Hindawi
Journal of Healthcare Engineering
Volume 2023, Article ID 4845850, 13 pages
https://doi.org/10.1155/2023/4845850

*Research Article*

# Provably Secure and Lightweight Patient Monitoring Protocol for Wireless Body Area Network in IoHT

**Qi Xie, Dongnan Liu, Zixuan Ding, Xiao Tan, and Lidong Han**

*Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China*

Correspondence should be addressed to Qi Xie; qixie68@126.com

Received 21 April 2022; Revised 4 June 2022; Accepted 24 November 2022; Published 13 February 2023

Academic Editor: Muhammad Asghar Khan



As one of the important applications of Internet of Health Things (IoHT) technology in the field of healthcare, wireless body area network (WBAN) has been widely used in medical therapy, and it can not only monitor and record physiological information but also transmit the data collected by sensor devices to the server in time. However, due to the unreliability and vulnerability of wireless network communication, as well as the limited storage and computing resources of sensor nodes in WBAN, a lot of authentication protocols for WBAN have been devised. In 2021, Alzahrani et al. designed an anonymous medical monitoring protocol, which uses lightweight cryptographic primitives for WBAN. However, we find that their protocol is defenseless to off-line identity guessing attacks, known-key attacks, and stolen-verifier attacks and has no perfect forward secrecy. Therefore, a patient monitoring protocol for WBAN in IoHT is proposed. We use security proof under the random oracle model (ROM) and automatic verification tool ProVerif to demonstrate that our protocol is secure. According to comparisons with related protocols, our protocol can achieve both high computational efficiency and security.

## 1. Introduction

Wireless body area network (WBAN) exists as a transmission network for body monitoring. It has intellectual network appliances, such as personal wireless terminals, wearable devices, and wireless sensors. Individuals can use network devices to build personalized health networks based on WBAN, and they are substantial participants in the Internet of Health Things (IoHT) application. WBAN is widely used in patient monitoring, physiological parameter measurement, and so on. The measured data are transmitted by the sensor to the devices with a forwarding function in real time using wireless network transmission and then stored in the database of the remote server [1–3]. Using WBAN-based systems, patient-specific electronic medical records can be established, and professionals can analyze medical data through patient electronic records. Moreover, the electronic data of patients can be used for later analysis and diagnosis, and medical personnel can provide targeted medical services based on these data [4].

The communication and interaction of WBAN are based on an open wireless channel, so it is inevitable to face a series of challenges. Attackers can eavesdrop, tamper, intercept publicly transmitted information, and use the obtained information to launch attacks and obtain patients' privacy. This poses a great threat to the medical IoHT and patient privacy [5, 6]. In addition, the WBAN system requires real-time data transmission and timely processing of a large number of communication requests, which makes the energy consumption of infrastructures with limited efficiency very heavy [7]. However, most devices for WBAN have limited computing power, so they cannot perform traditional cryptographic calculations. Moreover, intensive computation will bring about overblown network loads, which will affect the performance of the system. Therefore, the medical field urgently needs a lightweight privacy-protected secure key agreement to meet the above challenges.

In recent years, a lot of anonymous medical key agreements have been proposed. An innovative dynamic ID-based key agreement in telecare medical information system (TMIS) was presented by Chen et al. [8]. However, Xie et al. [9] state that Chen et al.'s scheme cannot defend against off-line password guessing attacks and impersonation attacks and has no privacy protection and perfect forward secrecy. Xie et al. [10] presented a novel authentication protocol for TMIS in 2014, which is considered to be pragmatic and secure. Radhakrishnan and Muniyandi [11] submitted a two-factor key agreement for TMIS based on elliptic curve cryptography (ECC). In 2015, Wang and Zhang [12] solved the anonymity of authentication in WBAN using bilinear pairs, and their scheme could defend against known-key attacks and man-in-middle attacks. However, according to the research of Jiang et al. [13], the protocol cannot resist client forgery attacks, is not suitable for practical applications, and may lead to nonsynchronization of system logs. In 2017, Li et al. [14] proposed an anonymous authentication scheme. It employs lightweight cryptographic primitives (e.g., hash function operations) and asserts that it has realized the mutual authentication of the sensor nodes worn by patients and the hub node and has realized unlinkability and anonymity. Later, Koya et al. [15] stated that it is not feasible because their scheme assumes that the central node is entirely credible. Moreover, it is defenseless to sensor impersonation attacks. Soni and Singh [16] submitted a lightweight authentication scheme employing low-cost operations for WBAN. Based on the wireless medical sensor network, Jan et al. [17] submitted a patient key agreement for the healthcare system to realize secure and efficient communication between users and sensors. Recently, Ullah et al. [18] submitted a hyperelliptic curve and pragmatic IoT-based crossdomain authentication scheme for WBAN. In addition, Ullah et al. [19–21] proposed a multimessage signcryption protocol, anonymous certificateless signcryption protocol, and certificate-founded signcryption protocol for IoHT. Khan et al. [22] proposed an online-offline certificate-less signature protocol for IoHT.

Wu et al. [23] designed an identity authentication scheme using unilateral bilinear pairing technology which only performs bilinear pairing at the access point (AP). After that, Chen and Peng [24] declared that it cannot realize mutual authentication and is also susceptible to client forgery attacks. Li et al. [25] devised a key agreement founded on ECC to realize user anonymity. But Sowjanya et al. [26] found that their scheme not only has the problems of clock nonsynchronization and excessive control power of users but also no perfect forward secrecy. Kalra and Sood [27] submitted a secure key agreement that is not affected by time synchronization, which is based on the password. In 2021, Chunka et al. [28] reviewed their scheme and found that it had many security issues. For instance, due to the defects in the gateway design, the scheme cannot confirm the authenticities of sensor nodes, so it cannot resist the sensor nodes captured attacks, and the gateway private key is prone to be leaked. In addition, a large number of redundant multiple hash calculations increase the computational burden on the system. Xu et al. [29] raised an anonymous and lightweight patient monitoring protocol using lightweight cryptographic primitives. The survey of Alzahrani et al. [30] shows that off-line identity guessing attacks will wreck its anonymity, and it is also defenseless to key compromise attacks and replay attacks.

*1.1. Motivation and Contributions.* According to the summary of the existing literature [30–33], we found that some protocols using lightweight cryptographic primitives cannot resist various attacks, and many protocols based on asymmetric cryptography have high time complexity. In 2021, Alzahrani et al. [30] designed an anonymous medical monitoring scheme. Nevertheless, their scheme is defenseless to stolen-verifier attacks, known-key attacks, and off-line identity guessing attacks and has no perfect forward secrecy. To realize a secure and lightweight authentication protocol in WBAN systems, we propose a patient monitoring protocol. Here, our contributions are as follows:

(i) We reviewed Alzahrani et al.'s [30] protocol and analyzed its drawbacks, for example, known-key attacks, stolen-verifier attacks, and off-line identity guessing attacks

(ii) A patient monitoring protocol is proposed to realize the security and lightweight requirements of WBAN systems

(iii) Using the automated verification tool ProVerif and formal security proof in ROM, we demonstrate the proposed protocol is secure

(iv) Our protocol is relatively pragmatic and secure by performance comparison

The remaining section is constructed as follows: the system model and preliminaries are given in Section 2. In Section 3, we describe the review and drawbacks of Alzahrani et al.'s protocol. Section 4 proposes a patient monitoring scheme. Its security is analyzed in Sections 5 and 6. Its security properties, computation cost, storage cost, and communication cost between ours and some related protocols are evaluated in Section 7. Section 8 concludes the paper.

## 2. System Model and Preliminaries

In this section, we present the system model and attack model. Concurrently, we describe the physically unclonable function (PUF).

*2.1. System Model.* Figure 1 illustrates its system model. It adopts the centralized two-hop architecture of WBAN, which includes the following devices: sensor nodes (SNs), relay nodes (RNs), and medical server node (MS). RN is the intermediate node, and only needs to forward messages between SN and MS, and it can add or delete its identity before forwarding messages. RN is always within the communication coverage of MS, and SN is covered by at least one RN. Resource-constrained SN monitors and collects patients' medical health data by being worn or embedded into patients.

