Hindawi
Security and Communication Networks
Volume 2024, Article ID 9805893, 1 page
https://doi.org/10.1155/2024/9805893

*Retraction*

# Retracted: Fog-Driven Secure Authentication and Key Exchange Scheme for Wearable Health Monitoring System

**Security and Communication Networks**

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

## References

[1] T.-Y. Wu, L. Yang, Q. Meng, X. Guo, and C.-M. Chen, "Fog-Driven Secure Authentication and Key Exchange Scheme for Wearable Health Monitoring System," *Security and Communication Networks*, vol. 2021, Article ID 8368646, 14 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8368646, 14 pages
https://doi.org/10.1155/2021/8368646

*Research Article*

# Fog-Driven Secure Authentication and Key Exchange Scheme for Wearable Health Monitoring System

**Tsu-Yang Wu, Lei Yang, Qian Meng, Xinglan Guo, and Chien-Ming Chen**

*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*

Correspondence should be addressed to Chien-Ming Chen; chienmingchen@ieee.org

Received 10 August 2021; Revised 13 September 2021; Accepted 17 September 2021; Published 6 October 2021

Academic Editor: Azees M

Smart wearable devices, as a popular mobile device, have a broad market. Smart wearable medical devices implemented in wearable health monitoring systems can monitor the data pertaining to a patient's body and let the patient know their own physical condition. In addition, these data can be stored, analyzed, and processed in the cloud to effectively prevent diseases. As an Internet-of-things technology, fog computing can process, store, and control data around devices in real time. However, the distributed attributes of fog nodes make the monitored body data and medical reports at risk of privacy disclosure. In this paper, we propose a fog-driven secure authentication and key exchange scheme for wearable health monitoring systems. Furthermore, we conduct a formal analysis using the Real-Oracle-Random model, Burrows–Abadi–Needham logic, and ProVerif tools and an informal analysis to perform security verification. Finally, a performance comparison with other related schemes shows that the proposed scheme has the best advantages in terms of security, computing overhead, and communication cost.

## 1. Introduction

The Internet of things (IoT) [1, 2] refers to the communication, transmission, analysis, and control between things through the Internet. In other words, the IoT is an expansion and extension of the Internet, providing various devices with the ability to communicate. Smart mobile devices, as popular IoT devices, have entered the stage of commercialization, and their development is relatively mature. Smart mobile devices, such as smart watches, smart glasses, and smart helmets, have been widely used in the fields of medical health and reasonable sports. Owing to the rapid development of mobile medical platforms and increasing attention to physical health, smart wearable medical devices (SWMDs) have a broad market in the field of artificial intelligence [3–6]. In addition, SWMDs have the advantages of simple operation, reduced treatment costs, and prevention of diseases. As a specific application of SWMDs, wearable health monitoring systems are of great significance to both doctors and patients. Patients can evaluate their health in real time without visiting a hospital. SWMDs can monitor blood pressure, heart rate, sleep status, and other indicators. Patients with hypertension, coronary heart disease, and other chronic diseases need not visit a hospital frequently for examinations, thereby saving a significant amount of time and reducing the cost of diagnosis. Doctors can provide timely feedback on the health status based on the SWMDs worn by patients. Furthermore, using the information uploaded by SWMDs, doctors can better understand the data pertaining to a patient's body data to obtain more accurate diagnosis results. From the perspective of medical resources, the application of wearable health monitoring systems reduces the number of patients seeking medical treatment and alleviates problems regarding the lack of hospital beds.

As a relatively mature IoT technology, fog computing can extend cloud services to the edge of a network. The principle of fog computing and cloud computing is to upload data for analysis, storage, and processing. The difference is that cloud computing uploads all data to the same center, and fog computing disperses the data to many central nodes. When the data load is too large, cloud computing cannot meet the application requirements of high mobility and low latency. For example, SWMDs are placed far from the cloud server, and transmission delays occur when patients need a real-time diagnosis. As an extension of cloud computing, fog computing can process, store, and control data around devices in real time. Fog nodes are deployed between the cloud and SWMDs, and these are located at a low position in the network topology and have less network delay. Figure 1 shows the typical structure of a fog-based wearable health monitoring system.

In this structure, SWMDs and fog nodes need to register with the cloud server to obtain a legal identity before being used. Fog nodes are deployed between the cloud server and the users of SWMDs. These users send the data pertaining to their body to the fog node through the communication protocol. After filtering and aggregating the received information, fog nodes send it to the cloud server through a wireless network. The cloud server analyzes and stores the received body data and then returns the diagnosis results in real time through the fog node.

### 1.1. Related Work.

*1.1. Related Work.* Wearable health monitoring systems have significant practical value in medical health monitoring. SWMDs can monitor the basic information and health data of patients and transmit these data to medical staff. During the transmission process, if the health data or diagnostic records are intercepted or tampered with by an adversary, then the lives of patients can be directly impacted. Many authentication and key agreement (AKA) protocols for SWMDs have been proposed. In 2008, Venkatasubramanian et al. [7] designed an AKA scheme based on electrocardiogram (ECG) data transmission for patients with heart diseases in body sensor networks. In 2009, Sriram et al. [8] used a wearable ECG sensor to monitor biometric ECGs for verifying the identity of patients in remote health monitoring. Venkatasubramanian et al. [9] proposed an AKA scheme based on the physiological signal in the body area network, which can realize secure communication between sensors without initialization or pre-deployment. In 2013, Hu et al. [10] proposed an AKA scheme based on ordered physiological features in wireless body area networks. This scheme does not require initialization or the pre-deployment phase and can calculate the biological characteristics according to the physiological signals of different parts of the human body. In 2017, Masdari et al. [11] reported that the scheme proposed in [7] has a high time complexity and low security. During the process of message transmission, the scheme in [10] has a lower energy consumption and smaller storage space than the scheme in [9], but they have similar efficiency and time variance in generating keys.

