | Once Principle F says the statement X. Then F sends a message including X. |
| $F \implies X$ | Principle F jurisdiction over X, So, F has authority on X statement. |
| $(X)$ | X is a fresh statement. |
| $(X, Y)$ | X statement or Y statement is a part of formula (X, Y). |
| $\langle X \rangle$ | X statement combined with Y statement. |
| $\{X\}_K$ | X statement is encrypted by key K. |
| $F \leftrightarrow K$ | Principles F and Q use the shared key K to communicate with each other. |
| $F \equiv Q$ | A secret X statement is known only for principles F and Q. |
| $SK$ | The session key used in the current session. |

Table 3: Rules of BAN logic model

| Rule | Formula |
|------|---------|
| Message meaning rule | $F \models F \leftrightarrow K \iff F \triangleleft \langle X \rangle_K \implies F \models Q \models \neg X$ |
| Freshness concatenation rule | $F \models \#(X)$ |
| Belief rule | $F \models X, F \models Y \implies F \models (X, Y)$ |
| Nonce verification rule | $F \models \#(X), F \models Q \models \neg X \implies F \models Q \models X$ |
| Jurisdiction rule | $F \models Q \implies X, F \models Q \models X \implies F \models X$ |
| Session key rule | $F \models \#(X), F \models Q \models X \implies F \models F \leftrightarrow Q$ |

The basic BAN logic rules, idealized form, and assumptions in Tabs. 2, 5, and 6 are used to validate the authentication phases.
Table 4: Authentication phase goals

| Phase                     | Goal 1 | Goal 2 | Goal 3 | Goal 4 | Goal 5 | Goal 6 | Goal 7 | Goal 8 | Goal 9 | Goal 10 | Goal 11 | Goal 12 | Goal 13 | Goal 14 | Goal 15 | Goal 16 |
|---------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|---------|
| Long-term authentication  | GWN | GWN | SDj | SDj | GWN | SDj | P | P | SDj | SDj | SDj | P | SDj | SDj | SDj | SDj |
| Short-term authentication | P | P | SDj | SDj | SDj | SDj | P | P | SDj | SDj | SDj | P | SDj | SDj | SDj | SDj |
| WMSN node authentication  | SDj | SDj | SDj | SDj | SDj | SDj | P | P | SDj | SDj | SDj | P | SDj | SDj | SDj | SDj |

Table 5: Idealized form of authentication phase messages

| Phase                     | Message | Idealized form |
|---------------------------|---------|----------------|
| Long-term authentication  | Mesg1   | ID_i, CT_i0, V_i0: (TP0, R5)TPKi |
|                           | Mesg2   | C0_j, CT_j0, V_j0: (TGWN0, R0)TSKj |
|                           | Mesg3   | ID_j, CT_j1, V_j1: (TS_D, R7)TSKj |
|                           | Mesg4   | CT_j1, V_j1: (Rj, TGWNI)TPKj |
| Short-term authentication | Mesg5   | C0_ij, CT_i2, V_i2: (TP_i, ID0_j, R_0)PSij |
|                           | Mesg6   | ID1_ij, CT_i2, V_i3: (TP_i, R10)PSij |
| WMSN node authentication  | Mesg7   | CT_k, V_k0, SS_k0: (SK_k, SS_k0)^1(SK_k0||SID_k) |
|                           | Mesg8   | ID_k, V_k2: (SK_k, SS_k0)^1(SK_k0||SID_k) |

3.1.1 Validation of Long-Term Authentication Phase

The validation process of the long-term authentication phase can be summarized as follows.

Using (Mesg1), a1 (GWN < ID_i, CT_i0, V_i0: (TP0, R5)TPKi) can be seen. From (a1), (Assmp9), the belief rule, and the message meaning rule, a2 (GWN ≡ P_i ≡ TP0,R5)TPKi can be obtained. Using (Assmp3) and the freshness concatenation rule, a3 (GWN ≡ #((TP0, R5)TPKi)) can be obtained. Using (a2), (a3), and the nonce verification rule, a4 (GWN ≡ P_i ≡
\((T_{P_0}, R_5)\)\(_{TPK_i}\) can be obtained. Therefore, from (a3), (a4), and the session key rule, \(a5\) (GWN \(\equiv\) GWN \(\leftrightarrow\) \(P_i\)) can be inferred, which represents (Goal1). Using (Assmp3), (a5), and the nonce verification rule, \(a6\) (GWN \(\equiv\) \(P_i\) \(\equiv\) GWN \(\leftrightarrow\) \(P_i\)) can be inferred, which represents (Goal2) as well.

| Phase                         | Assumption                      | Description |
|-------------------------------|---------------------------------|-------------|
| Long-term authentication      | Assmp1 \(P_i \equiv \#(R_8, T_{GWN1})\) |             |
|                               | Assmp2 \(P_i \equiv\) GWN \(\implies\) (R_8, T_{GWN1}) |             |
|                               | Assmp3 GWN \(\equiv\) \#(R_5, R_7, T_{P_0}, T_{SD}) |             |
|                               | Assmp4 GWN \(\equiv\) \(P_i\) \(\implies\) (R_5, T_{P_0}) |             |
|                               | Assmp5 GWN \(\equiv\) SD_j \(\implies\) (R_7, T_{SD}) |             |
|                               | Assmp6 SD_j \(\equiv\) \#(R_6, T_{GWN0}) |             |
|                               | Assmp7 SD_j \(\equiv\) GWN \(\implies\) (R_6, T_{GWN0}) |             |
|                               | Assmp8 \(P_i \equiv\) \(P_i\) \(\leftrightarrow\) GWN. |             |
|                               | Assmp9 GWN \(\equiv\) GWN \(\leftrightarrow\) \(P_i\) |             |
|                               | Assmp10 GWN \(\equiv\) GWN \(\leftrightarrow\) SD_j |             |
|                               | Assmp11 SD_j \(\equiv\) SD_j \(\leftrightarrow\) GWN |             |
|                               | Assmp12 \(P_i \equiv\) \#(R_{10}, TP_j) |             |
|                               | Assmp13 \(P_i \equiv\) SD_j \(\implies\) (R_{10}, TP_j) |             |
|                               | Assmp14 SD_j \(\equiv\) \#(TP_j, R_9) |             |
|                               | Assmp15 SD_j \(\equiv\) \(P_i\) \(\implies\) (R_9, TP_j) |             |
|                               | Assmp16 \(P_i \equiv\) \(P_i\) \(\leftrightarrow\) SD_j |             |
|                               | Assmp17 SD_j \(\equiv\) SD_j \(\leftrightarrow\) \(P_i\) |             |
|                               | Assmp18 SD_j \(\equiv\) \#(SK_k, SS_{K_0}) |             |
|                               | Assmp19 SD_j \(\equiv\) \(S_k\) \(\implies\) (SK_k, SS_{K_0}) |             |
| WMSN node authentication      | Assmp20 \(S_k \equiv\) \#(SK_k, SS_{K_0}) |             |
|                               | Assmp21 SD_j \(\equiv\) SD_j \(\leftrightarrow\) (SK_k, SS_{K_0}) |             |
|                               | Assmp22 \(S_k \equiv\) \#(SK_k, SS_{K_0}) |             |

Similarly, using (Mesg2), \(b1\) (SD_j \(\mathcal{C} C_{0_j}, CT_{0_j}, V_{0_j}: \langle(T_{GWN0}, R_6)\rangle_{TSK_j}\)) can be seen. Therefore, from (b1), (Assmp11), the belief rule, and the message meaning rule, \(b2\) (SD_j \(\equiv\) GWN \(\equiv\) \#(\langle(T_{GWN0}, R_6)\rangle_{TSK_j}) can be obtained. Next, using (Assmp6) and the freshness conjunctenation rule, \(b3\) (SD_j \(\equiv\) \#(\langle(T_{GWN0}, R_6)\rangle_{TSK_j}) can be obtained. Then, using (b2), (b3), and the nonce verification rule, \(b4\) (SD_j \(\equiv\) GWN \(\equiv\) \#(\langle(T_{GWN0}, R_6)\rangle_{TSK_j}) can be obtained. Therefore, from (b3), (b4), and the session key rule, \(b5\) (SD_j \(\equiv\) \#(SK_k, SS_{K_0}) \(\leftrightarrow\) GWN) can be inferred, which represents (Goal3). Using
(Assmp6), (b5), and the nonce verification rule, b6 (SD_j |=${} GWN |={} SD_j^{SK} \leftrightarrow GWN) can be inferred, which represents (Goal4) as well.