*2.2. Attack Model.* Presuming the attacker (AR) maintains the following capacities:

(1) AR can capture messages transmitted via open channels and may eavesdrop, replace, replay, or intercept the data in these messages

(2) AR can obtain verifier table stored in MS, but cannot obtain its secret key

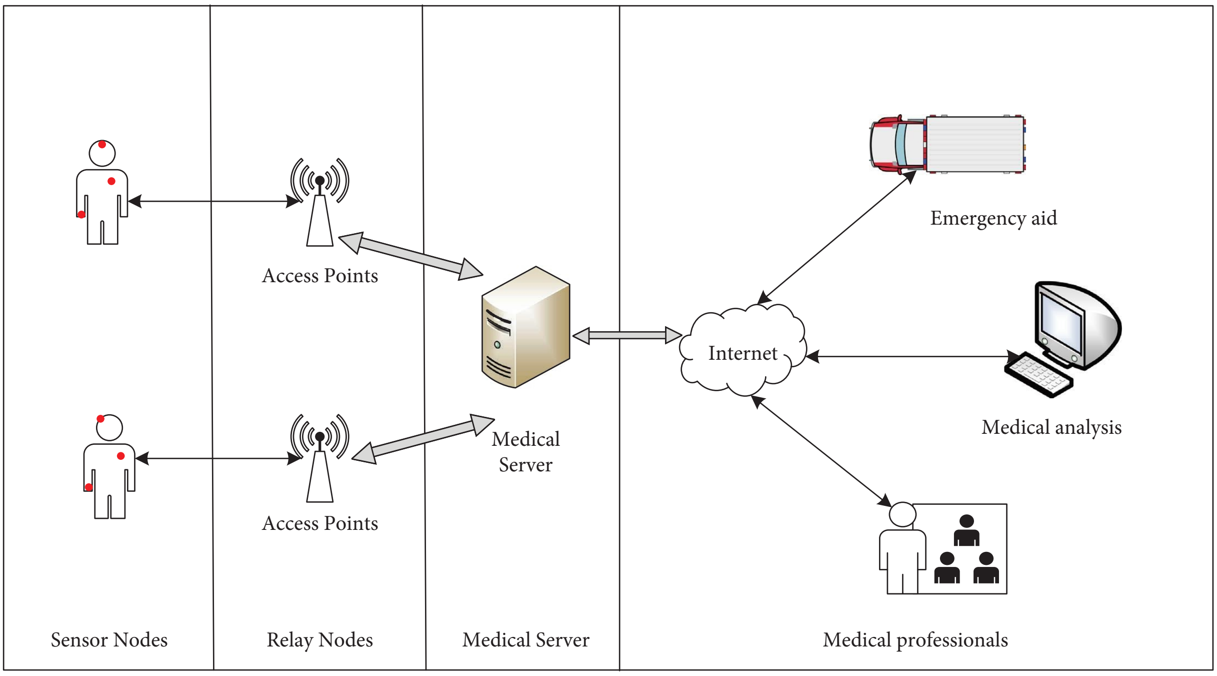


Figure 1: System model.

(3) AR can capture $SN_j$ and RN and then retrieve all data stored in their memory

(4) We adopt Dolev–Yao threat model [34] and assume that the public channel is insecure

*2.3. Physically Unclonable Function.* As a hardware security technology, a physically unclonable function (PUF) can be regarded as the "digital fingerprint" of the chip [35]. It uses the inherent physical differences to produce a specific unclonable response to a given challenge. Therefore, it is difficult to be predicted before production and cloned after production. It has broad application prospects in the field of security. According to the same challenge, the response of PUF can remain unchanged under different conditions. Any detection or observation of PUF will change the circuit characteristics, and the output of PUF will also change. Therefore, PUF is often used to protect crucial data in cryptography [36].

All notations in our paper are illustrated in Table 1.

Table 1: Notations.

| Notations | Description |
|---|---|
| $SN_j$ | $j^{th}$ sensor node |
| $RN$ | Relay node |
| $MS$ | Medical server node |
| $SA$ | Server administrator |
| $AR$ | The adversary |
| $id_j$, $id_R$ | Identity of $SN_j$/identity of $RN$ |
| $K_{MS}$, $Q$ | Secret key and public key of $MS$, where $Q = K_{MS} \bullet P$ |
| $K_{SH}$ | Session key |
| $r$, $P_{R1}$, $P_{R2}$ | Random integers |
| $b_j$ | Random number generated by $SN_j$ |
| $m$, $r^{new}$ | Random integers generated by $MS$ |
| $a_j$, $r$, $P_{R1}$, $P_{R2}$ | Random integers generated by $SA$ |
| $T, T_1, T_2, T_3, T_4$ | Timestamps |
| $P$ | The base point of the elliptic curve |
| $\oplus$ | XOR operation |
| PUF$(\bullet)$ | Physically unclonable function |
| $h(\bullet)$ | Hash function |
| $\Delta T$ | The maximum transmission delay |

## 3. Drawbacks of Alzahrani et al.'s Scheme

*3.1. Review of Alzahrani et al.'s Scheme.* We briefly review Alzahrani et al.'s [30] anonymous authentication protocol, which involves three steps: (1) system initialization; (2) device registration; (3) mutual authentication and key agreement. SA performs step (1) and step (2) through a private channel as follows.

*3.1.1. System Initialization*

(i) SA generates a long-term master secret key $K_{MS}$ for MS

(ii) Subsequently, MS reserves the master secret key $K_{MS}$

*3.1.2. Devices Registration*

(i) SA selects three random integers $r$, $P_{R1}$, $P_{R2}$, and an identity $id_j$ for the sensor node $SN_j$ and reserves tuple $<id_j, P_{R1}, P_{R2}>$ in the memory of MS

(ii) SA computes $x_{Nj} = r \oplus K_{MS}$, $y_{Nj} = id_j \oplus h(K_{MS}, r)$

(iii) SA reserves tuple $<id_j, x_{Nj}, y_{Nj}, P_{R1}, P_{R2}>$ in the memory of $SN_j$

(iv) Finally, the verification table of MS is $<id_j, P_{R1}, P_{R2}, id_R>$

*3.1.3. Mutual Authentication and Key Agreement.* The communications between $SN_j$ and MS are as follows:

(i) $SN_j$ creates a current timestamp $T_1$ and computes the validation $Vid_j = h(id_j, x_{Nj}, y_{Nj}, P_{R2}, T_1)$, where $id_j$ is $SN_j$'s identity, $x_{Nj} = r \oplus K_{MS}$, $y_{Nj} = id_j \oplus h(K_{MS}, r)$, $P_{R2}$ denotes a random integer, and the current timestamp is denoted as $T_1$.

(ii) $SN_j$ submits Message1 tuple $<x_{Nj}, y_{Nj}, Vid_j, T_1>$ to RN.

(iii) RN appends its identity $id_R$ and forwards the Message2 tuple $<x_{Nj}, y_{Nj}, Vid_j, T_1, id_R>$ to MS.

(iv) MS scans the identity $id_R$ and finishes the session if no record is found in its memory. Otherwise, MS creates the current timestamp $T_2$ and checks if $|T_2 - T_1| \le \Delta T$, and if not, finishes the session. Otherwise, MS computes $r^* = x_{Nj} \oplus K_{MS}$, $id_j^* = y_{Nj} \oplus h(K_{MS}, r^*)$. MS checks the validity of the identity $id_j^*$, if so, MS extracts the tuple $<id_j^*, P_{R1}, P_{R2}>$ from its memory, computes $Vid_j^* = h(id_j^*, x_{Nj}, y_{Nj}, P_{R2}, T_1)$, and checks $Vid_j^* ?= Vid_j$. If so, MS generates random nonce $m$ and $r^{new}$ and computes $s = id_j^* \oplus y_{Nj}$, $j = id_j^* \oplus x_{Nj}$, $v = m \oplus s$, $x_{Nj}^{new} = r^{new} \oplus K_{MS}$, $y_{Nj}^{new} = id_j^* \oplus h(K_{MS}, r^{new})$, $g = h(m, s, j, P_{R2})$, $u = x_{Nj}^{new} \oplus g$, $n = y_{Nj}^{new} \oplus g$, $\Delta = h(m, id_j^*, s, x_{Nj}^{new}, y_{Nj}^{new})$, and the session key $K_{SH} = h(m, j, P_{R1}, P_{R2})$. Afterwards, MS sends the Message3 tuple $<v, u, \Delta, n, id_R>$ to RN. MS displaces $P_{R1}$ with $P_{R2}$ and $P_{R2}$ with $K_{SH}$.