SWMDs are the key applications of IoT technology, in which identity AKA is of great significance in protecting the security of health data. Therefore, privacy protection [12–16] has become an important security attribute of the protocols proposed by researchers. In 2017, to ensure anonymity and low energy consumption, Zhang et al. [17] designed an AKA scheme based on dynamic authentication and three factors for an e-health system. In the same year, Li et al. [18] designed a lightweight, centralized, and two-hop anonymous AKA scheme for wireless body area networks. In 2018, Chen et al. [19] showed that the scheme proposed in [18] is vulnerable to offline identity guessing attacks, sensor node impersonation attacks, and hub node spoofing attacks. Subsequently, they improved Li et al.'s scheme. Koya and Deepthi [20] found that the scheme in [18] is vulnerable to sensor node impersonation attacks and that the assumption that hub nodes are trustworthy is not feasible. Therefore, they provided an anonymous two-way AKA scheme for wireless body area networks. In 2019, Kompara et al. [21] reported that the scheme in [18] does not provide untraceability for sensor nodes, and thus they proposed a robust and efficient AKA scheme with untraceability in wireless body area networks. In the same year, Aghili et al. [22] found that the scheme in [17] fails to resist user traceability attacks, desynchronization attacks, denial-of-service attacks, and internal attacks. Further, they proposed a new lightweight AKA and ownership transfer scheme for e-health systems in an IoT environment. In 2020, Sowjanya et al. [23] conducted cryptanalysis on the scheme proposed in [18] and found that it cannot support perfect forward security and key control and is vulnerable to desynchronous attacks. To overcome these limitations, an enhanced anonymous AKA protocol [23] in a wearable health monitoring system was proposed.

SWMDs using IoT technologies, such as cloud computing and fog computing, also participate in the AKA process of wearable health monitoring systems. In 2019, Jia et al. [24] proposed a fog-driven AKA scheme for IoT medical systems. In the same year, Wazid et al. [25] designed a secure AKA scheme based on fog computing. In 2020, Chen et al. [26] showed that the scheme in [24] suffers from ephemeral secret leakage attacks and proposed a secure AKA scheme based on fog computing. In 2021, Shamshad et al. [27] reported that the scheme in [24] is vulnerable to impersonation attacks and cannot provide anonymity for users and fog nodes. Wu et al. [28] also reported that the scheme in [24] exhibits security vulnerabilities, such as known session-specific temporary information attacks and a lack of preverification. Thus, they proposed an improved fog-driven AKA scheme for IoT medical systems. In the same year, Ali et al. [29] analyzed and determined that the scheme in [25] is vulnerable to traceability and clogging attacks. Therefore, they proposed an anti-clogging AKA scheme based on fog computing. Some important related works are summarized in Table 1.

*1.2. Our Contribution.* According to the earlier analysis, medical health monitoring systems based on fog computing need further improvement. We propose a fog-driven secure authentication and key exchange scheme for wearable health monitoring systems to ensure the security and privacy of the monitoring information and diagnostic reports of SWMDs.

(1) Our scheme can provide user device anonymity, fog node anonymity, and perfect forward security and resist replay attacks, impersonation attacks, known session-specific temporary information attacks, and insider attacks.

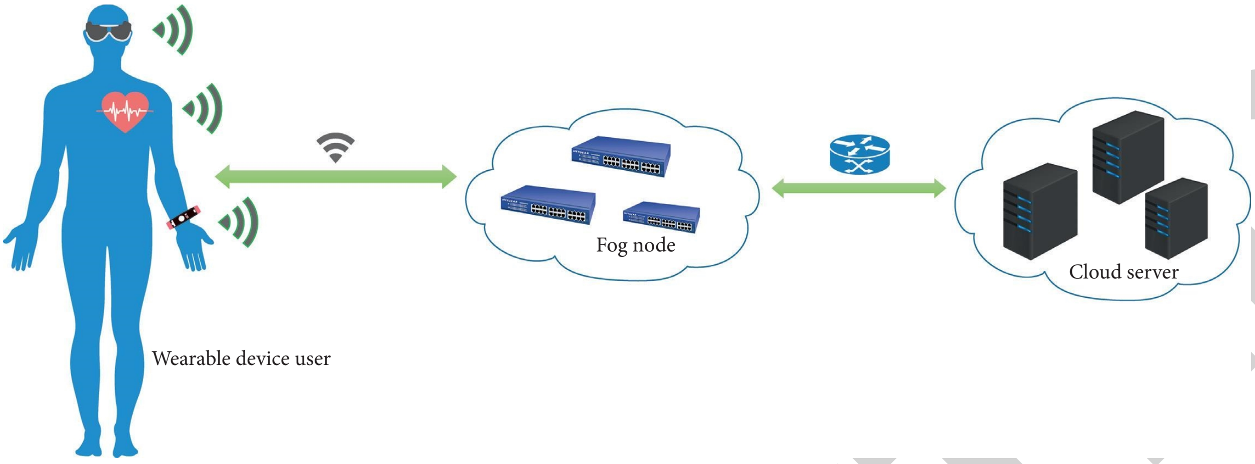


FIGURE 1: A typical structure of a fog-based wearable health monitoring system.