Similarly, using (Mesg3), then c1 (GWN |${} ID_j \bowtie CT_j, V_{ij}: (TSD_j,R_j)\rightarrow TSK_j) can be seen. So, from (c1), (Assmp10), the belief rule, and the message meaning rule, c2 (GWN |={} SD_j |={} (TSD_j,R_j)\rightarrow TSK_j) can be obtained. Next, using (Assmp3) and the freshness conjunction rule, c3 (GWN |=${} #((TSD_j,R_j)\rightarrow TSK_j)) can be obtained. Then, using (c2), (c3), and the nonce verification rule, c4 (GWN |={} SD_j |={} (TSD_j,R_j)\rightarrow TSK_j) can be obtained. Therefore, from (c3), (c4), and the session key rule, c5 (GWN |={} GWN |${} SD_j) can be inferred, which represents (Goal5).

Using (Assmp3), (c5), and the nonce verification rule, c6 (GWN |={} SD_j |={} GWN^{SK} |${} SD_j) can be inferred, which represents (Goal6) as well.

Finally, using (Mesg4), d1 (P_i \bowtie CT_i, V_{ij}: (R_8,TGW_{N1})\rightarrow TPK_j) can be seen. Thus, from (d1), (Assmp8), the belief rule, and the message meaning rule, d2 (P_i |={} GWN |={} (R_8,TGW_{N1})\rightarrow TPK_j) can be obtained. Next, using (Assmp1) and the freshness conjunction rule, d3 (P_i |=${} #((R_8,TGW_{N1})\rightarrow TPK_j)) can be obtained. Then, using (d2), (d3), and the nonce verification rule, d4 (P_i |={} GWN |={} (R_8,TGW_{N1})\rightarrow TPK_j) can be obtained. Therefore, from (d3), (d4), and the session key rule, d5 (P_i |={} P_i^{SK} \leftrightarrow GWN) can be inferred, which represents (Goal7). Also, using (Assmp1), (d5), and the nonce verification rule, d6 (P_i |={} GWN |={} P_i^{SK} \leftrightarrow GWN) can be inferred, which represents (Goal8).

The goals of the long-term authentication phase using the BAN logic model are proved. Therefore, mutual authentication can be achieved between the communication principles throughout this phase.

3.1.2 Validation of Short-Term Authentication Phase

The steps in the validation of the short-term authentication phase can be summarized as follows.

Using (Mesg5), e1 (SD_j \bowtie C0_{ij}, CT_{ij}, V_{ij}: ((TP_i,R_0))\rightarrow PS_{ij}) can be seen. So, from (e1), (Assmp17), the belief rule, and the message meaning rule, e2 (SD_j |={} P_i |={} ((TP_i,R_0))\rightarrow PS_{ij}) can be obtained. Next, using (Assmp14) and the freshness conjunction rule, e3 (SD_j |={} #((TP_i,R_0))\rightarrow PS_{ij}) can be obtained. Then, using (e2), (e3), and the nonce verification rule, e4 (SD_j |={} P_i |={} ((TP_i,R_0))\rightarrow PS_{ij}) can be obtained. Therefore, from (e3), (e4), and the session key rule, e5 (SD_j |={} SD_j^{SK} \leftrightarrow P_i) can be inferred, which represents (Goal9). Using (Assmp14), (e5), and the nonce verification rule, e6 (SD_j |={} P_i |={} SD_j^{SK} \leftrightarrow P_i) can be inferred, which represents (Goal10) as well.

Similarly, using (Mesg6), f1 (P_i \bowtie ID_{ij}, CT_{ij}, V_{ij}: ((TP_j,R_{10}))\rightarrow PS_{ij}) can be seen. Thus, from (f1), (Assmp16), the belief rule, and the message meaning rule, f2 (P_i |={} SD_j |={} ((TP_j,R_{10}))\rightarrow PS_{ij}) can be obtained. Next, using (Assmp13) and the freshness conjunction rule, f3 (P_i |={} #((TP_j,R_{10}))\rightarrow PS_{ij}) can be obtained. Then, using (f2), (f3), and the nonce verification rule, f4 (P_i |={} SD_j |={} ((TP_j,R_{10}))\rightarrow PS_{ij}) can be obtained. Therefore, from (f3), (f4), and the session key rule, f5 (P_i |={} P_i^{SK} \leftrightarrow SD_j) can be inferred, which represents (Goal11). Also, using (Assmp13), (f5), and the nonce verification rule, f6 (P_i |={} SD_j |={} P_i^{SK} \leftrightarrow SD_j) can be inferred, which represents (Goal12).
The goals of the short-term authentication phase using the BAN logic model are proved. Therefore, mutual authentication can be achieved between the communication principles throughout this phase.

3.1.3 Validation of WMSN Node Authentication Phase

The validation process of the WMSN node authentication phase can be summarized as follows.

Using ( Mesg7), g1 (Sk ⊆ CTk, Vk0, SSk0: (SSk0, SKk)h1(SNk0∥SIDk)) can be seen. So, from (g1), (Assmp22), the belief rule, and the message meaning rule, g2 (Sk = SDj ∼ ((SSk0, SKk)) h1(SNk0∥SIDk)) can be obtained. Next, using (Assmp19) and the freshness concatenation rule, e3 (Sk = #( ((SSk0, SKk))h1(SNk0∥SIDk)) can be obtained. Using (g2), (g3), and the nonce verification rule, e4 (Sk = SDj = ((SSk0, SKk))h1(SNk0∥SIDk)) can be obtained. Therefore, from (g3), (g4), and the session key rule, g5 (Sk = SKj SDj) can be inferred, which represents (Goal13).

Using (Assmp19), (g5), and the nonce verification rule, g6 (Sk = SDj = SKj SDj) can be inferred, which represents (Goal14).

Similarly, using (Mesg8), q1 (SDj ⊆ IDj, Vk2: (SSk0, SKk)h1(SNk0∥SIDk)) can be seen. So, from (g1), (Assmp21), the belief rule, and the message meaning rule, q2 (SDj = Sk ∼ ((SSk0, SKk)) h1(SNk0∥SIDk)) can be obtained. Next, using (Assmp17) and the freshness concatenation rule, q3 (SDj = #( ((SSk0, SKk))h1(SNk0∥SIDk)) can be obtained. Using (q2), (q3), and the nonce verification rule, e4 (SDj = Sk = ((SSk0, SKk))h1(SNk0∥SIDk)) can be obtained. Therefore, from (q3), (q4), and the session key rule, q5 (SDj = SDj SKj) can be inferred, which represents (Goal15). Also, using (Assmp17), (q5), and the nonce verification rule, q6 (SDj = Sk = SDj SKj) can be inferred, which represents (Goal16).

The goals of the WMSN node authentication phase using the BAN logic model are proved, and mutual authentication can be achieved between the communication principles throughout this phase.

3.2 Further Informal Security Analysis

When authentication is performed via unsecured public communication channels between authentication nodes, an adversary can capture, intercept, alternate, trace, impersonate, and retransmit authentication messages over these channels. We show how the proposed authentication scheme can prevent common attacks in such an environment. Comparisons with related authentication schemes are also presented.

3.2.1 Session and Key Agreement

To achieve session and key agreement, communication nodes should be able to securely create and agree on one or more session keys. After that, communication nodes can use different security techniques based on the session keys to establish secure communication. In the proposed authentication scheme, the (TPKj), (TSKj), and (PSij) keys are created in the long-term authentication phase, and the (SKk) key is created during WMSN node authentication.

Pi and the GWN node can create TPKj = (IDj ⊕ PKi) to achieve mutual authentication. (TPKj) is changed according to renewal of the value of (IDj) by performing IDj = h1(IDj∥R5) on both sides for each authentication session. But (PKi) cannot be extracted without inserting (PSCi).
on the $P_i$ side. $(PK_i)$ is computed on the GWN side as $PK_i = h_1(\text{PID}_i||X_i)$, where $(X_i)$ is known only to the GWN node.