(v) RN removes its identity $id_R$ and forwards the Message 4 tuple $<v, u, \Delta, n>$ to $SN_j$.

(vi) $SN_j$ computes $s^* = id_j \oplus y_{Nj}$, $m^* = v \oplus s^*$, $j^* = id_j \oplus x_{Nj}$, $g^* = h(m^*, s^*, j^* P_{R2})$, $x_{Nj}^{new+} = u \oplus g^*$, $y_{Nj}^{new+} = n \oplus g^*$, $\Delta^* = h(m^*, id_j, s^*, x_{Nj}^{new+}, y_{Nj}^{new+})$. Afterwards, $SN_j$ checks $\Delta^* ?= \Delta$. If so, $SN_j$ computes the session key $K_{SH} = h(m^*, j^*, P_{R1}, P_{R2})$. $SN_j$ displaces $x_{Nj}$ and $y_{Nj}$ with $x_{Nj}^{new+}$ and $y_{Nj}^{new+}$, and stores them in its memory. Finally, $SN_j$ displaces $P_{R1}$ with $P_{R2}$ and $P_{R2}$ with $K_{SH}$.

### 3.2. Drawbacks

*3.2.1. Off-Line Identity Guessing Attack.* Supposing an adversary (AR) can eavesdrop on the conversation between $SN_j$ and MS. AR intercepts the first round of $x_{Nj-1}$, $y_{Nj-1}$, and the second round of $x_{Nj-2}$, $y_{Nj-2}$, where $x_{Nj-2}$ and $y_{Nj-2}$ are the first round of $x_{Nj-1}^{new+}$ and $y_{Nj-1}^{new+}$. AR computes $\Delta^* = h(m^*, id_j, s^*, x_{Nj-1}^{new+}, y_{Nj-1}^{new+})$, where $m^* = v \oplus s^*$, $s^* = id_j \oplus y_{Nj}$. Only $id_j$ in $\Delta^*$ is unknown, and AR guesses $id_j$ to verify if $\Delta^* ?= \Delta$. If so, AR obtains $id_j$ successfully. Otherwise, guesses $id_j$ again.

*3.2.2. Desynchronization Attack.* If AR intercepts Message4 and drops it, the $SN_j$ will miss it. The insecurity is that MS has updated $x_{Nj}$, $y_{Nj}$, $P_{R1}$, $P_{R2}$, but $SN_j$ has not. This will make every subsequent authentication process between $SN_j$ and MS fail.

*3.2.3. Stolen-Verifier Attack.* If the verifier table $<id_j, P_{R1}, P_{R2}, id_R>$ of MS is stolen, AR can obtain all the data in it. AR eavesdrops on the communication between $SN_j$ and MS, intercepts Message1 tuple $<x_{Nj}, y_{Nj}, Vid_j, T_1>$, Message 4 tuple $<v, u, \Delta, n>$, computes $s^* = id_j \oplus y_{Nj}$, $m^* = v \oplus s^*$, and $j^* = id_j \oplus x_{Nj}$, and computes the session key $K_{SH} = h(m^*, j^*, P_{R1}, P_{R2})$. That is, AR can obtain the session key.

*3.2.4. Known-Key Attack.* If the session keys of two consecutive rounds are leaked, AR will get $P_{R1-3}$ and $P_{R2-3}$ of the third round. According to identity guessing attacks, AR obtains the SN's identity $id_j$. In the third round of protocol execution, AR intercepts message 1 and message 4 and computes $s^* = id_j \oplus y_{Nj-3}$, $m^* = v \oplus s^*$, $g^* = h(m^*, s^*, j^* P_{R2-3})$, $x_{Nj-3}^{new+} = u \oplus g^*$, $y_{Nj-3}^{new+} = n \oplus g^*$, $K_{SH} = h(m^*, j^*, P_{R1-3}, P_{R2-3})$. Therefore, the session key of the subsequent round will be obtained by the AR.

*3.2.5. No Perfect Forward Security.* If the long-term secret key $K_{MS}$ and short-term secret key $P_{R1}$ and $P_{R2}$ of the Alzahrani et al.'s [30] scheme are leaked, AR calculates $r^* = x_{Nj} \oplus K_{MS}$, $id_j = y_{Nj} \oplus h(K_{MS}, r^*)$. Then, AR calculates $s^* = id_j \oplus y_{Nj}$, $m^* = v \oplus s^*$, $g^* = h(m^*, s^*, j^*, P_{R2})$. Finally, AR can compute the session key $K_{SH} = h(m^*, j^*, P_{R1}, P_{R2})$. Therefore, it doesn't achieve perfect forward secrecy.

## 4. Proposed Protocol

A security-enhanced protocol is presented, which involves three steps: (1) system initialization; (2) device registration; (3) mutual authentication and key agreement. SA executes initialization and registration steps through a private channel as follows.

*4.1. Initialization.* SA executes as follows:

(1) The master secret key $K_{MS}$ is generated by SA

(2) Subsequently, MS accepts the master secret key $K_{MS}$ via a secure channel and keeps it secretly

(3) SA chooses an elliptic curve $E_c(\alpha, \beta)$ of large order. $P$ is a base point. SA computes $Q = K_{MS} \bullet P$. Afterwards, SA chooses a hash function $h(\bullet)$.

*4.2. Registration.* The registration phase can be described as follows:

(1) SA chooses the random integer $a_j$ and the identity $id_j$ for the sensor node $SN_j$, an identity $id_R$ for RN, and reserves $id_j$ and $id_R$ in the memory of MS

(2) SA computes $x_{Nj} = a_j \oplus h(K_{MS}, T_j)$, $y_{Nj} = id_j \oplus h(K_{MS}, a_j, T_j)$, $MH_j = h(id_j, K_{MS})$, where $T_j$ is the current timestamp, and $K_{MS}$ is MS's secret key

(3) SA reserves the tuple <$id_j$, $x_{Nj}$, $y_{Nj}$, $MH_j$, $T_j$> in the memory of $SN_j$, and $SN_j$ generates a challenge $\text{Cha}_j$ and computes $\text{Res}_j = \text{PUF}(\text{Cha}_j)$, $ST_j = h(\text{Res}_j) \oplus MH_j$, where PUF is deployed in the sensor node $SN_j$

(4) Finally, $SN_j$ stores $\{id_j, x_{Nj}, y_{Nj}, \text{ST}_j, \text{Cha}_j, \text{T}_j\}$, and the verification table of MS is $\{id_R, id_j\}$

*4.3. Mutual Authentication and Key Agreement.* This phase is shown in Figure 2.

(1) $SN_j$ chooses the random integer $b_j$ and the timestamp $T_1$ and calculates $MH_j = h(\text{PUF}(\text{Cha}_j)) \oplus ST_j$, $A_1 = b_j \bullet P$, $A_2 = b_j \bullet Q$, $Vid_j = h(id_j, x_{Nj}, y_{Nj}, A_1, A_2, h(A_2, MH_j), T_j, T_1)$.

(2) $SN_j$ submits the Message1 tuple <$x_{Nj}$, $y_{Nj}$, $Vid_j$, $A_1$, $T_j$, $T_1$> to RN.

(3) RN appends its identity $id_R$ and forwards the Message 2 tuple <$x_{Nj}$, $y_{Nj}$, $Vid_j$, $A_1$, $T_j$, $T_1$, $id_R$> to MS.

(4) MS scans the identity $id_R$ and finishes the session if no record is found in its memory. Otherwise, MS creates the current timestamp $T_2$ and checks if $|T_2 - T_1| \le \Delta T$, and if not, finishes the session. Otherwise, MS computes $a_j = x_{Nj} \oplus h(K_{MS}, T_j)$, $id_j^* = x_{Nj} \oplus h(K_{MS}, a_j, T_j)$. MS calculates $A_2^* = K_{MS} \bullet A_1$, $Vid_j^* = h(id_j^*, x_{Nj}, y_{Nj}, A_1, A_2^*, h(A_2^*, h(id_j^*, K_{MS})), T_j, T_1)$ and checks $Vid_j^* ?= Vid_j$. If so, MS creates random numbers $a_i$ and $b_i$. Next, MS computes $A_3 = b_i \bullet P$, $A_4 = b_i \bullet A_1$, $x_{Nj}^{\text{new}} = a_i \oplus h(K_{MS}, T_2)$, $y_{Nj}^{\text{new}} = id_j^* \oplus h(K_{MS}, a_i, T_2)$, $\mu = x_{Nj}^{\text{new}} \oplus h(A_2^*, h(id_j^*, K_{MS}), T_2)$, $\lambda = y_{Nj}^{\text{new}} \oplus h(T_2, A_2^*, h(id_j^*, K_{MS}))$, the session key $K_{SH} = h(A_1, A_2^*, A_3, A_4, id_j^*, T_2)$, and $\Delta = h(x_{Nj}^{\text{new}}, y_{Nj}^{\text{new}}, K_{SH}, T_2)$. Afterwards, MS sends the Message3 tuple <$\mu$, $\lambda$, $\Delta$, $A_3$, $T_2$, $id_R$> to RN.