TABLE 1: Cryptographic techniques and limitations.

| Scheme | Cryptographic techniques | Limitations |
|---|---|---|
| Zhang et al. [12] | (i) Utilized one-way hash function<br>(ii) Dynamic string generating algorithm | (i) Does not resist user traceability attacks<br>(ii) Does not resist desynchronization attacks<br>(iii) Does not resist denial-of-service attacks<br>(iv) Does not resist internal attacks |
| Li et al. [13] | (i) Utilized one-way hash function | (i) Does not resist offline identity guessing attacks<br>(ii) Does not resist sensor node impersonation attacks<br>(iii) Does not resist hub node spoofing attacks<br>(iv) Does not provide untraceability<br>(v) Does not resist desynchronization attacks<br>(vi) Does not provide perfect forward security<br>(vii) Does not provide key control |
| Jia et al. [19] | (i) Utilized one-way hash function<br>(ii) Based on bilinear pairings<br>(iii) Based on Diffie–Hellman problem | (i) Does not resist ephemeral secret leakage attacks<br>(ii) Does not resist impersonation attacks<br>(iii) Does not resist known session-specific temporary information attacks<br>(iv) Does not provide anonymity<br>(v) Does not provide pre-verification |
| Wazid et al. [20] | (i) Utilized one-way hash function<br>(ii) Utilized ECC | (i) Does not provide traceability<br>(ii) Does not resist clogging attacks |

(2) Using the Real-Oracle-Random (ROR) model, we provide the probability of breaking the symmetric encryption and decryption algorithms and prove that our protocol has a secure authentication process and session key. By using the Burrows–Abadi–Needham (BAN) logic, ProVerif tools, and an informal analysis, we prove that the security of the proposed protocol can resist all known attacks.

(3) The proposed protocol and five related protocols are analyzed for performance evaluation. We find that the proposed protocol has advantages in terms of security, computing overhead, and communication cost.

*1.3. Paper Organization.* The remainder of this paper is organized as follows. Section 2 describes the proposed security scheme in detail. Section 3 presents the verification of the security of the proposed scheme, including a formal analysis using the Real-Oracle-Random (ROR) model, BAN logic, and ProVerif tool and an informal analysis. In Section 4, the performance of the proposed scheme is analyzed and compared with those of five related schemes. The conclusions are presented in Section 5.

## 2. Proposed Scheme

The proposed scheme involves three entities: wearable device ($W_i$), fog node ($F_j$), and cloud server ($CS$). The entire scheme consists of four phases: initialization, SWMD registration, fog node registration, and AKA. The symbols used are listed in Table 2.

*2.1. Initialization.* $CS$ completes the initialization of the functions and defines the required parameters involved in the scheme. $CS$ chooses its own secret key, $s$, and defines the one-way hash function, $h(\cdot)$, and the symmetric encryption and decryption function, $E_k(\cdot)/D_k(\cdot)$. Then, $CS$ publishes $\{h(\cdot), E_k(\cdot)/D_k(\cdot)\}$.

Table 2: Symbols and descriptions.

| Symbol | Description |
|---|---|
| $W_i$ | Smart wearable medical device |
| $F_j$ | Fog node |
| $CS$ | Cloud server |
| $\mathscr{A}$ | Adversary |
| $s$ | Private key of $CS$ |
| $E_k(\cdot)$ | Symmetric encryption function |
| $D_k(\cdot)$ | Symmetric decryption function |
| $h(\cdot)$ | Hash function |
| $\|$ | Connect operation |
| $\oplus$ | Exclusive or operation |

*2.2. SWMD Registration Phase.* SWMDs worn by users must be registered with the cloud server before being used. $W_i$ inputs identity $ID_W$ and password $PW_W$, generates a random number, $r_W$, and calculates $RPW_W = h(ID_W\|PW_W\|r_W)$. $W_i$ sends $\{ID_W, RPW_W\}$ to $CS$. After $CS$ receives the request, it generates a random number, $s_W$, and calculates $RID_W = h(ID_W\|s_W\|s)\oplus h(ID_W\|RPW_W)$. Subsequently, $CS$ stores $\{RID_W, ID_W, s_W\}$ in the database and sends $\{RID_W, s_W\}$ to $W_i$. After receiving the response, $W_i$ calculates $P_W = r_W\oplus h(ID_W\|PW_W\|RID_W)$ and $R_W = h(RPW_W\|P_W s_W\|\|RID_W)$ and stores $\{P_W, R_W, RID_W, s_W\}$ in memory. The wearable device registration phase is shown in Figure 2.

*2.3. Fog Node Registration Phase.* Fog nodes must register with the cloud service before collecting and transmitting user data. $F_j$ inputs identity $ID_F$, generates a random number, $r_F$, and calculates $Q_F = h(ID_F\|r_F)$. $F_j$ sends $\{ID_F, Q_F\}$ to $CS$. After $CS$ receives the request, it generates a random number, $s_F$, and calculates $RID_F = h(s_F\|s)\oplus h(ID_F\|Q_F)$. Subsequently, $CS$ stores $\{RID_F, s_F\}$ in the database and sends $\{RID_F, s_F\}$ to $F_j$. After receiving the response, $F_j$ calculates $P_F = r_F\oplus h(ID_F\|s_F\|RID_F)$ and stores $\{P_F, RID_F, s_F\}$ in memory. The fog node registration phase is shown in Figure 3.