Similarly, $(TSK_{ij})$ is established by $SD_j$ and the GWN node as $TSK_{ij} = (ID_j \oplus SK_j)$ to achieve mutual authentication. $(TSK_{ij})$ is changed according to the renewal of $(ID_j)$ as $ID_j = h_1(ID_j||SN_j)$ on both sides for each authentication session. But $(SK_j)$ cannot be extracted without inserting the security code $(SSC_j)$ on the $SD_j$ side. $(SK_i)$ is computed by the GWN node side as $SK_j = h_1(SID_j||X_j)$, where $(X_j)$ is known only to the GWN node.

The session key is generated by the GWN node as $PS_{ij} = h_2((\text{PID}_i \oplus X_j)||(SID_j \oplus X_j)||SQ_{ij})$, where the sequence number of the current authentication session $(SQ_{ij})$ is incremented when a new authentication session is executed between the authentication nodes. $(PS_{ij})$ is exchanged between $P_i$ and $SD_j$ as encrypted messages through the GWN node, where $P_i$ and $SD_j$ verify the extracted value of $(PS_{ij})$ using the verification codes $(V_{ij})$ and $(V_{ji})$, respectively.

The $(SK_k)$ key is created randomly by $SD_j$ to achieve mutual authentication with $S_k$. This key can be retrieved by $S_k$ as $SN_{k,ij} = h_1(SN_{k,ij}||SID_k)$, where $(SN_{k,ij})$ is changed according to the renewed value of $(\Delta SS_k)$ in each authentication session between them.

Therefore, session and key agreement service can be securely supported by the proposed authentication scheme, where the adversary can determine no session keys, either in the long-term phase or during WMSN node authentication phase. It should be noted that when long-term authentication is executed one time, short-term authentication may be executed $(C_{ij})$ times more than the $(TPK_i)$, and $(TSK_j)$ keys in the optimal case.

### 3.2.2 Mutual Authentication Service

Mutual authentication is considered an essential security service in most secure communication schemes, regardless of the system environment. Therefore, communication nodes should be able to authenticate each other to achieve trusted communication [34–43]. The proposed authentication scheme can support fully mutual authentication between all communication nodes through the long- and short-term authentication phases as well as through WMSN node authentication phase.

In the long-term authentication phase, the GWN node is considered the trusted node between $P_i$ and $SD_j$. Therefore, explicit mutual authentication can be achieved between communication nodes as follows. $P_i$ and the GWN node can prove each other’s authenticity by exchanging $M2$ and $M4$ based on symmetric encryption using the shared key ($TPK_i$).

**M1:** When the GWN node receives this message from $P_i$, it decrypts $(CT_{ij})$ to extract the authentication parameters $(TP_0)$, $(R_5)$, and $(SID_j)$, then computes the verification code function $XV_{\Delta 0} = h_3(TP_0||TPK_i||SN_i||ID_i||R_5)$, where the secret shared values $(SN_i)$ and $(ID_i)$ are changed in each authentication session. The GWN node checks the following conditions during this procedure: whether $P_i$ has permission to monitor the medical state of patient $SID_j$; if $(TP_0)$ is a fresh value; and if the received $(V_{\Delta 0})$ value matches $(XV_{\Delta 0})$. If these conditions are met, then the GWN node can ensure that this message has been transmitted from a legitimate $P_i$.

**M4:** When $P_i$ receives this message from the GWN node, $P_i$ decrypts $CT_{il}$ to extract the authentication parameters $(TGWN_{i1})$, $(R_8)$, and $(PS_{ij})$, and computes the verification code function $XV_{il} = h_3(PID_i||PS_{ij}||R_8||SN_i||TGWN_{i1})$, where the secret shared values $(SN_i)$ and $(PS_{ij})$ are changed in each authentication session. $P_i$ checks the following conditions during this procedure: whether $(TGWN_{i1})$ is a fresh value; and whether the received $V_{il}$ matches $XV_{il}$. If these conditions are met, then $P_i$ ensures that this message has been transmitted from a trusted GWN node.
Similarly, SD\(_j\) and the GWN node can prove each other’s authenticity by exchanging M2 and M3 based on symmetric encryption using the shared key TSK\(_j\), and the synchronized one-way hash function based on serial numbers C0\(_{ij}\) and C1\(_{ij}\).

**M2:** When SD\(_j\) receives this message from the GWN node, it computes \(\Delta C_j = (C0_j - C1_j)\) to compute the shared key (TSK\(_j\)); decrypts CT\(_{j0}\) to extract the authentication parameters (T\(_{GWN0}\), (R\(_6\)), and (PS\(_{ij}\)); and computes the pseudonym identity function \((\Delta C_j - 1)\) times as ID\(_{jp}\) = h\(_1\)(SID\(_j\)||ID\(_{jp}\)). SD\(_j\) computes the verification code function XV\(_{j0}\) = h\(_3\)(T\(_{GWN0}\)||PS\(_{ij}\)||ID\(_{j}\)||SID\(_j\)||C0\(_j\)), where the secret shared values (ID\(_{jp}\)) and (PS\(_{ij}\)) are changed in each authentication session. SD\(_j\) checks whether 1 ≤ \(\Delta C_j\) ≤ \(\mu_2\), T\(_{GWN0}\) is a fresh value, and the received V\(_{j0}\) matches XV\(_{j0}\). If these conditions are met, then SD\(_j\) can ensure that this message has been transmitted from a trusted GWN node.

**M3:** When the GWN node receives this message from SD\(_j\), it decrypts CT\(_{j1}\) to extract authentication parameters (T\(_{SD}\), (R\(_7\)), and (C1\(_j\)). It computes the verification code function XV\(_{j1}\) = h\(_3\)(T\(_{SD}\)||TSK\(_j\)||PS\(_{ij}\)||ID\(_1\)||R\(_7\)), where the secret shared values (TSK\(_j\)) and (ID\(_1\)) are changed in each authentication session. The GWN node checks whether T\(_{SD}\) is a fresh value, and the received V\(_{j1}\) matches XV\(_{j1}\). If these conditions are met, then the GWN node can ensure that this message has been transmitted from a legitimate SD\(_j\).

When mutual authentication is achieved between P\(_i\) and the GWN node and between the GWN node and SD\(_j\), the GWN node is considered a trusted node for both P\(_i\) and SD\(_j\). Then, mutual authentication has been achieved indirectly between P\(_i\) and SD\(_j\) through the GWN node after long-term authentication.

P\(_i\) and SD\(_j\) can authenticate each other during short-term authentication by exchanging M1 and M2. This phase is based on the symmetric encryption method using the shared key (PS\(_{ij}\)), and the synchronized one-way hash function method based on two serial numbers (C0\(_{ij}\)) and (C1\(_{ij}\)) as described in the following:

**M1:** When SD\(_j\) receives this message from P\(_i\), SD\(_j\) computes \(\Delta C_{ij} = (C0_{ij} - C1_{ij})\); decrypts CT\(_{i2}\) to extract the authentication parameters (TP\(_i\), (R\(_9\)), and (C0\(_{ij}\)); and computes the verification code function XV\(_{i3}\) = h\(_3\)(TP\(_i\)||SID\(_j\)||PS\(_{ij}\)||ID1\(_{ij}\)||R\(_9\)), where the secret shared value (ID1\(_{ij}\)) is changed in each authentication session. SD\(_j\) checks whether TP\(_i\) is a fresh value, 1 ≤ \(\Delta C_{ij}\) ≤ \(\mu_1\), and the received V\(_{ij}\) matches XV\(_{ij}\). If these conditions are met, then SD\(_j\) can ensure that this message has been transmitted from a legitimate P\(_i\).