(5) RN removes its identity $id_R$ and forwards the Message4 tuple <$\mu$, $\lambda$, $\Delta$, $A_3$, $T_2$> to $SN_j$.

(6) $SN_j$ creates the current timestamp $T_3$ and checks if $|T_3 - T_2| \le \Delta T$, and if not, finishes the session. Otherwise, $SN_j$ computes $A_4^* = b_j \bullet A_3$, $x_{Nj}^{\text{new}*} = \mu \oplus h(A_2, MH_j, T_2)$, $y_{Nj}^{\text{new}*} = \lambda \oplus h(T_2, A_2, MH_j)$, $K_{SH} = h(A_1, A_2, A_3, A_4^*, id_j, T_2)$, $\Delta^* = h(x_{Nj}^{\text{new}*}, y_{Nj}^{\text{new}*}, K_{SH}, T_2)$. $SN_j$ checks if $\Delta^* ?= \Delta$. If so, $SN_j$ successfully establishes the session key $K_{SH}$ with MS and updates <$x_{Nj}$, $y_{Nj}$, $\text{T}_j$> with <$x_{Nj}^{\text{new}*}$, $y_{Nj}^{\text{new}*}$, $T_2$>.

## 5. Informal Security Analysis

*5.1. Off-Line Identity Guessing Attack.* If an adversary(AR) can eavesdrop on the open channel and guess $id_j$ of the sensor node $\text{SN}_j$, it is not feasible for him/her to verify whether $Vid_j^* ?= Vid_j$ is correct or not without knowing $A_2$, where $A_2 = b_j \bullet K_{MS} \bullet P$, $Vid_j = h(id_j, x_{Nj}, y_{Nj}, A_1, A_2, h(A_2, MH_j), T_j, T_1)$, $MH_j = h(id_j, K_{MS})$. Because of computational Diffie–Hellman problem (CDHP), AR cannot compute $A_2 = b_j \bullet K_{MS} \bullet P$ from $A_1 = b_j \bullet P$ and $Q = K_{MS} \bullet P$. Therefore, off-line identity guessing attack is infeasible.

*5.2. Desynchronization Attack.* In the improved protocol, $x_{Nj}$ and $y_{Nj}$ are updated as $x_{Nj}^{\text{new}}$ and $y_{Nj}^{\text{new}}$ on the side of the MS. Even if AR intercepts the Message4, it has no impact on the next session between the sensor node $SN_j$ and the MS.

*5.3. Stolen-Verifier Attack.* Stolen-verifier attack means that an adversary can obtain verification table except the secret key from MS by trespassing on the device or side channel attack and then launch attacks. In the proposed scheme, the verification table of MS only contains the identities $id_j$ and $id_R$ of $SN_j$ and $RN$. So the adversary cannot launch any attacks even if he or she obtains these identities. Thus, the protocol defends against stolen-verifier attacks.

*5.4. Known-Key Attack.* Assuming that AR knows the session key $K_{SH} = h(A_1, A_2, A_3, A_4^*, id_j, T_2)$, because $K_{SH}$ only contained in $\Delta^* = h(x_{Nj}^{\text{new}*}, y_{Nj}^{\text{new}*}, K_{SH}, T_2)$, so AR cannot launch any attack.

*5.5. Smart Card Lost Attack.* By the side-channel attack, AR is able to get all data reserved in the smart card when it is lost, and then launch attacks. However, in our protocol, smart card isn't used, so the protocol defends against the smart card lost attack.

*5.6. Sensor Node Captured Attack.* In the improved protocol, the sensor node $SN_j$ stores $\{id_j, x_{Nj}, y_{Nj}, ST_j, \text{Cha}_j, T_j\}$, where $id_j$ is $SN_1$'s identity, $x_{Nj} = a_j \oplus h(K_{MS}, T_j)$, $y_{Nj} \oplus id_j \oplus h(K_{MS}, a_j, T_j)$, $ST_j = h(\text{PUF}(\text{Cha}_j)) \oplus MH_j$, $\text{Cha}_j$ is the challenge of PUF, $T_j$ is the timestamp, and $K_{MS}$ is the secret key of MS. Assuming that the sensor node $SN_j$ is captured by AR, he/she cannot obtain the secret parameter $MH_j$ to impersonate $SN_j$ because of PUF. In addition, AR cannot obtain the secret key $K_{MS}$. Therefore, the sensor node captured attack cannot influence the security of nodes and the sensor network.

*5.7. Anonymity and Unlinkability.* The identity $id_j$ of the sensor node $SN_j$ is in Message 1 = $\{x_{Nj}, y_{Nj}, Vid_j, A_1, T_j, T_1\}$ and transmitted via an open channel, where $Vid_j = h(id_j, x_{Nj}, y_{Nj}, A_1, A_2, h(A_2, MH_j)T_j, T_1)$, $MH_j = h(id_j, K_{MS})$, $y_{Nj} = id_j \oplus h(K_{MS}, a_j, T_j)$. So an adversary cannot compute the identity $id_j$ of the sensor $SN_j$ because he can not know the secret key $K_{MS}$ of MS. Thus, our scheme achieves anonymity. Moreover, because each session will generate new $b_j$ and $T_j$, the identity $id_j$ of the sensor node $SN_j$ cannot be tracked by AR.

*5.8. Perfect Forward Secrecy.* If AR obtains all the secret information of the sensor node $SN_j$ and the long-term master secret key $K_{MS}$ of MS, because of CDHP, he/she still

| $SN_j$ | $RN$ | $MS$ |
|---|---|---|
| 1. Generates a random integer $b_j$ and a current timestamp $T_1$.<br>Computes $\mathrm{MH_j} = \mathrm{h(PUF(Cha_j))} \oplus \mathrm{ST_j}$, $A_1=b_j \cdot P$, $A_2=b_j \cdot Q$, $Vid_j=h(id_j, x_{N_j}, y_{N_j}, A_1, A_2, h(A_2, \mathrm{MH_j}), \mathrm{T_j}, \mathrm{T_1})$.<br>Message1=$\{x_{N_j},y_{N_j},Vid_j,A_1,T_j,T_1\}$ → | | |
| | 2. Appends its identity $id_R$, and forwards the Message2.<br>Message2=$\{x_{N_j},y_{N_j},Vid_j,A_1,T_j,T_1,id_R\}$ → | |
| | | 3. Checks the validity of the identity $id_R$.<br>Creates the current timestamp $T_2$ and checks if $\lvert T_2 - T_1 \rvert \le \Delta T$.<br>Computes $a_j = x_{N_j} \oplus h(K_{MS}, T_j)$, $id_j^* = x_{N_j} \oplus h(K_{MS}, a_j, T_j)$.<br>Calculates $A_2^* = K_{MS} \cdot A_1$, $Vid_j^* = h(id_j^*, x_{N_j}, y_{N_j}, A_1, A_2^*, h(A_2^*, h(id_j^*, K_{MS})), T_j, T_1)$.<br>Checks $Vid_j^* \ ?= Vid_j$.<br>Creates $a_i$ and $b_i$, both of which are integers.<br>Computes $A_3 = b_i \cdot P$, $A_4 = b_i \cdot A_1$, $x_{N_j}^{new} = a_i \oplus h(K_{MS}, T_2)$, $y_{N_j}^{new} = id_j^* \oplus h(K_{MS}, a_i, T_2)$, $\mu = x_{N_j}^{new} \oplus h(A_2^*, h(id_j^*, K_{MS}), T_2)$, $\lambda = y_{N_j}^{new} \oplus h(T_2, A_2^*, h(id_j^*, K_{MS}))$, $K_{SH} = h(A_1, A_2^*, A_3, A_4, id_j^*, T_2)$, $\Delta = h(x_{N_j}^{new}, y_{N_j}^{new}, K_{SH}, T_2)$.<br>Message3=$\{\mu,\lambda,\Delta,A_3,T_2,id_R\}$ ← |
| | 4. Removes its identity $id_R$, and forwards the Message4.<br>Message4=$\{\mu,\lambda,\Delta,A_3,T_2\}$ ← | |
| 5. Creates the current timestamp $T_3$.<br>checks if $\lvert T_3 - T_2 \rvert \le \Delta T$.<br>Computes $A_4^* = b_j \cdot A_3$, $x_{N_j}^{new*} = \mu \oplus h(A_2, MH_j, T_2)$, $y_{N_j}^{new*} = \lambda \oplus h(T_2, A_2, MH_j)$, $K_{SH} = h(A_1, A_2, A_3, A_4^*, id_j, T_2)$, $\Delta^* = h(x_{N_j}^{new*}, y_{N_j}^{new*}, K_{SH}, T_2)$.<br>Checks if $\Delta^* \ ?= \Delta$.<br>If so, $\mathrm{SN_j}$ successfully establishes the session key $K_{SH}$ with MS. Update $< x_{N_j}, y_{N_j}, T_j >$ with $< x_{N_j}^{new*}, y_{N_j}^{new*}, T_2 >$ | | |