*2.4. Authentication and Key Exchange Phase.* The SWMD regularly uploads the data pertaining to the user's physical condition to the nearby fog node, which pre-processes the data and then sends it to CS. After receiving the user's body data, CS provides timely feedback to the SWMD through the fog node. The details are as follows.

(1) $W_i$ inputs $ID_W$ and$PW_W$, calculates $r_W = P_W\oplus h(ID_W\|PW_W\|RID_W)$ and $RPW_W = h(ID_W\|PW_W\|r_W)$, and checks $R_W \overset{?}{=} h(RPW_W\|P_W\|s_W\|RID_W)$. If the equation does not hold, the session is terminated. Otherwise, $W_i$ selects $a$, calculates $V_1 = a\oplus RID_W\oplus h(ID_W\|RPW_W)$ and $V_2 = h(a\|ID_W\|RID_W\|RID_F\|V_1)$, and sends $M_1 = \{V_1, V_2, RID_W\}$ to $F_j$.

(2) After receiving $M_1$, $F_j$ selects $b$ and calculates $r_F = P_F\oplus h(ID_F\|s_F\|RID_F)$, $Q_F = h(ID_F\|r_F)$, $V_3 = b\oplus RID_F\oplus h(ID_F\|Q_F)$, and $V_4 = h(b\|s_F\|RID_W\|RID_F\|V_3)$. Then, $F_j$ sends $M_2 = \{V_1, V_2, RID_W, V_3, V_4, RID_F\}$ to $CS$.

(3) After receiving $M_2$, $CS$ finds the corresponding $\{ID_W, s_W\}$ and $\{s_F\}$ in the database according to $RID_W$ and $RID_F$, respectively. $CS$ calculates $a = V_1\oplus h(ID_W\|s_W\|s)$ and $b = V_3\oplus h(s_F\|s)$ and checks $V_2 \overset{?}{=} h(a\|ID_W\|RID_W\|RID_F\|V_1)$ and $V_4 \overset{?}{=} h(b\|ID_W\|RID_W\|RID_F\|V_3)$. If the equations do not hold, the session is terminated. Otherwise, $CS$ selects $c$, calculates $s'_W = c\oplus h(RID_W\|s_W\|a)$, $s'_F = c\oplus h(RID_F\|s_F\|b)$, and $RID'_W = h(ID_W\|s'_W\|s)$, and updates $\{RID'_W, ID_W, s'_W\}$ and $\{RID_F, s'_F\}$ in the database. Further, $CS$ computes $V_5 = h(RID_W\|ID_W\|s_W)$and $V_6 = h(RID_F\|s_F)$ and encrypts $E_1 = E_{RID_W\oplus h(ID_W\|s_W\|s)}(b, c, s'_W, RID'_W, RID_F, V_5)$and $E_2 = E_{RID_F\oplus h(s_F\|s)}(a, c, s'_F, RID'_W, V_6)$. Finally, $CS$ computes the session key, $SK_c = h(a\|b\|\|cRID'_W\|RID_F)$, and sends $M_3 = \{E_1, E_2\}$ to $F_j$.

(4) After receiving $M_3$, $F_j$ calculates $V'_F = h(RID_F\|s_F)$, decrypts $(a, c, s'_F, RID'_W, V_6 = D_{h(ID_F\|Q_F)}(E_2))$, and checks $V'_6 \overset{?}{=} V_6$. If the equation does not hold, the session is terminated. Otherwise, $F_j$ updates $\{RID_F, s'_F\}$ in the memory, computes $SK_f = h(a\|b\|c\|RID'_W/\text{parallel}\, RID_F)$, and sends $M_4 = \{E_1\}$ to $W_i$.

(5) After receiving $M_4$, $W_i$ calculates $V'_5 = h(RID_W\|ID_W\|s_W)$, decrypts $(b, c, s'_W, RID'_W, RID_F, V_5) = D_{h(ID_W\|RPW_W)}(E_1)$, and checks $V'_5 \overset{?}{=} V_5$. If the equation does not hold, the session is terminated. Otherwise, $W_i$ updates $\{RID'_W, s'_W\}$ in the memory and computes $SK_w = h(a\|b\|c\|RID'_W\|RID_F)$.

$W_i$ and $CS$ complete mutual AKA through $F_j$, and $\{RID_W, s_W, s_F\}$ is updated simultaneously. The authentication and key exchange phase is shown in Figure 4.

## 3. Security Analysis

*3.1. Formal Proof.* In the ROR model [30, 31], some queries are used to verify the security robustness of the proposed scheme. In the scheme, participants $W_i$, $F_j$, and $CS$ generate many communication instances in the process of interaction. For the convenience of proof, we define $\prod_W^x$, $\prod_F^y$, and $\prod_{CS}^z$ as the $x$-th instance of $W_i$, $y$-th instance of $F_j$, and $z$-th instance of $CS$, respectively.

*3.1.1. Queries.* In this model, the queries used by adversary $\mathscr{A}$ are defined as follows.

(1) Execute $(\prod_W^x, \prod_F^y, \prod_{CS}^z)$: the query passively captures the information transmitted by entities in the public channel and outputs message records.

(2) Hash (str): the query inputs string str and outputs the corresponding hash value.

(3) Send $(\prod_W^x, \prod_F^y, \prod_{CS}^z, M)$: the query actively intercepts information transmitted between entities in the public channel and forges them as $M$. Then, $\mathscr{A}$ sends $M$ to $\prod_W^x, \prod_F^y$, or $\prod_{CS}^z$ and receives the corresponding response.