**M2:** When SD\(_j\) receives this message from P\(_i\), SD\(_j\) decrypts CT\(_{i2}\) to extract the authentication parameters (TP\(_i\), (R\(_9\)), and (C0\(_{ij}\)); determines \(\Delta C_{ij} = (C0_{ij} - C1_{ij})\); computes ID1\(_{ij}\) = h\(_1\)(SID\(_j\)||ID1\(_{ij}\)) function for (\(\Delta C_{ij} - 1\)) times; and computes the verification code function XV\(_{i3}\) = h\(_3\)(TP\(_i\)||SID\(_j\)||PS\(_{ij}\)||ID1\(_{ij}\)||R\(_9\)), where the secret shared value (ID1\(_{ij}\)) is changed in each authentication session. SD\(_j\) checks whether TP\(_i\) is a fresh value, 1 ≤ \(\Delta C_{ij}\) ≤ \(\mu_1\), and the received V\(_{ij}\) matches XV\(_{ij}\). If these conditions are met, then SD\(_j\) can ensure that this message has been transmitted from a legitimate P\(_i\).

Therefore, mutual authentication can be achieved between P\(_i\) and SD\(_j\) through the exchange of M1 and M2 when short-term authentication is executed C\(_{ij}\) times.

S\(_k\) and SD\(_j\) can authenticate each other during WMSN node authentication by exchanging M1 and M2. This is based on the synchronized one-way hash function based on serial numbers SS\(_{k0}\) and SS\(_{k1}\), as follows.
M1: When $S_k$ receives this message from $SD_j$, $S_k$ finds $\Delta S_k = (SS_{k0} - SS_{k1})$, computes $SN_{k1} = h_1(SN_{k1}||SID_k)$ for $\Delta S_k$ times, and computes $(SK_k||ST) = CT_k \oplus h_2(SK_k||SID_k||SS_{k0})$ and verification code function $V_{k1} = h_3(ST||SID_k||SK_k||SN_{k1}||SS_{k0} - 1)$. $SD_j$ checks whether $1 \leq \Delta S_k \leq \mu 0$, and whether the received $V_{k0}$ matches $V_{k1}$. If these conditions are met, then $S_k$ can ensure that this message has been transmitted from a legitimate $SD_j$.

M2: When $SD_j$ receives this message from $S_k$, $SD_j$ computes $V_{k2} = h_3(ST||SID_k||SK_k||SN_{k0}||SS_{k0})$ and $SD_j$ node checks whether $V_{k2} = V_{k1}$ as received from $S_k$. If so, then $S_k$ is considered a legitimate WMSN node. Therefore, $P_i$ and $S_k$ can achieve mutual authentication through the exchange of M1 and M2.

3.2.3 Anonymity and Untraceability Service

To support user anonymity and untraceability, a user’s real identity should be protected to prevent an unauthorized node from realizing the user identity and from recognizing who communicates with whom [18,25,26,43].

The proposed authentication scheme hides the actual identities of the physician ($PID_i$), patient ($SID_j$), and WMSN node ($SID_k$) during authentication. During long- and short-term authentication, neither $P_i$ nor $SD_j$ uses its actual identity. Also, the actual identity of $S_k$ is not used during WMSN node authentication.

In long-term authentication, $P_i$ computes a pseudonym identity ($ID_i$) to achieve mutual authentication with the GWN node. $ID_i$ is initiated as $ID_i = h_1(PID_i||SN_i)$ during physician login authentication, where $PID_i$ is inserted by the physician. After that, $P_i$ and the GWN node synchronously renew $ID_i = h_1(ID_i||R5)$, where the random number $R5$ is generated in each authentication session.

Similarly, $SD_j$ computes a new pseudonym identity ($ID_j$) to achieve mutual authentication with the GWN node. $ID_j$ is initiated as $ID_j = h_1(SID_j||SN_j)$, where $SID_j$ is inserted by the patient. $SD_j$ and the GWN node synchronously renew $ID_j = h_1(SID_j||SN_j)$ based on a refresh session number that is renewed using the one-way hash function as $SN_j = h_0(SN_j)$ in each authentication session.

In short-term authentication, $P_i$ and $SD_j$ use new pseudonym identities for each session. On the $P_i$ side, a new identity for $SD_j$ is computed as $ID0_{ij} = h_1(SID_j||ID0_{ij})$. On the $SD_j$ side, its identity is computed as $ID1_{ij} = h_1(SID_j||ID1_{ij})$. It should be noted that to synchronize the values of ($ID1_{ij}$) and ($ID0_{ij}$), $SD_j$ executes the one-way hash function $(\Delta C_{ij} - 1)$ times, where $(\Delta C_{ij})$ is changed in each session.

In WMSN node authentication, a new pseudonym identity for $S_k$ is used in each session. $SD_j$ and $S_k$ can compute $ID_k = h_1(SK_k||SID_k)$, where $(SK_k)$ is changed in each session.

Therefore, the proposed authentication scheme can support full anonymity and untraceability service during all phases.

3.2.4 Perfect Forward Secrecy Service

To achieve forward secrecy, encryption and session keys are generated to ensure that past communication channels cannot be recovered even if the long-term secret keys are disclosed [18, 25,26,42,43].

To ensure that the proposed authentication scheme can support forward secrecy, we consider the following scenarios.
**Scenario 1:** Suppose the (TPK, i), (TSK, j), and (PS, ij) keys of the current authentication session have been disclosed during long-term authentication. The (TPK, i) and (TSK, j) keys are updated according to the fresh pseudonym identities for Pi and SDj computed as IDj = h1(IDj∥RNj) and IDj = h1(IDj∥SNj), respectively. PS, ij is updated by the GWN node as PS, ij = h2((PID, i ⊕ XI)||(SID, j ⊕ Xj)||SQ, ij) based on a fresh sequence number (SQ, ij). Since the session keys used in this phase are updated after each successful authentication session, the secrecy of previous and future communications will not be affected.

**Scenario 2:** Suppose an adversary discloses the (PS, ij) key of the current session during short-term authentication. The (PS, ij) key is updated in each authentication session according to the fresh pseudonym identity for SDj, which is computed as ID0, ij = h1(SID, j∥ID0, ij). As a result, the secrecy of previous and future communications will not be affected.

**Scenario 3:** Suppose the (SK, k) key of the current authentication session is disclosed to an adversary during WMSN node authentication. The (SK, k) key is generated randomly in each authentication session by SDj. Thus, the secrecy of previous and future communications will not be affected.

Based on the above, the proposed authentication scheme can support forward secrecy during all authentication phases.

### 3.2.5 Attacks Resistance Analysis

We illustrate how the proposed authentication scheme can prevent related and common attacks of such an environment according to previously mentioned vulnerability assumptions.

**Desynchronization Attack**

The most commonly used techniques to achieve user anonymity and perfect forward secrecy are the pseudonym identity, timestamp, encryption, and hashing techniques. Authentication schemes mostly renew the user identity and generate a new session key to be used in subsequent authentication sessions. The incorrect use of such techniques can lead to a desynchronization attack [18,26,42,43]. Therefore, synchronization between communication nodes in terms of identities and session keys is critical. The proposed authentication scheme can preserve synchronization between communication nodes in each authentication session. It should be noted that the desynchronization attack may be able to temporarily suspend the proposed authentication scheme but cannot impact resuming the authentication sessions in future.

**Replay Attack**

Authentication schemes usually deal with replay attacks using current timestamps, sequence or serial numbers, random numbers, and nonce values [18,26], which can generally prevent the reuse of authentication request messages gained by eavesdropping. Therefore, these methods can maintain the freshness of exchanged authentication messages between nodes. The proposed authentication scheme employs a set of timestamps, random numbers, and serial numbers as part of all challenge-and-response messages.

To ensure the proposed authentication scheme can resist the replay attack, consider the following attack scenarios.

**Scenario 1:** Suppose an adversary resends the authentication request message \{M1: IDi, CT, i0, V, i0\} to the GWN node, which was sent during long-term authentication. The GWN node will reject the authentication request and terminate the session because the value of (TP0) is out of range.
**Scenario 2:** Suppose an adversary resends the authentication request message \{M2: C0, CTj, Vj\} to SDj, which was sent during long-term authentication. SDj will reject the authentication request and terminate the session because the value of \(\Delta C_j\) may be out of the system requirement, and the value of \(\Delta (GW_{N0})\) out of the range.

**Scenario 3:** Suppose an adversary resends the short-term authentication request message \{M1: C0i, CTij, Vij\} to SDj, which was sent during short-term authentication. In response, SDj will reject the authentication request and terminate the session because the value of \(\Delta C_{ij}\) may be out of the system requirement, and the value of \(\Delta (TP_j)\) out of range.