Figure 2: Mutual authentication and key agreement phase.

cannot successfully calculate $K_{SH} = h(A_1, A_2, A_3, A_4^*, id_j, T_2)$ without knowing $A_4^*$. Therefore, the protocol achieves perfect forward secrecy.

*5.9. Impersonation Attack.* This attack means that AR can impersonate a legal user to generate and send a message, and the message can be passed through the authentication by the

```
(*--Channel--*)
free PC:channel [private].(*Public channel*)

(*--Types--*)
type key.
type nonce.
type timestamp.

(*--Constants&Variables--*)
const P: bitstring.(*--The base point--*)
free KMS:key[private]. (*--The master secret key of server--*)
free IDj: bitstring[private]. (*-- The identity of Sensor Node--*)
free KSHi: key[private]. (*--The session key of server--*)
free KSHj: key[private]. (*--The session key of sensor--*)

(*--Constructors--*)
fun h( bitstring ) : bitstring.(*--Hash operation--*)
fun CON(bitstring, bitstring): bitstring.(*--Concat operation--*)
fun XOR(bitstring,bitstring):bitstring.(*--XOR operation--*)
fun ECC( bitstring , bitstring ) : bitstring.(*--ECC operation--*)
fun bit_timestamp(timestamp): bitstring.(*--Bit operation--*)
fun bit_key(key): bitstring.(*--Bit operation--*)
fun bit_nonce(nonce): bitstring.(*--Bit operation--*)
fun key_bit(bitstring): key.(*--Bit operation--*)
fun PUF( bitstring ) : bitstring.(*--PUF operation--*)
fun timestampcheck(bitstring, bool): bool(*--Check timestamp operation--*)

(*--Destructors & Equations--*)
reduc forall T: bitstring;
timestampcheck(T, true) = true
otherwise forall T: bitstring;
timestampcheck(T, false) = false.(*--Check timestamp Fresh operation--*)
equation forall a:bitstring,b:bitstring; XOR(XOR(a,b),b)=a.(*--XOR operation--*)
```

Figure 3: Definitions.

receiver. That is to say, the receiver confirms that the message is initiated by a legitimate user. In our protocol, AR impersonates the sensor node $SN_j$ to generate and send $\{x_{Nj}, y_{Nj}, Vid_j, A_1, T_j, T_1\}$ to RN, where $x_{Nj} = a_j \oplus h(K_{MS}, T_j)$, $y_{Nj} = id_j \oplus h(K_{MS}, a_j, T_j)$, $Vid_j = h(id_j, x_{Nj}, y_{Nj}, A_1, A_2, h(A_2, MH_j) T_j, T_1)$, $K_{MS}$ is MS's secret key, and $T_1$ is the timestamp. The adversary cannot forge $x_{Nj}$ and $y_{Nj}$ without knowing $K_{MS}$. On the other hand, the adversary cannot compute $MH_j$ even if he/she can obtain all data stored in $MH_j$ due to the property of PUF. Therefore, the adversary cannot generate the valid $Vid_j$.

*5.10. Replay Attack.* If AR can obtain a message and replay it to the receiver, the message can be passed through the authentication of the receiver. In the proposed scheme, the timestamps and random nonce are used, so the protocol defends against the replay attack.

## 6. Formal Security Analysis

*6.1. Formal Verification Using ProVerif.* As an automated verification cryptographic scheme tool, ProVerif [37] is founded on the Dolev–Yao model and Prolog language. It verifies many cryptographic primitives, for example, public-key cryptography, hash function, and equations. When using ProVerif tool for verifying insecure cryptographic protocols, the tool will give a corresponding attack sequence.

The open channel, types, constants, variables, constructors, and destructors of our proposed protocol are represented in Figure 3. We designed four events for the improved protocol, which are BeginSNj(), BeginMS(), EndSNj(), and EndMS() as depicted in Figure 4. BeginSNj() represents that the sensor node $SN_j$ begins the key agreement session with MS. BeginMS() represents that MS starts the key agreement session with $SN_j$. $SN_j$ successfully established a session key with MS, which is indicated as EndSNj(). EndMS() represents MS successfully established a session key with the sensor node $SN_j$.

Queries are shown in Figure 5. Figures 6 and 7 are exhibiting the processes of the sensor node $SN_j$ and MS. The main process is represented in Figure 8.

For testifying the improved scheme's correctness, we propose some queries and finally implement them through simulation, as shown in Figure 9.

Results (1)–(4) proved that the secret parameters and session key are secure, and sensor nodes are anonymous in our protocol. Results (5)-(7) showed that the two processes began and terminated successfully in sequence.

*6.2. Formal Security Proof.* After identifying the random oracle model (ROM), we calculate the advantage of breaking our protocol $\mathcal{P}$ by the adversary $A$. The notions of ROM are clarified as follows.

```
(*--Events--*)
event BeginSNj (bitstring).
event EndSNj (bitstring).
event BeginMS(bitstring).
event EndMS(bitstring).
```

FIGURE 4: Events.

```
(*--Queries--*)
query attacker(KMS).
query attacker(KSHj).
query attacker(KSHi).
query attacker(IDj).
query IDj:bitstring;event(EndSNj(IDj))==>event(BeginMS(IDj)).
query IDj:bitstring;inj-event(EndMS(IDj))==>inj-event(BeginSNj(IDj)).
query IDj:bitstring;inj-event(EndSNj(IDj))==>inj-event(BeginMS(IDj)).
```

FIGURE 5: Queries.

```
(*--The process of sensor node SNj--*)
let SNj(IDj:bitstring,P:bitstring,Q:bitstring,XNj:bitstring,YNj:bitstring,Chaj:bitstring,STj: bitstring,Tj:bitstring)=
        event BeginSNj (IDj);
        new bj_1:nonce;
        new T1_1:timestamp;
        let T1=bit_timestamp(T1_1) in
        let bj=bit_nonce(bj_1) in
        let A1=ECC(bj,P) in
         let MHj=XOR(Hash(PUF(Chaj)),STj) in
        let A2=ECC(bj,Q) in
        let Vidj=h(CON(IDj,CON(XNj,CON(YNj,CON(A1,CON(h(CON(A2,MHj)), CON(Tj, CON(Tj,T1)))))))) in
        out(PC,(XNj,YNj,Vidj,A1, Tj,T1));
        in(PC,(u:bitstring,L:bitstring,D:bitstring,A3:bitstring,T2:bitstring));
        if timestampcheck(T3, true) then
                let A4=ECC(bj,A3) in
                let XNjn_1=XOR(u,h(CON(A2,CON(MHj,T2)))) in
                let YNjn_1=XOR(L,h(CON(T2,CON(A2,MHj)))) in
                let KSHj_1=h(CON(A1,CON(A2,CON(A3,CON(A4,T1))))) in
                let D_2=h(CON(XNjn_1,CON(YNjn_1,CON(KSHj_1,T2)))) in
                let KSHj=key_bit(KSHj_1) in
                if D_2=D then
                event EndSNj(IDj).
```

FIGURE 6: The process of the sensor node $\mathbf{SN_j}$.