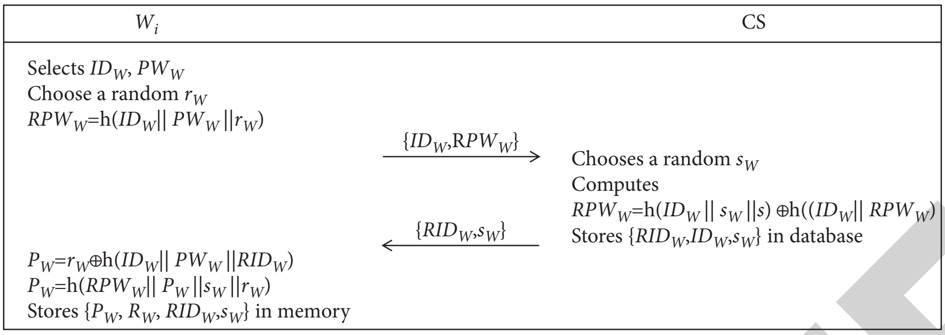

$W_i$
CS
Selects $ID_W$, $PW_W$
Choose a random $r_W$
$RPW_W$=h($ID_W$|| $PW_W$||$r_W$)
{$ID_W$,$RPW_W$}
Chooses a random $s_W$
Computes
$RPW_W$=h($ID_W$|| $s_W$||$s$) ⊕h(($ID_W$|| $RPW_W$)
Stores {$RID_W$,$ID_W$,$s_W$} in database
{$RID_W$,$s_W$}
$P_W$=$r_W$⊕h($ID_W$|| $PW_W$||$RID_W$)
$P_W$=h($RPW_W$|| $P_W$||$s_W$||$r_W$)
Stores {$P_W$, $R_W$, $RID_W$,$s_W$} in memory

Figure 2: SWMD registration phase.

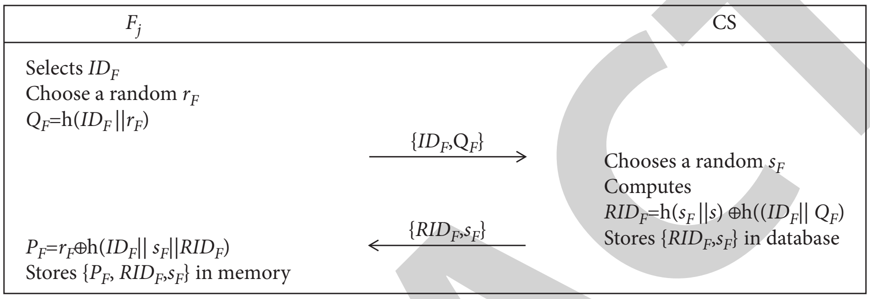

$F_j$
CS
Selects $ID_F$
Choose a random $r_F$
$Q_F$=h($ID_F$||$r_F$)
{$ID_F$,$Q_F$}
Chooses a random $s_F$
Computes
$RID_F$=h($s_F$||$s$) ⊕h(($ID_F$|| $Q_F$)
Stores {$RID_F$,$s_F$} in database
{$RID_F$,$s_F$}
$P_F$=$r_F$⊕h($ID_F$|| $s_F$||$RID_F$)
Stores {$P_F$, $RID_F$,$s_F$} in memory

Figure 3: Fog node registration phase.