**Scenario 4:** Suppose an adversary resends the request authentication message \{M1: CTk, Vk0, SSk0\} to Sk, which was sent during WMSN node authentication. In response, Sk will reject the authentication request and terminate the session because the value of \(\Delta SS_k\) may be out of the system requirement, and the value of \(\Delta (TP_k)\) out of range.

The values of timestamps and serial numbers are used in all authentication messages, and are updated after each successful authentication session. In the previous attack scenarios, the proposed authentication scheme could resist a replay attack during authentication.

**Smartcard Loss Attack**

It has been pointed out that an adversary can uncover the two authentication factors (identity and password) of the user from a stolen smartcard based on a power analysis attack or an offline procedure within polynomial time [18,26,44,45]. Therefore, this attack should be considered when designing an authentication scheme using smartcards.

The proposed authentication scheme is based on three authentication factors (identity, password, and secret security code). It should be noted that the secret security code may be computed by imprinting a biometric method (e.g., fingerprint, iris scan, or face recognition) using the smart devices of the physician and patient. The proposed authentication scheme employs a set of parameters and one-way hash functions to prevent such an attack.

It is useful to consider the following attack scenarios to ensure that the proposed authentication scheme can resist a smartcard loss attack using a fuzzy verifier [26].

**Scenario 1:** Suppose an adversary steals a physician’s smartcard (SCi) and finds the data [SNi, PFi, PVi], where SNi = h0(R1), PKi = h1(PIDi∥Xi), PFi = (PKi⊕PSCi), PVi = h1((SNi∥PSCi)⊕(Ci∥PKi)), and Ci = h2(PIDi∥PPWi∥R0). The adversary cannot retrieve and guess the correct values of (PIDi) and (PPWi), even of (PSCi), since there is an imperial address space of candidates for (PIDi), (PPWi), and (PSCi), which can be calculated by \(|\text{PID}_i| \times |\text{PPW}_i| \times |\text{PSC}_i|/1024\), where \(|\text{PID}_i|\), \(|\text{PPW}_i|\), and \(|\text{PSC}_i|\) are the address spaces of the physician’s identity, password, and security code, respectively.

**Scenario 2:** Suppose an adversary steals a patient’s smartcard (SCj) and finds the data [SNj, SFj, SVj], where SNj = h0(R3), SN = (SSCj⊕SNj), SKj = h1(SIDj∥Xj), SFj = (SKj⊕SSCj), SVj = h1((SNj∥SSCj)⊕(Cj∥SKj)), and Cj = h2(SIDj∥SPWj∥R2). Similar to the previous scenario, the adversary cannot retrieve and guess the correct value of (SIDj) or (SSCj), not even (SSCj), since there is an imperial address space of candidates for (SIDj), (SPWj), and (SSCj), which can be calculated by \(|\text{SID}_j| \times |\text{SPW}_j| \times |\text{SSC}_j|/1024\), where \(|\text{SID}_j|\), \(|\text{SPW}_j|\), and \(|\text{SSC}_j|\) are the address spaces of the patient’s identity, password, and security code, respectively.

The proposed authentication scheme can resist attacks on both the physician’s side and patient’s side.
Impersonation Attack

An adversary can generally intercept and forge authentication request messages transmitted through public channels to impersonate a communication node in the system. The adversary uses previously collected information to generate valid authentication parameters and initiate an illegal authentication request. Under the proposed authentication scheme, authentication request messages include infeasible authentication parameters that cannot be generated by the adversary. We consider the following attack scenarios to ensure the proposed scheme can resist an impersonation attack.

Scenario 1: Suppose an adversary intercepts the authentication request message \{M1: \text{ID}_i, \text{CT}_{i0}, \text{V}_{i0}\} that has been sent to the GWN node to impersonate \text{P}_i during long-term authentication. The encrypted value (\text{CT}_{i0}) is infeasible because the adversary does not know the secret keys (\text{TPK}_i), nor the current (\text{SN}_i) value. Thus, the adversary cannot compute (\text{V}_{i0}) using different (\text{T}_{i0}), (\text{SN}_i), and (\text{R}_5), and therefore cannot impersonate \text{P}_i.

Scenario 2: Suppose an adversary intercepts the authentication request message \{M2: \text{C}_0j, \text{CT}_{j0}, \text{V}_{j0}\} that has been sent to \text{SD}_j to impersonate the GWN node during long-term authentication. The encrypted value of (\text{CT}_{j0}) is infeasible because the adversary does not know the secret keys (\text{TSK}_j), nor the value of (\text{SID}_j). Thus, the adversary cannot compute (\text{V}_{j0}) using different (\text{PS}_{ij}), (\text{T}_{\text{GWN}0}), and (\text{R}_6), and therefore cannot impersonate the GWN node.

Scenario 3: Suppose an adversary intercepts the short-term authentication request message \{M1: \text{C}_0ij, \text{CT}_{i2}, \text{V}_{i3}\} that has been sent to \text{SD}_j to impersonate \text{P}_i during short-term authentication. The encrypted value of (\text{CT}_{i2}) is infeasible because the adversary does not know the secret keys (\text{PS}_{ij}), nor the value of (\text{SID}_j). Thus, the adversary cannot compute (\text{V}_{i3}) using different (\text{TP}_i), (\text{ID}_0ij), and (\text{R}_9). Therefore, the adversary cannot impersonate \text{P}_i.

Scenario 4: Suppose an adversary intercepts the request authentication message \{M1: \text{CT}_k, \text{V}_{ko}, \text{SS}_{ko}\} that has been sent to the \text{S}_k node to impersonate the \text{SD}_j node when the WMSN node authentication phase has been executed. However, the values of (\text{SN}_{ko}) and (\text{CT}_k) are infeasible because the adversary does not know (\text{SID}_k). Thus, the adversary cannot compute (\text{V}_{ko}) using different (\text{SK}_k) and (\text{SN}_{ko}), and therefore cannot impersonate \text{SD}_j.

The proposed authentication scheme can resist attacks when the adversary tries to impersonate the physician, GWN, and patient nodes.

Man-in-the-Middle Attack

Through the man-in-the-middle attack, an adversary can intercept and forge an authentication message transmitted through public channels to control the connection between communication nodes in the system. The adversary resends these authentication messages to make the nodes believe they are connected directly through forged authentication messages.

In the proposed authentication scheme, challenge and response messages exchanged between communication nodes are protected throughout all authentication phases. The long-term authentication phase uses (\text{TPK}_j) and (\text{TSK}_j) as secret keys to protect M1, M2, M3, M4, and M5, and (\Delta C_j) is used to guarantee synchronization between connection sides. The secret key (\text{PS}_{ij}) is used in short-term authentication to protect M1 and M2, and (\Delta C_{ij}) is used to guarantee synchronization between connection sides. The secret key (\text{SK}_k) is used in WMSN node authentication to protect M1 and M2, and (\Delta SS_k) is used to guarantee synchronization between connection sides. The proposed authentication scheme can resist the man-in-the-middle attack when the
adversary tries to intercept and forge authentication requests and response messages to control the connection between communication nodes.

**Wrong Login Attack**

Wrong login detection is considered fundamental to user login authentication. This not only can prevent a wrong login attack but can save needless computation and communication costs that can affect network congestion. When a smartcard receives the wrong login authentication data, the proposed authentication scheme provides a detection mechanism to prevent such an attack at the beginning of the physician or patient login authentication phases without unnecessary computation.

When SC_i receives the wrong login information, whether in (PID_i), (PPW_i), or (PSC_i) at the physician login authentication phase, SC_i fetches (R_0) and computes
\[ C_i = h_2(PID_i||PPW_i||R_0), \]
\[ PK_i = (PF_i \oplus PSC_i) \]
and verification code
\[ XPV_i = h_1((SN_i||PSC_i) \oplus (C_i||PK_i)). \]
SC_i verifies whether (XPV_i) matches (PV_i) as stored in its memory. If not, then SC_i rejects the login request and terminates the session.