*6.2.1. Participants & States.* Three participants $P$ is in $\mathscr{P}$, sensor node $SN$, relay node $RN$, and medical server node $MS$. In *i-th* instance, $P$, $SN$, $RN$, and $MS$ are recorded as $\text{INS}_P^i$, $\text{INS}_{SN}^i$, $\text{INS}_{RN}^i$, and $\text{INS}_{MS}^i$, respectively. The oracles in ROM have only three states: Accept, Reject, and ⊥. Accept represents a correct message that is received by an oracle. If the message is illegal, the oracle in Reject. ⊥ means both the conditions above have not occurred.

If the oracle $\text{INS}_{SN}^i$ ($\text{INS}_{MS}^i$) is in Accept, and the session key $K_{SN}^i$ ($K_{MS}^i$) has been agreed with $\text{INS}_{MS}^i$ ($\text{INS}_{SN}^i$), then $\text{INS}_{SN}^i$ ($\text{INS}_{MS}^i$) gets the session identity $\text{SID}_{SN}^i$ ($\text{SID}_{MS}^i$), and its participant's identity is $\text{PID}_{SN}^i$ ($\text{PID}_{MS}^i$).

*6.2.2. Partnering.* If $\text{INS}_{SN}^i$ and $\text{INS}_{MS}^i$ are in Accept, the session key is negotiated. Two partners meet below requirements:

(1) $K_{SN}^i = K_{MS}^i$

(2) $\text{SID}_{SN}^i = \text{SID}_{MS}^i$

(3) $\text{PID}_{SN}^i = \text{INS}_{MS}^i$, $\text{PID}_{MS}^i = \text{INS}_{SN}^i$

*6.2.3. Queries.* Queries can emulate multiple attacks.

*Execute* ($\text{INS}_P^i$)if the query is lunched by $A$, he/she gets all the transcripts.

*Send* ($\text{INS}_P^i$, *Message*): which simulates that *Message* is sent to $\text{INS}_P^i$. If the message is correct, $\text{INS}_P^i$ responses $A$, else, the message is ignored.

*Reveal* ($\text{INS}_{SN}^i$, $\text{INS}_{MS}^i$)if $\text{INS}_{SN}^i$ and $\text{INS}_{MS}^i$ are in the state Accept, the session key has been agreed, and the query *Test* has not been executed yet. Then, the session key will be revealed by this query. Else, return null.

*Corrupt* ($\text{INS}_{SN}^i$)which simulates the attack of intercepting $SN_j$ and returns the stored information $\{id_j, x_{Nj}, y_{Nj}, ST_j, \text{Cha}_j, \text{PUF}, T_j\}$ in it.

*Test* ($\text{INS}_{SN}^i$)this query produces a random bit $r$, which is performed no more than once. If $r = 1$ and the session key has been agreed, the real session key is returned to $A$, else, the query returns a random session key.

```
(*--The process of Server MS--*)
let mserver(IDj:bitstring,P:bitstring,Q:bitstring,KMS:bitstring,Tj:bitstring)=
        new T2_1:timestamp;
        let T2=bit_timestamp(T2_1) in
        in(PC,(XNj:bitstring,YNj:bitstring,Vidj:bitstring,A1:bitstring,T1:bitstring));
        if timestampcheck(T1,true) then

                let aj=XOR(XNj,h(CON(KMS,Tj))) in
                let IDj_1=XOR(XNj,h(CON(KMS,CON(aj,Tj)))) in
                if IDj_1=IDj then
                        let A2=ECC(KMS,A1) in
                        let     Vidj_1=h(CON(IDj,CON(XNj,CON(YNj,CON(A1,CON(A2,h(CON(CON(A2,h(CON(IDj,KMS))),
CON(Tj,T1))))))))) in
                        event BeginMS(IDj);
                        new ai_1:nonce;
                        new bi_1:nonce;
                        let ai=bit_nonce(ai_1) in
                        let bi=bit_nonce(bi_1) in
                        let A3=ECC(bi,P) in
                        let A4=ECC(bi,A1) in
                        let XNjn1=XOR(ai,h(CON(KMS,T2))) in
                        let YNjn1=XOR(IDj,h(CON(KMS,CON(ai,T2)))) in
                        let u=XOR(XNjn1,h(CON(A2,h(CON(CON(IDj,KMS),T2))))) in
                        let L=XOR(YNjn1,h(CON(T2,CON(A2,h(CON(IDj,KMS)))))) in
                        let KSHi_1=h(CON(A1,CON(A2,CON(A3,CON(A4,CON(IDj,T2)))))) in
                        let D=h(CON(XNjn1,CON(YNjn1,CON(KSHi_1,T2)))) in
                        let KSHi=key_bit(KSHi_1) in
                        out(PC,(u,L,D,A3,T2));
                        event EndMS(IDj).
```

Figure 7: The process of MS.

```
(*--Main process--*)
process
        let KMSn=bit_key(KMS) in
        let Q=ECC(KMSn,P) in
        new aj_1:nonce;
        let aj=bit_nonce(aj_1) in
        new Tj_1:timestamp;
        let Tj=bit_timestamp(Tj_1) in
        let XNj=XOR(aj,h(CON(KMSn,Tj))) in
        let YNj=XOR(IDj,h(CON(KMSn,CON(aj,Tj)))) in
        let MHj=h(CON(IDj,KMSn)) in
        (!SNj(IDj,P,Q,XNj,YNj,MHj)|!mserver(IDj,P,Q,KMSn,Tj))
```

Figure 8: Main process.

```
Verification summary:

Query not attacker(KMS[]) is true.

Query not attacker(KSHj[]) is true.

Query not attacker(KSHi[]) is true.

Query not attacker(IDj[]) is true.

Query event(EndSNj(IDj_4)) ==> event(BeginMS(IDj_4)) is true.

Query inj-event(EndMS(IDj_4)) ==> inj-event(BeginSNj(IDj_4)) is true.

Query inj-event(EndSNj(IDj_4)) ==> inj-event(BeginMS(IDj_4)) is true.
```

Figure 9: Results.

*6.2.4. Freshness.* If the ensuing requirements are met, $\text{INS}_P^i$ can be defined as fresh.

(1) $\text{INS}_{SN}^i$ and $\text{INS}_{MS}^i$ are in the state Accept
(2) Reveal has not been executed
(3) Corrupt is executed at most once

*6.2.5. Semantic Security.* The random bit $r$ in *Test* query determines the output of *Test*. Meanwhile, $A$ generates a random $r'$, if $r' = r$, $A$ knows if the output is session key. The advantage of guessing the correct bit is $\text{Adv}_{\mathscr{P}}^A = |2\Pr[r = r'] - 1| = |2\Pr[\text{suc}(A)] - 1|$. $\mathscr{P}$ is secure when $\text{Adv}_{\mathscr{P}}^A < \eta$, where $\eta$ is sufficiently small.

CDHP: the CDHP is specified that given $P$, $aP$, and $bP$, computing $abP$ is computationally infeasible in probabilistic polynomial time (PPT). $P$ is the generator point, $a, b \in Z_p$. Subsequently, the advantage of solving CDHP is $\text{Adv}_A^{\text{CDHP}} = \Pr\ [A(P, aP, bP) = abP\colon P \in E(F_p); a, b \in Z_p]$, $\text{Adv}_A^{\text{CDHP}} < \eta$.

**Theorem 1.** *Suppose the adversary A tends to break the proposed scheme $\mathscr{P}$ in PPT. The queries Execute, Send, and Hash are executed $q_E$, $q_S$, and $q_H$ times, respectively. Query Test is allowed to be executed at most once. $l_h$ is the bit-length of the hash operation's the output. $n = 2^{l_t}$, where $l_t$ is the average length of other transcripts. The advantage of breaking $\mathscr{P}$ by A in PPT can be expressed as follows:*

$$\text{Adv}_A^{\mathscr{P}} \le \frac{(q_S + q_E)^2}{n} + \frac{q_H^2}{2^{l_h}} + 2\text{Adv}_A^{\text{CDHP}} + 2\text{Adv}_A^{\text{PUF}}. \quad (1)$$

*Proof.* To simulate the attacks on $\mathscr{P}$, we define various games $\text{Game}_i\,(0 < i < 3)$. The event $\text{Success}_A^i\,(0 < i < 3)$ corresponding to $\text{Game}_i$ means that $A$ completes his/her goal in $\text{Game}_i$.