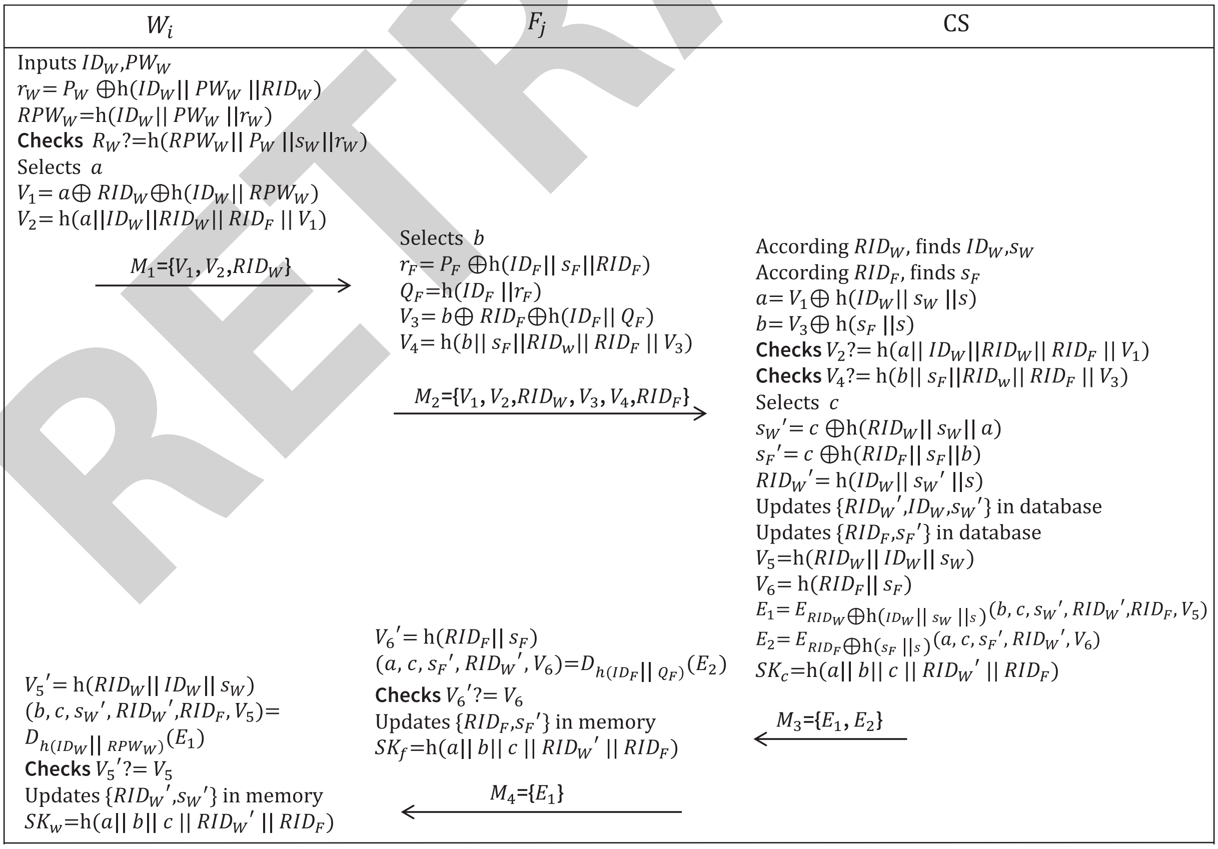

$W_i$
$F_j$
CS
Inputs $ID_W$,$PW_W$
$r_W$= $P_W$ ⊕h($ID_W$|| $PW_W$ ||$RID_W$)
$RPW_W$=h($ID_W$|| $PW_W$ ||$r_W$)
Checks $R_W$?=h($RPW_W$|| $P_W$ ||$s_W$||$r_W$)
Selects $a$
$V_1$= $a$⊕ $RID_W$⊕h($ID_W$|| $RPW_W$)
$V_2$= h($a$||$ID_W$||$RID_W$|| $RID_F$ || $V_1$)
$M_1$={$V_1$, $V_2$,$RID_W$}
Selects $b$
$r_F$= $P_F$ ⊕h($ID_F$|| $s_F$||$RID_F$)
$Q_F$=h($ID_F$ ||$r_F$)
$V_3$= $b$⊕ $RID_F$⊕h($ID_F$|| $Q_F$)
$V_4$= h($b$|| $s_F$||$RID_w$|| $RID_F$ || $V_3$)
$M_2$={$V_1$, $V_2$,$RID_W$, $V_3$, $V_4$,$RID_F$}
According $RID_W$, finds $ID_W$,$s_W$
According $RID_F$, finds $s_F$
$a$= $V_1$⊕ h($ID_W$|| $s_W$ ||$s$)
$b$= $V_3$⊕ h($s_F$ ||$s$)
Checks $V_2$?= h($a$|| $ID_W$||$RID_W$|| $RID_F$ || $V_1$)
Checks $V_4$?= h($b$|| $s_F$||$RID_w$|| $RID_F$ || $V_3$)
Selects $c$
$s_W'$= $c$ ⊕h($RID_W$|| $s_W$|| $a$)
$s_F'$= $c$ ⊕h($RID_F$|| $s_F$||$b$)
$RID_W'$= h($ID_W$|| $s_W'$ ||$s$)
Updates {$RID_W'$,$ID_W$,$s_W'$} in database
Updates {$RID_F$,$s_F'$} in database
$V_5$=h($RID_W$|| $ID_W$|| $s_W$)
$V_6$= h($RID_F$|| $s_F$)
$E_1$= $E_{RID_W \oplus h(ID_W || s_W || s)}(b, c, s_W', RID_W', RID_F, V_5)$
$E_2$= $E_{RID_F \oplus h(s_F || s)}(a, c, s_F', RID_W', V_6)$
$SK_c$=h($a$|| $b$|| $c$ || $RID_W'$ || $RID_F$)
$M_3$={$E_1$, $E_2$}
$V_6'$= h($RID_F$|| $s_F$)
($a$, $c$, $s_F'$, $RID_W'$, $V_6$)=$D_{h(ID_F || Q_F)}(E_2)$
Checks $V_6'$?= $V_6$
Updates {$RID_F$,$s_F'$} in memory
$SK_f$=h($a$|| $b$|| $c$ || $RID_W'$ || $RID_F$)
$M_4$={$E_1$}
$V_5'$= h($RID_W$|| $ID_W$|| $s_W$)
($b$, $c$, $s_W'$, $RID_W'$,$RID_F$, $V_5$)=
$D_{h(ID_W || RPW_W)}(E_1)$
Checks $V_5'$?= $V_5$
Updates {$RID_W'$,$s_W'$} in memory
$SK_w$=h($a$|| $b$|| $c$ || $RID_W'$ || $RID_F$)

Figure 4: Authentication and key exchange phase.

(4) Corrupt$(\prod_W^x, \prod_F^y, \prod_{CS}^z)$: the query can capture a private value in an entity, such as the private key of $CS$ or a random number.

(5) Test$(\prod_W^x, \prod_F^y, \prod_{CS}^z)$: in this model, coin $o$ is tossed randomly. If $o = 1$, the correct session key is returned; otherwise, a random string of the same length as the session key is returned.