Similarly, when SC_i receives the wrong login information, whether in (SID_j), (SPW_j), or (SSC_j), at the patient login authentication phase, SC_j fetches (R_2) and computes
\[ SN_j = (SSC_j \oplus SN), \]
\[ C_j = h_2(SID_j||SPW_j||R_2), \]
\[ SK_j = (SF_j \oplus SSC_j), \]
and
\[ XSV_j = h_1((SN_j||SSC_j) \oplus (C_j||SK_j)). \]
SC_j verifies whether (XSV_j) matches (SV_j) as stored in its memory. If not, then SC_j terminates the login request and terminates the session. The proposed authentication scheme can resist an unauthorized login attack without extra communication with the GWN node.

**Insider Attack**

In an insider attack, a gateway administrator or other privileged insider can use registration data to imitate a user through another system gateway. The proposed authentication scheme does not give the chance for privileged insiders to perform such attack, whether through execution of the physician or patient registration phases.

In the physician registration phase, the physician sends a registration request message \{PID_i, C_i, and PSC_i\} to the GWN node. Therefore, an adversary cannot get the physician’s password (PPW_i), whose value has been transmitted using the one-way hash function
\[ C_i = h_2(PID_i||PPW_i||R_0) \]
instead of the clear value. Similarly, a patient sends the registration request message \{SID_j, C_j, and SSC_j\} to the GWN node at the patient registration phase. An adversary cannot get the patient’s password (SPW_j), whose value has been transmitted using the one-way hash function
\[ C_j = h_2(SID_j||SPW_j||R_2) \]
instead of the clear value. Hence, the proposed authentication scheme can resist and avoid an insider attack.

**Stolen Password-verifier Table Attack**

An adversary can use a stolen password-verifier attack to steal a password from the password-verifier table stored in the network gateway to impersonate an authorized user and login to the system. Under the proposed authentication scheme, the GWN has no password-verifier table containing a physician’s password (PPW_i) or patient’s password (SPW_j). Hence, the scheme can resist such an attack.

### 3.2.6 Security Comparisons

We compare the proposed authentication scheme to other schemes [38–43] in terms of security services and resistance to attacks. The main security issues that distinguish the proposed authentication scheme from the other schemes can be summarized as follows.
Throughout the authentication phases of E2EA, the actual identities of the communication nodes are not used completely, all authentication messages are protected by both symmetric encryption and cryptographic hash functions, and all authentication messages include fresh and nonce values to synchronize the communication nodes. Patients can determine and control the connected sensor nodes with them, and can prevent their sensor nodes from being used by others.

As illustrated in Tab. 7, the other schemes [38–43] fail to provide anonymity and untraceability for patients and sensor nodes. Schemes [38–43] cannot support full mutual authentication. The other schemes fail to resist a patient’s smartcard loss attack, patient impersonation attack, sensor node impersonation attack, or wrong patient login attack. Scheme [38] cannot support the physician’s anonymity and untraceability. Schemes [38–41] fail to support perfect forward secrecy, and cannot resist a desynchronization attack. Scheme [40] fails to detect a physician impersonation attack, insider attack, or stolen password-verifier table attack. It should be noted that, compared to the other new authentication schemes [38–43], the proposed authentication scheme can fulfill more security features and can resist all related attacks.

Table 7: Security feature comparisons

| Security features                                      | [38] | [39] | [40] | [41] | [42] | [43] | E2EA |
|-------------------------------------------------------|------|------|------|------|------|------|------|
| Session and key agreement achieved                     | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Full mutual authentication achieved                    | No   | No   | No   | No   | No   | No   | No   |
| Physician’s anonymity and untraceability achieved     | No   | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Patient’s anonymity and untraceability achieved       | No   | No   | No   | No   | No   | No   | No   |
| Sensor node’s anonymity and untraceability achieved   | No   | No   | No   | No   | No   | No   | No   |
| Perfect forward secrecy achieved                       | No   | No   | No   | No   | Yes  | Yes  | Yes  |
| Three authentication factors (3F)                     | No   | No   | No   | Yes  | No   | No   | Yes  |
| Resistance to desynchronization attack                | No   | No   | No   | No   | Yes  | No   | Yes  |
| Resistance to replay attack                           | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Resistance to physician’s smart card loss attack      | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Resistance to patient’s smart card loss attack        | No   | No   | No   | No   | No   | No   | No   |
| Resistance to physician impersonate attack            | Yes  | Yes  | No   | Yes  | Yes  | Yes  | Yes  |
| Resistance to GWN node impersonate attack             | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Resistance to patient impersonate attack              | No   | No   | No   | No   | No   | No   | No   |
| Resistance to sensor node impersonate attack          | No   | No   | No   | No   | No   | No   | No   |
| Resistance to man-in-the-middle attack                | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Resistance to wrong physician login attack            | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  | Yes  |
| Resistance to wrong patient login attack              | No   | No   | No   | No   | No   | No   | No   |
| Resistance to insider attack                          | Yes  | Yes  | No   | Yes  | Yes  | Yes  | Yes  |
| Resistance to stolen password-verifier table attack   | Yes  | Yes  | No   | Yes  | Yes  | Yes  | Yes  |

4 Performance Analysis

We analyze the performance of the proposed authentication scheme and compare its cost to schemes [38–43] in terms of storage space, communication, and computation.

The storage space cost analysis is performed throughout the registration of physician, smart device, and WMSN nodes. Communication and computation cost analyses are performed for the
long-term, short-term, and WMSN node authentication phases. Other phases are not examined, as these are not executed frequently in any of the schemes.

In long-term authentication, a physician node sends an authentication request to the GWN node to obtain permission to monitor the physiological data of a specific patient, and delegates the GWN node to perform mutual authentication with the patient. Since to monitor the physiological data of the patient through the GWN node is expensive in terms of the size of data signaling and access time, the physician and patient obtain the session key to directly authenticate each other (n) times by short-term authentication without going back to the GWN node. The patient executes WMSN node authentication (n + 1) times with the connected sensor nodes, as shown in Fig. 12.

Figure 12: Timeline of authentication phases in proposed scheme

Using the same execution timeline, Figs. 12 and 13 show that while the proposed authentication scheme executes long-term authentication (m) times, the other schemes execute login authentication (m × n + m) times. According to an analytic model proposed to find the desired values of (n) [47], the best value satisfies 1 ≤ n ≤ 5. Therefore, in our analysis, we select m and n as 1 and 5, respectively. Thus, login authentication is executed six times in other authentication schemes [38–43] while in the proposed scheme, long-term authentication will be executed once, short-term authentication five times, and WMSN authentication six times. To perform valid comparisons, the sizes of all identities, passwords, security codes, random numbers, sequential
numbers, and timestamps are set to 128 bits. The input and output block sizes of symmetric encryption and decryption functions are multiples of 128 bits, and the output of the hash functions is 160 bits. According to experimental results \[18\], \[26\], the running time of SHA-1 and AES cryptographic functions are \(T_h \cong 0.00032 \text{ s}\), and \(T_{E/D} \cong 0.0056 \text{ s}\), respectively. So, we have \(T_h \cong 0.00032 \text{ s}\), and \(T_{E/D} \cong 0.0056 \text{ s}\).

4.1 Storage Space Cost Analysis

One of the main challenges in such a system is to optimize the storage space costs of sensor nodes and smartcards. To facilitate analysis, the size of embedded hash functions is not considered.

Tab. 8 shows the storage space costs of the smartcards and WMSN node in the proposed authentication scheme and schemes \[38–43\]. For the proposed scheme, the storage space costs for the physician’s smartcard \(\{\text{SN}_i, \text{PF}_i, \text{and PV}_i\}\), patient’s smartcard \(\{\text{SN}, \text{SF}_j, \text{and SV}_j\}\), and sensor node \(\{\text{SS}_{k1}, \text{SN}_{k0}\}\) require \((160 + 128 + 160) = 448 \text{ bits}\), \((160 + 128 + 160) = 448 \text{ bits}\), and \((128 + 128) = 256 \text{ bits}\), respectively.

| Scheme       | Smartcard/physician (bits) | Smartcard/patient (bits) | Sensor node (bits) |
|--------------|----------------------------|--------------------------|--------------------|
| [38]         | 608                        | N/A                      | 288                |
| [39]         | 480                        | N/A                      | 416                |
| [40]         | 736                        | N/A                      | 416                |
| [41]         | 800                        | N/A                      | 320                |
| [42]         | 864                        | N/A                      | 256                |
| [43]         | 736                        | N/A                      | 832                |
| E2EA         | 448                        | 448                      | 256                |

As illustrated in Tab. 8, the other authentication schemes \[38–43\] do not include the patient registration phase, as in the proposed scheme. Results indicate that the proposed scheme has the minimum required storage space whether in the smartcard of the physician or the sensor node.