$\text{Game}_0$: which simulates the real attack, at the first, the probability of $A$ cracking $\mathscr{P}$ is

$$\text{Adv}_A^{\mathscr{P}} = \left|2\Pr\left[\text{Success}_A^0\right] - 1\right|. \quad (2)$$

$\text{Game}_1$: which simulates that $A$ launches *Execute* and *Test* queries to verify the output according to the transcripts $\{\text{Message1}, \text{Message2}, \text{Message3}, \text{Message4}\}$. Among the transcripts, $\{A_1, \Delta, A_3, T_2\}$ are related to the session key. However, $A$ cannot figure out the relation between them the transcripts and the output of *Test* because of the random numbers. Therefore, we have

$$\Pr\left[\text{Success}_A^1\right] = \Pr\left[\text{Success}_A^0\right]. \quad (3)$$

$\text{Game}_2$: In this game, we simulate $A$ computes the session key $K_{SH}$ through the messages transmitted openly. $K_{SH} = h(A_1, A_2^*, A_3, A_4, id_j^*, T_2)$, which is based on CDHP. The advantage of calculating $K_{SH}$ by $A$ is $\text{Adv}_A^{\text{CDHP}}$. Therefore, we have

$$\Pr\left[\text{Success}_A^2\right] - \Pr\left[\text{Success}_A^1\right] = \text{Adv}_A^{\text{CDHP}}. \quad (4)$$

Table 2: Security properties comparison.

| Attacks/Properties | [14] | [25] | [29] | [30] | Ours |
|---|---|---|---|---|---|
| Anonymity | Yes | Yes | No | No | Yes |
| Mutual authentication | Yes | Yes | Yes | Yes | Yes |
| Forger and impersonation attack | No | Yes | Yes | Yes | Yes |
| Off-line identity guessing attack | Yes | Yes | No | No | Yes |
| Sensor node capture attack | Yes | Yes | Yes | Yes | Yes |
| Smart card loss attack | Yes | Yes | Yes | Yes | Yes |
| Desynchronization attack | Yes | No | Yes | No | Yes |
| Stolen-verifier attack | Yes | Yes | Yes | No | Yes |
| Man-in-middle attack | Yes | Yes | Yes | Yes | Yes |
| Replay attack | Yes | Yes | No | Yes | Yes |
| Know-key attack | Yes | Yes | No | No | Yes |
| Untraceability | Yes | Yes | Yes | Yes | Yes |
| Perfect forward secrecy | No | No | No | No | Yes |

$\text{Game}_3$: This game simulates $A$ performs $\mathit{Corrupt}(\text{INS}_{SN}^i)$ to acquire the reserved information $\{id_j, x_{Nj}, y_{Nj}, ST_j, Cha_j, T_j\}$ in $SN_j$ and try to calculate $\Delta^* = h(x_{Nj}^{\text{new}*}, y_{Nj}^{\text{new}*}, K_{SH}, T_2)$ to testify the $K_{SH}$'s correctness, where $x_{Nj}^{\text{new}*} = \mu \oplus h(A_2, MH_j, T_2)$, $y_{Nj}^{\text{new}*} = \lambda \oplus h(T_2, A_2, MH_j)$, and $MH_j = h(PUF(\text{Cha}_j)) \oplus ST_j$. $A$ has to break PUF to obtain $MH_j$. The probability of breaking PUF is $Adv_A^{\text{PUF}}$. Therefore, we have

$$\Pr\left[\text{Success}_A^3\right] - \Pr\left[\text{Success}_A^2\right] \le \text{Adv}_A^{\text{PUF}}. \quad (5)$$

$\text{Game}_4$: which simulates *Execute* and *Send* queries are executed by $A$ to launch the collision attacks. In line with the birthday paradox's definition, the possibility of a hash collision is $q_H^2/2^{l_h+1}$. Meanwhile, the collision probability of other transcripts is $(q_S + q_E)^2/2n$. Hence, we have

$$\Pr\left[\text{Success}_A^4\right] - \Pr\left[\text{Success}_A^3\right] \le \frac{(q_S + q_E)^2}{2n} + \frac{q_H^2}{2^{l_h+1}}. \quad (6)$$

The random bit $r \in (0, 1)$, the probability of guessing $r$ is 1/2, which is equal to guessing the session key. That is,

$$\Pr\left[\text{Success}_A^4\right] = \frac{1}{2}. \quad (7)$$

Combining (1) with (6), we got

$$\frac{1}{2}\text{Adv}_A^{\mathscr{P}} \le \frac{(q_S + q_E)^2}{2n} + \frac{q_H^2}{2^{l_h+1}} + \text{Adv}_A^{\text{CDHP}} + \text{Adv}_A^{\text{PUF}}. \quad (8)$$

(8) can be expressed as follows:

$$\text{Adv}_A^{\mathscr{P}} \le \frac{(q_S + q_E)^2}{n} + \frac{q_H^2}{2^{l_h}} + 2\text{Adv}_A^{\text{CDHP}} + 2\text{Adv}_A^{\text{PUF}}. \quad (9)$$

□

## 7. Performance Analysis

We study and compare security and performance efficiency between ours with others. According to the comparison of the security attributes which are given in Table 2, we earn better security. In Windows 10 professional 64-bit, Intel(R)

Table 3: The computation cost comparison.

| Schemes | Server | $SN_j$ (sensor) | Total |
|---|---|---|---|
| [14] | $5T_{HS}$ | $3T_{HS}$ | $8T_{HS}$ (0.544*ms*) |
| [25] | $3T_{HS} + 3T_{EA} + 2T_{SE}$ | $2T_{HS} + 2T_{EA} + T_{SE}$ | $5T_{HS} + 5T_{EA} + 3T_{SE}$ (14.525*ms*) |
| [29] | $6T_{HS}$ | $4T_{HS}$ | $10T_{HS}$ (0.680*ms*) |
| [30] | $6T_{HS}$ | $4T_{HS}$ | $10T_{HS}$ (0.680*ms*) |
| Ours | $5T_{HS} + 3T_{EA}$ | $13T_{HS} + 3T_{EA}$ | $18T_{HS} + 6T_{EA}$ (16.230*ms*) |

Table 4: The storage cost comparison.

| Protocols | | Storage cost (bits) | Total (bits) |
|---|---|---|---|
| | Sensor | 544 | |
| [14] | RN | 32 | 864 |
| | Server | 288 | |
| | Sensor | 1536 | |
| [25] | RN | 0 | 1952 |
| | Server | 416 | |
| | Sensor | 800 | |
| [29] | RN | 32 | 1108 |
| | Server | 276 | |
| | Sensor | 1056 | |
| [30] | RN | 32 | 1664 |
| | Server | 576 | |
| | Sensor | 832 | |
| Ours | RN | 32 | 928 |
| | Server | 64 | |

Table 5: The communication cost comparison.

| Schemes | [14] | [25] | [29] | [30] | Ours |
|---|---|---|---|---|---|
| Communication cost (bits) | 4196 | 2752 | 3712 | 3712 | 3936 |

Core(TM) i5-4590, we earn $T_{HS} = 0.068$ms (millisecond), $T_{EA} = 2.501$ms, $T_{SE} = 0.56$ms [36], where $T_{HS}$ is hash operation, $T_{EA}$ represents ECC operation, and $T_{SE}$ is symmetric key encryption. As Table 3 revealed, we describe the computational cost comparison between other protocols and the proposed protocol. In [14], the server's and sensor's total computation cost is $5T_{HS} + 3T_{HS} = 8T_{HS}$ (0.544ms). Accordingly, the schemes [29, 30] both need $6T_{HS} + 4T_{HS} = 10T_{HS}$ (0.544ms), and scheme [25] needs $5T_{HS} + 5T_{EA} + 3T_{SE}$ (14.525ms), and ours is $18T_{HS} + 6T_{EA}$ (16.230ms). Because our protocol is safer than others and achieves perfect forward secrecy, so ours achieve both high computational efficiency and security.