*3.1.2. Definitions. Symmetric Encryption and Decryption Algorithm (Ω).* Here, we specify the security key in the symmetric encryption and decryption algorithm as $k$, which includes $k_1$, $k_2$, …, $k_n$. Each key corresponds to an independent encryption oracle: $E_{k_1}$, $E_{k_2}$, …, $E_{k_n}$. Then, in polynomial time $\xi$, the advantage that $\mathscr{A}$ can break $k$ is $\text{Adv}_{\mathscr{A}}^{\Omega,k}(\xi) = |2\Pr[\mathscr{A}\leftarrow E_{k_1}; (b_0, b_1)\leftarrow\mathscr{A}; \alpha\leftarrow 0, 1; \beta\leftarrow E_{k_1}(b_\alpha): \mathscr{A}(\beta) = \alpha] - 1|$. For a sufficiently small number, $\gamma$, we have $\text{Adv}_{\mathscr{A}}^{\Omega,k}(\xi) < \gamma$.

*3.1.3. Theorem.* $\mathscr{A}$ has the ability to operate Execute, Hash, Send, Corrupt, and Test queries. Then, in polynomial time $\xi$, the advantage that $\mathscr{A}$ can break the proposed scheme, $S$, is $\text{Adv}_{\mathscr{A}}^{S}(\xi) \le q_{\text{hash}}^2/2^{l-1} + q_{\text{send}}/2^{l-1} + 2C' \cdot q_{\text{send}}^{s'} + 2\text{Adv}_{\mathscr{A}}^{\Omega,k}(\xi)$, where $q_{\text{hash}}$ and $q_{\text{send}}$ are the times of Hash and Send queries, respectively, $l$ is the length of the hash value, and $C'$ and $s'$ are constants.

*Proof.* The game sequence, $GM_0 - GM_6$, is defined to verify the security robustness of $S$. Here, $\text{Succ}_{\mathscr{A}}^{GM_n}(\xi)$ is the event that $\mathscr{A}$ wins in $GM_n$. The proof is as follows.

$GM_0$: in this round of the game, $\mathscr{A}$ simulates a real attack and does not launch any query. We derive that

$$\text{Adv}_{\mathscr{A}}^{S}(\xi) = \left|2\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - 1\right|. \tag{1}$$

$GM_1$: in this round of the game, $\mathscr{A}$ launches an Execute query. Because of the properties of the query itself, $\mathscr{A}$ only passively receives messages $M_1 = \{V_1, V_2, RID_W\}$, $M_2 = \{V_1, V_2, RID_W, V_3, V_4, RID_F\}$, $M_3 = \{E_1, E_2\}$, and $M_4 = \{E_1\}$. Thus, we have

$$\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] = \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right]. \tag{2}$$

$GM_2$: in this round of the game, $\mathscr{A}$ launches a Hash query. According to the birthday paradox, the probability of a hash conflict occurring in a query is

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right]\right| \le \frac{q_{\text{hash}}^2}{2^{l+1}}, \tag{3}$$

where $l$ is the length of a hash value.

$GM_3$: in this round of the game, $\mathscr{A}$ launches a Send query. According to Zipf's law [32], the probability of a transmission text collision in the query is

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_2}(\xi)\right]\right| \le \frac{q_{\text{send}}}{2^l}. \tag{4}$$

$GM_4$: in this round of the game, $\mathscr{A}$ attempts to make offline password-guessing attacks. $\mathscr{A}$ launches a Corrupt$(\prod_W^x)$ query to obtain parameters $\{P_W, R_W, RID_W, s_W\}$ in the memory of the wearable device, where $P_W = r_W \oplus h(ID_W \| PW_W \| RID_W)$, $R_W = h(RPW_W \| P_W \| \| s_W RID_W)$, and $RID_W = h(ID_W \| s_W \| s) \oplus h(ID_W \| RPW_W)$. In this calculation process, because $r_W$ and $RPW_W$ are unknown, $\mathscr{A}$ cannot calculate identity $ID_W$ and password $PW_W$. According to Zipf's law [32], it can be deduced that

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_3}(\xi)\right]\right| \le C' \cdot q_{\text{send}}^{s'}, \tag{5}$$

where $C'$ and $s'$ are constants.

$GM_5$: the purpose of this game round is to verify the security of the session key. We divide it into the following two cases.

(1) *Perfect Forward Security.* $\mathscr{A}$ launches a Corrupt$(\prod_{CS}^z)$ query to obtain the private key, $s$, of CS.

(2) *Known Session-Specific Temporary Information Attacks.* $\mathscr{A}$ launches a Corrupt$(\prod_W^x)$, Corrupt$(\prod_F^y)$, or Corrupt$(\prod_{CS}^z)$ query to obtain one of the random numbers.

The session key of $S$ is $SK = h(a\|b\|c\|RID'_W\|RID_F)$. In the first case, $\mathscr{A}$ knows $s$ and cannot calculate parameters $\{a, b, c, RID'_W\}$ needed in the session key. In the second case, based on the assumption that $\mathscr{A}$ obtains random number $c$, $\{a, b, RID'_W\}$ cannot be calculated. The same is true for $a$ or $b$. To summarize, if $\mathscr{A}$ wants to calculate the session key, it must decrypt symmetrically on $E_1$ or $E_2$, that is,

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_4}(\xi)\right]\right| \le \text{Adv}_{\mathscr{A}}^{\Omega,k}(\xi). \tag{6}$$