4.2 Communication Cost Analysis

Communication costs are calculated based on the size of the total bits of the authentication messages that are exchanged between communication nodes during the authentication phases.

The communication costs of the proposed scheme can be summarized as follows. The authentication messages of the long-term authentication phase, \(\{\text{M1: ID}_i, \text{CT}_{i0}, \text{and V}_{i0}\}\), \(\{\text{M2: C0}_j, \text{CT}_{j0}, \text{and V}_{j0}\}\), \(\{\text{M3: ID}_j, \text{CT}_{j1}, \text{and V}_{j1}\}\), \(\{\text{M4: CT}_{i1}, \text{and V}_{i1}\}\), and \(\{\text{M5: ID}_i, \text{and V}_{i2}\}\), require \((128 + 384 + 160) = 672 \text{ bits}\), \((128 + 384 + 160) = 672 \text{ bits}\), \((128 + 384 + 160) = 672 \text{ bits}\), \((384 + 160) = 544 \text{ bits}\), and \((128 + 160) = 288 \text{ bits}\), respectively. The authentication messages of the short-term authentication phase, \(\{\text{M1: C0}_{ij}, \text{CT}_{j2}, \text{V}_{i3}\}\) and \(\{\text{M2: ID}_{ij}, \text{CT}_{j2}, \text{V}_{j3}\}\), require \((128 + 384 + 160) = 672 \text{ bits}\) and \((128 + 384 + 160) = 672 \text{ bits}\), respectively. Authentication messages of the WMSN authentication phase, \(\{\text{M1: CT}_k, \text{V}_{k0}, \text{SS}_{k0}\}\) and \(\{\text{M2: ID}_k, \text{and V}_{k2}\}\), require \((160 + 160 + 128) = 448 \text{ bits}\) and \((128 + 160) = 288 \text{ bits}\), respectively.
Tab. 9 shows the total communication costs for the proposed authentication scheme and schemes [38–43]. The results indicate that the proposed scheme has the minimum required communication costs.

| Scheme | Phase 1 | Phase 2 | Phase 3 | Total 4 (bits) |
|--------|---------|---------|---------|---------------|
|        | M1      | M2      | M3      | M4 | M5 | M1 | M2 | M1 | M2 |
| [38]   | 864     | 480     | 448     | 640 | N/A | N/A | N/A | N/A | N/A | 14502 |
| [39]   | 1692    | 448     | 800     | N/A | N/A | N/A | N/A | N/A | N/A | 17640 |
| [40]   | 896     | 608     | 320     | 800 | N/A | N/A | N/A | N/A | N/A | 15774 |
| [41]   | 1088    | 1248    | 578     | 800 | N/A | N/A | N/A | N/A | N/A | 22284 |
| [42]   | 704     | 800     | 448     | 320 | 288 | N/A | N/A | N/A | N/A | 15360 |
| [43]   | 768     | 960     | 608     | 800 | N/A | N/A | N/A | N/A | N/A | 18816 |
| E2EA   | 672     | 672     | 672     | 544 | 288 | 672 | 672 | 448 | 288 | 13984 |

1Long-term/login authentication phase, 2Short-term authentication phase, 3WMSN authentication phase, 4Total is calculated when n = 5, and m = 1.

4.3 Computation Cost Analysis

We compare the proposed scheme with schemes [38–43] in terms of computation costs. These are calculated based on the total execution time of the cryptographic functions in each authentication node. Tab. 10 shows the total cryptographic functions in each authentication node.

| Scheme | Physician node | GWN node | Patient node | Sensor node |
|--------|----------------|----------|--------------|-------------|
| [38]   | 12T_h          | 19T_h    | N/A          | 6T_h        |
| [39]   | 20T_{E/D} + 8T_h | 24T_{E/D} + 4T_h | N/A | 18T_{E/D} + 4T_h |
| [40]   | 11T_h          | 17T_h    | N/A          | 6T_h        |
| [41]   | 10T_{E/D} + 11T_h | 16T_{E/D} + 16T_h | N/A | 5T_{E/D} + 7T_h |
| [42]   | 1T_{E/D} + 15T_h | 12T_h    | N/A          | 17T_h       |
| [43]   | 10T_h          | 17T_h    | N/A          | 7T_h        |
| E2EA   | 6T_{E/D} + 5T_h | 12T_{E/D} + 13T_h | 6T_{E/D} + 5T_h | 6(7T_h) |
|        | + 5(6T_{E/D} + 5T_h) | + 5(6T_{E/D} + 4T_h) + 6(6T_h) |               |             |

1Total value is calculated when n = 5, and m = 1.

Tabs. 10 and 11 show the computation costs for the proposed authentication scheme as well as for schemes [38–43]. The results indicate that the proposed scheme has lower computation costs than authentication schemes [39,41], which use both cryptographic one-way hash functions and symmetric encryption functions. The proposed authentication scheme has higher computation costs than schemes [38,40,42,43], which use only the one-way hash functions.
Table 11: Computation cost analysis

| Scheme | Total crypto functions | Cost (s) |
|--------|------------------------|----------|
| [38]   | 222T_h                 | 0.07104  |
| [39]   | 312T_{E/D} + 96T_h     | 1.77792  |
| [40]   | 204T_h                 | 0.06528  |
| [41]   | 186T_{E/D} + 204T_h    | 1.10688  |
| [42]   | 6T_{f_c} + 264T_h      | 0.18708  |
| [43]   | 204T_h                 | 0.06528  |
| E2EA   | 84T_{E/D} + 146T_h     | 0.51712  |

1 Cost is calculated for n = 5, and m = 1

5 Conclusion

We proposed an end-to-end authentication scheme for healthcare IoT systems using WMSN (E2EA) to overcome current security weaknesses and make such systems more widely deployed and accepted. E2EA has appealing security features such as fully mutual authentication, full anonymity, and perfect forward service in all authentication phases. To design the E2EA authentication scheme, a usable architecture model for healthcare systems using WMSN was proposed. The BAN logic model was used to verify the mutual authentication between all nodes during all authentication phases. Throughout several attack scenarios, the security level of the E2EA authentication scheme was shown. Therefore, it cannot only support appealing security features but can resist common attacks such as desynchronization, impersonation, smartcard loss, replay, man-in-the-middle, insider, wrong login information, and password table. Moreover, compared to new state-of-the-art authentication schemes, E2EA authentication has the highest security level. A performance analysis illustrated that E2EA authentication incurs the minimum cost in terms of storage space and communication, and has a suitable level of computation costs compared to the other new authentication schemes. Finally, E2EA is applicable to healthcare IoT systems to remotely monitor a patient’s physiological data.

Acknowledgement: The author expresses his gratitude to all members of the Computer and Information Sciences College at Jouf University for their support.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declares no conflicts of interest to report regarding the present study.

References
[1] S. R. Patil, D. R. Gawade and S. N. Divekar, “Remote wireless patient monitoring system,” International Journal of Electronics & Communication Technology, vol. 6, no. 1, pp. 9–13, 2015.
[2] C. Assaba and S. Gite, “IOT based health care remote monitoring and context-aware appointment system,” International Journal of Current Engineering and Technology, vol. 7, no. 6, pp. 2347–5161, 2017.
[3] M. A. Uddin, A. Stranieri, I. Gondal and V. Balasubramanian, “Continuous patient monitoring with a patient centric agent: A block architecture,” IEEE Access, vol. 6, pp. 32700–32726, 2018.
[4] S. Zahoor-ul-Huq Nayemuddin, K. V. R. Reddy and P. P. Prasad, “IoT based real time health care monitoring system using labVIEW,” International Journal of Recent Technology and Engineering, vol. 8, no. 1S4, pp. 170–174, 2019.
[5] P. H. Waghmare and A. N. Bhute, “Healthcare monitoring system using smartphone,” *International Journal of Innovative Research in Science*, vol. 6, no. 6, pp. 12407–12413, 2017.