According to [38], outputs of identity, timestamp, and password are 32 bits, and a random integer, hash function, or block encryption is 256 bits, and a point in the elliptic curve is 160 bits. We calculate the storage overhead of the devices participating in authentication. Storage costs comparison is indicated in Table 4, ours maintain the lowest storage overhead. In addition, messages in login and mutual authentication are transmitted 4 times in our scheme. We calculate our communication costs and others, and ours is equivalent to other schemes from Table 5.

## 8. Conclusion

We first point out that Alzahrani et al.'s protocol can't defend against stolen-verifier attacks, desynchronization attacks, known-key attacks, and off-line identity guessing attacks and has no perfect forward secrecy. After that, we design a patient monitoring scheme based on ECC for WBAN in IoHT. We use verification tool ProVerif and formal security proof to demonstrate the security of our scheme. Through comparative analysis, our protocol is safer and more efficient to suit the lightweight and secrecy in medical scenarios. In the future, we will research more pragmatic and anonymous authentication protocol for more complex WBAN scenarios.

## Data Availability

All data are included in manuscript.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. U21A20466) and the National Key R&D Program of China (Grant no. 2017YFB0802000).

## References

[1] M. Seyedi, B. Kibret, D. T. Lai, and M. Faulkner, "A survey on intrabody communications for body area network applications," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2067–2079, 2013.

[2] V. Esteves, A. Antonopoulos, E. Kartsakli, M. Puig-Vidal, P. Miribel-Català, and C. Verikoukis, "Cooperative energy harvesting-adaptive MAC protocol for WBANs," *Sensors*, vol. 15, no. 6, Article ID 12635, 2015.

[3] R. Punj and R. Kumar, "Technological aspects of WBANs for health monitoring: a comprehensive review," *Wireless Networks*, vol. 25, no. 3, pp. 1125–1157, 2019.

[4] B. Narwal and A. K. Mohapatra, "A review on authentication protocols in wireless body area networks (WBAN)," in *Proceedings of the 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 227–232, IEEE, Gurgaon, India, October 2018.

[5] M. Salayma, A. Al-Dubai, I. Romdhani, and Y. Nasser, "Wireless body area network (WBAN) a survey on reliability, fault tolerance, and technologies coexistence," *ACM Computing Surveys*, vol. 50, no. 1, pp. 1–38, 2018.

[6] T. Limbasiya and N. Doshi, "An analytical study of biometric based remote user authentication schemes using smart cards," *Computers & Electrical Engineering*, vol. 59, pp. 305–321, 2017.

[7] Q. Liu, K. G. Mkongwa, and C. Zhang, "Performance issues in wireless body area networks for the healthcare application: a survey and future prospects," *SN Applied Sciences*, vol. 3, no. 2, Article ID 155, pp. 1–19, 2021.

[8] H. M. Chen, J. W. Lo, and C. K. Yeh, "An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3907–3915, 2012.

[9] Q. Xie, J. Zhang, and N. Dong, "Robust anonymous authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 37, no. 2, pp. 9911–9918, 2013.

[10] Q. Xie, B. Hu, and N. Dong, "Anonymous three-party password-authenticated key exchange scheme for telecare medical information systems," *PLoS One*, vol. 9, no. 7, Article ID e102747, 2014.

[11] N. Radhakrishnan and A. P. Muniyandi, "Dependable and provable secure two-factor mutual authentication scheme using ECC for IoT-based telecare medical information system," *Journal of Healthcare Engineering*, vol. 2022, Article ID 9273662, 2022.

[12] C. Wang and Y. Zhang, "New authentication scheme for wireless body area networks using the bilinear pairing," *Journal of Medical Systems*, vol. 39, no. 11, pp. 136–138, 2015.

[13] Q. Jiang, X. Lian, C. Yang, J. Ma, Y. Tian, and Y. Yang, "A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth," *Journal of Medical Systems*, vol. 40, no. 11, Article ID. 231, 2016.

[14] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K. K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Computer Networks*, vol. 129, pp. 429–443, 2017.

[15] A. M. Koya and P. P. Deepthi, "Anonymous hybrid mutual authentication and key agreement scheme for wireless body area network," *Computer Networks*, vol. 140, pp. 138–151, 2018.

[16] M. Soni and D. K. Singh, "LAKA: lightweight authentication and key agreement protocol for internet of things based wireless body area network," *Wireless Personal Communications*, vol. 127, no. 2, pp. 1067–1084, 2021.

[17] S. U. Jan, S. Ali, I. A. Abbasi, A. Alsanad, and H. Khattak, "Secure patient authentication framework in the healthcare system using wireless medical sensor networks," *Journal of Healthcare Engineering*, vol. 2021, Article ID 9954089, 2021.

[18] I. Ullah, S. Zeadally, N. U. Amin, M. Asghar Khan, and H. Khattak, "Lightweight and provable secure cross-domain access control scheme for internet of things (IoT) based wireless body area networks (WBAN)," *Microprocessors and Microsystems*, vol. 81, Article ID 103477, 2021.

[19] I. Ullah, M. A. Khan, A. Alkhalifah et al., "A multi-message multi-receiver signcryption scheme with edge computing for secure and reliable wireless internet of medical things communications," *Sustainability*, vol. 13, no. 23, Article ID 13184, 2021.

[20] I. Ullah, A. Alkhalifah, S. U. Rehman, N. Kumar, and M. A. Khan, "An anonymous certificateless signcryption scheme for internet of health things," *IEEE Access*, vol. 9, Article ID 101207, 2021.

[21] I. Ullah, N. U. Amin, M. A. Khan, H. Khattak, and S. Kumari, "An efficient and provable secure certificate-based combined signature, encryption and signcryption scheme for internet of things (IoT) in mobile health (M-Health) system," *Journal of Medical Systems*, vol. 45, no. 1, p. 4 2021.

[22] M. A. Khan, S. U. Rehman, M. I. Uddin et al., "An online-offline certificateless signature scheme for internet of health things," *Journal of Healthcare Engineering*, vol. 2020, Article ID 6654063, 10 pages, 2020.

[23] L. Wu, Y. Zhang, L. Li, and J. Shen, "Efficient and anonymous authentication scheme for wireless body area networks," *Journal of Medical Systems*, vol. 40, no. 6, p. 134 2016.

[24] R. Chen and D. Peng, "Analysis and improvement of a mutual authentication scheme for wireless body area networks," *Journal of Medical Systems*, vol. 43, no. 2, pp. 19–10, 2019.

[25] X. Li, J. Peng, S. Kumari, F. Wu, M. Karuppiah, and K. K. Raymond Choo, "An enhanced 1-round authentication protocol for wireless body area networks with user anonymity," *Computers & Electrical Engineering*, vol. 61, pp. 238–249, 2017.

[26] K. Sowjanya, M. Dasgupta, and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems," *International Journal of Information Security*, vol. 19, no. 1, pp. 129–146, 2020.

[27] S. Kalra and S. K. Sood, "Advanced password based authentication scheme for wireless sensor networks," *Journal of Information Security and Applications*, vol. 20, pp. 37–46, 2015.

[28] C. Chunka, S. Banerjee, and R. S. Goswami, "An efficient user authentication and session key agreement in wireless sensor network using smart card," *Wireless Personal Communications*, vol. 117, no. 2, pp. 1361–1385, 2021.

[29] Z. Xu, C. Xu, H. Chen, and F. Yang, "A lightweight anonymous mutual authentication and key agreement scheme for WBAN," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 14, Article ID e5295, 2019.

[30] B. A. Alzahrani, A. Irshad, A. Albeshri, and K. Alsubhi, "A provably secure and lightweight patient-healthcare authentication protocol in wireless body area networks," *Wireless Personal Communications*, vol. 117, no. 1, pp. 47–69, 2021.

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

[32] B. Hu, W. Tang, and Q. Xie, "A two-factor security authentication scheme for wireless sensor networks in IoT environments," *Neurocomputing*, vol. 500, pp. 741–749, 2022.

[33] T. Jabeen, H. Ashraf, and A. Ullah, "A survey on healthcare data security in wireless body area networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9841–9854, 2021.

[34] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[35] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: a tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[36] M. Potkonjak and V. Goudar, “Public physical unclonable functions,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1142–1156, 2014.
[37] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, *ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, pp. 05–16, 2018.
[38] Q. Xie, G. Wang, X. Tan, K. Chen, and L. Fang, “Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.