[6] M. M. Janet and R. Dharmalingam, “Enhanced IoT system in healthcare application using wireless body sensor networks,” *International Journal of Emerging Technology in Computer Science & Electronics*, vol. 24, no. 2, pp. 6–9, 2017.

[7] A. Julius and Z. Jian-Min, “IoT based patient health monitoring system using LabVIEW,” *International Journal of Science and Research*, vol. 6, no. 3, pp. 894–900, 2017.

[8] A. A. Ibrahim and W. Zhuopeng, “IoT patient health monitoring system,” *International Journal of Engineering Research and Application*, vol. 8, no. 1, pp. 77–80, 2018.

[9] F. Al-Turjman and S. Alturjman, “Context-sensitive access in industrial internet of things (IIoT) healthcare applications,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2736–2744, 2018.

[10] M. D. Babakerkhell and N. Pandey, “Analysis of different IoT based healthcare monitoring systems,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6S2, pp. 61–67, 2019.

[11] K. Premkumar, S. Padmapriya, R. Priyadharshani and K. A. Priyanka, “Survey on healthcare monitoring system using wireless sensor networks (WSN),” *International Journal of Pure and Applied Mathematics*, vol. 118, no. 14, pp. 485–492, 2018.

[12] G. Farzaneh and A. Rahnamaei, “Authentication in health care application using wireless medical sensor network: A survey,” *International Journal of Research in Computer Applications and Robotics*, vol. 4, no. 4, pp. 59–69, 2016.

[13] L. V. Morales, D. D. Ruiz and S. J. Rueda, “Comprehensive security for body area networks: A survey,” *International Journal of Network Security*, vol. 21, no. 2, pp. 342–354, 2019.

[14] A. A. Alli, A. I. Ikumola, O. A. Aliyu and O. A. Alli, “Development of a mobile remote health monitoring system–MRHMS,” *African Journal of Computing & ICT*, vol. 7, no. 4, pp. 15–22, 2014.

[15] A. Abdullah, A. Ismael, A. Rashid, A. Abou-ElNour and M. Tarique, “Real time wireless health monitoring application using mobile devices,” *International Journal of Computer Networks & Communications*, vol. 7, no. 3, pp. 13–30, 2015.

[16] K. Dhakal, A. Alsadoon, P. W. Prasad, R. S. Ali, L. Pham et al., “A novel solution for a wireless body sensor network: Telehealth elderly people monitoring,” *Egyptian Informatics Journal*, vol. 21, no. 2, pp. 91–103, 2020.

[17] A. Al-Qerem, F. Kharbat, S. Nashwan, S. Ashraf and K. Blaou, “General model for best feature extraction of EEG using discrete wavelet transform wavelet family and differential evolution,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 3, pp. 1–21, 2020.

[18] S. Nashwan, “AAA-WSN: Anonymous access authentication scheme for wireless sensor networks in big data environment,” *Egyptian Informatics Journal*, https://doi.org/10.1016/j.eij.2020.02.005.

[19] M. Al-Fayoumi and S. Nashwan, “Performance analysis of SAP-NFC protocol,” *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 125–130, 2018.

[20] S. Nashwan, “SE-H: Secure and efficient hash protocol for RFID system,” *International Journal of Communication Networks and Information Security*, vol. 9, no. 3, pp. 358–366, 2017.

[21] T. Thaier, B. J. Mohd, M. Imran, G. Almashaqbeh and A. V. Vasilakos, “Secure authentication for remote patient monitoring with wireless medical sensor networks,” *Sensors*, vol. 16, no. 24, pp. 1–5, 2016.

[22] Y. Lu, L. Li, Peng h. and Y. Yang, “An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks,” *Sensors*, vol. 16, no. 24, pp. 1–21, 2016.
[26] L. Xiong, T. Peng, H. Liang and Z. Liu, “A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks,” Sensors, vol. 17, no. 24, pp. 1–28, 2017.

[27] J. Jung, J. Kim, Y. Choi and D. Won, “An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks,” Sensors, vol. 16, no. 24, pp. 1–30, 2016.

[28] M. Wazid, A. K. Das and A. V. Vasilakos, “Authenticated key management protocol for cloud-assisted body area sensor networks,” Journal of Network and Computer Applications, vol. 123, no. 2, pp. 112–126, 2018.

[29] Y. Chen, Y. Ge, Y. Wang and Z. Zeng, “An improved three-factor user authentication and key agreement scheme for wireless medical sensor networks,” IEEE Access, vol. 7, pp. 85440–85451, 2019.

[30] S. Patil and S. Pardeshi, “Health monitoring system using IoT,” International Research Journal of Engineering and Technology, vol. 5, no. 4, pp. 1678–1682, 2018.

[31] M. K. Hasan, M. Shahjalal, M. Z. Chowdhury and Y. M. Jang, “Real-time healthcare data transmission for remote patient monitoring in patch-based hybrid OCC/BLE networks,” Sensors, vol. 19, no. 24, pp. 1–23, 2019.

[32] P. Kumar, S. Lee and J. Lee, “E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks,” Sensors, vol. 12, no. 2, pp. 1625–1647, 2012.

[33] D. He, K. Kumar, J. Chen, C. Lee, N. Chilamkurti et al., “Robust anonymous authentication protocol for healthcare applications using wireless medical sensor networks,” Multimedia Systems, vol. 21, no. 1, pp. 49–60, 2015.

[34] X. Li, J. Niu, S. Kumari, W. Liang, “A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity,” Security and Communication Networks, vol. 9, no. 15, pp. 2643–2655, 2016.

[35] F. Wu, L. Xu, S. Kumari and X. Li, “An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks,” Multimedia Systems, vol. 23, no. 2, pp. 195–205, 2017.

[36] O. Mir, J. Munilla and S. Kumari, “Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks,” Peer-to-Peer Networking and Applications, vol. 10, no. 1, pp. 79–91, 2017.

[37] A. K. Das, A. K. Sutrala, V. Odelu and A. Goswami, “A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks,” Wireless Personal Communications, vol. 94, no. 3, pp. 1899–1933, 2017.

[38] J. Srinivas, D. Mishra and S. Mukhopadhyay, “A mutual authentication framework for wireless medical sensor networks,” Journal of Medical Systems, vol. 41, no. 5, pp. 80–99, 2017.

[39] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari et al., “Lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks,” Future Generation Computer Systems, vol. 82, no. 1, pp. 727–737, 2017.

[40] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan and N. Kumar, “A robust and anonymous patient monitoring system using wireless medical sensor networks,” Future Generation Computer Systems, vol. 80, no. 4, pp. 483–495, 2018.

[41] R. Ali, A. K. Pal, S. Kumari, A. K. Sangaiah, X. Li et al., “An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring,” Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 1, pp. 74, 2018.

[42] M. Shuai, B. Liu, N. Yu and X. Xiong, “Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks,” Security and Communication Networks, vol. 2019, no. 12, pp. 1–14, 2019.

[43] M. Fotouhi, M. Bayat, A. K. Das, H. A. Far, S. M. Pournaghi et al., “A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT,” Computer Networks, vol. 177, no. 1, pp. 107333, 2020.
[44] T. Messerges, E. Dabbish and R. Sloan, “Examining smart-card security under the threat of power analysis attacks,” IEEE Transactions on Computers, vol. 51, no. 5, pp. 541–552, 2002.

[45] P. Kocher, J. Jaffe and B. Jun, “Differential power analysis,” in Proc. 19th Annual Int. Cryptology Conf. on Advances in Cryptology (CRYPTO ’99), Berlin, Germany, pp. 15–19, 1999.

[46] S. Nashwan and B. Alshammari, “Formal analysis of MCAP protocol against replay attack,” British Journal of Mathematics & Computer Science, vol. 22, no. 1, pp. 1–14, 2017.

[47] Y. Lin and Y. Chen, “Reducing authentication signaling traffic in third-generation mobile network,” IEEE Transactions on Wireless Communications, vol. 2, no. 3, pp. 493–501, 2003.