$GM_6$: in this round of game, $\mathscr{A}$ attempts to make impersonation attacks. $\mathscr{A}$ launches a $h(a\|b\|c\|RID'_W\|RID_F)$ query, and the probability of successfully guessing the key is

$$\left|\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] - \Pr\left[\text{Succ}_{\mathscr{A}}^{GM_5}(\xi)\right]\right| \le \frac{q_{\text{hash}}^2}{2^{l+1}}. \tag{7}$$

Because the probability of $\mathscr{A}$ guessing the key correctly and incorrectly is equal, we have

$$\Pr\left[\text{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right] = \frac{1}{2}. \tag{8}$$

According to formulas (1)–(8), we have

$$\begin{aligned}\frac{1}{2}\mathrm{Adv}_{\mathscr{A}}^{S}(\xi) &= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \frac{1}{2}\right| \\ &= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_0}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right]\right| \\ &= \left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_1}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_6}(\xi)\right]\right| \leq \sum_{i=0}^{5}\left|\Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_{i+1}}(\xi)\right] - \Pr\left[\mathrm{Succ}_{\mathscr{A}}^{GM_i}(\xi)\right]\right| \\ &= \frac{q_{\mathrm{hash}}^{2}}{2^{l}} + \frac{q_{\mathrm{send}}}{2^{l}} + C\prime \cdot q_{\mathrm{send}}^{\prime s} + \mathrm{Adv}_{\mathscr{A}}^{\Omega,k}(\xi).\end{aligned} \tag{9}$$

Further derivation yields the result as $\mathrm{Adv}_{\mathscr{A}}^{S}(\xi) \leq (q_{\mathrm{hash}}^{2}/2^{l-1}) + (q_{\mathrm{send}}/2^{l-1}) + 2C' \cdot q_{\mathrm{send}}^{\prime s} + 2\mathrm{Adv}_{\mathscr{A}}^{\Omega,k}(\xi)$. □

*3.2. BAN Logic.* BAN logic [33, 34] is often used to describe and prove the logic and correctness of cryptographic protocols. Before describing the logical reasoning of BAN, we define the symbols and idealize the interactive information. Furthermore, based on the concrete proof, the initial condition assumptions are made, and the set goals are finally obtained by reasoning.

*3.2.1. Rules*

(1) Message-meaning (M-M) rule: $P| \equiv P \longrightarrow^{K} Q$, $P\{X\}_K/P| \equiv Q| \sim X$, $P| \equiv P \overset{Y}{\rightleftharpoons} YQ, P \triangleleft \langle X \rangle_Y/P| \equiv Q| \sim X$.

(2) Nonce-verification (N-V) rule: $P| \equiv \#(X)$, $P| \equiv Q| \sim X/P| \equiv Q| \equiv X$.

(3) Jurisdiction rule: $P| \equiv Q| \Rightarrow X, P| \equiv Q| \equiv X/P| \equiv X$.

(4) Session key (S-K) rule: $P| \equiv \#(X)$, $P \equiv Q \mid \equiv X/P| \equiv P \overset{K}{\leftrightarrow} Q$.

(5) Freshness rule: $P| \equiv \#(X)/P| \equiv \#(X, Y)$.

*3.2.2. Goals*

(1) $G_1$: $W_i| \equiv W_i \longrightarrow^{S} K\ F_j$.

(2) $G_2$: $F_j| \equiv W_i \longrightarrow^{S} K\ F_j$.

(3) $G_3$: $CS| \equiv W_i \longrightarrow^{S} K\ F_j$.

(4) $G_4$: $W_i| \equiv F_j| \equiv W_i \longrightarrow^{S} K\ F_j$.

(5) $G_5$: $F_j| \equiv W_i| \equiv W_i \longrightarrow^{S} K\ F_j$.

(6) $G_6$: $CS| \equiv W_i| \equiv W_i \longrightarrow^{S} K\ F_j$.

(7) $G_7$: $CS| \equiv F_j| \equiv W_i \longrightarrow^{S} K\ F_j$.

*3.2.3. Idealizing the Communication Messages.*

(1) $M_1$: $W_i \longrightarrow CS$: $\{V_1, V_2, RID_W\}$.

(2) $M_2$: $F_j \longrightarrow CS$: $\{V_3, V_4, RID_F\}$.

(3) $M_3$: $CS \longrightarrow W_i$: $\{E_1\}$.

(4) $M_4$: $CS \longrightarrow F_j$: $\{E_2\}$.

*3.2.4. Initial Assumptions*

(1) $A_1$: $CS| \equiv CS \overset{RPW_W}{\rightleftharpoons} W_i$.

(2) $A_2$: $CS| \equiv \#(a)$.

(3) $A_3$: $CS| \equiv W_i| \Rightarrow a$.

(4) $A_4$: $CS| \equiv CS \overset{Q_F}{\rightleftharpoons} F_j$.

(5) $A_5$: $CS| \equiv \#(b)$.

(6) $A_6$: $CS| \equiv F_j| \Rightarrow b$.

(7) $A_7$: $CS| \equiv RID'_W$.

(8) $A_8$: $CS| \equiv RID_F$.

(9) $A_9$: $W_i| \equiv W_i \overset{h(ID_W \| RPW_W)}{\rightleftharpoons} CS$.

(10) $A_{10}$: $W_i| \equiv \#(c)$.

(11) $A_{11}$: $W_i| \equiv CS| \Rightarrow (b, c, s'_W, RID'_W, RID_F)$.

(12) $A_{12}$: $F_j| \equiv \#(b)$.

(13) $A_{13}$: $F_j| \equiv F_j \overset{h(ID_F \| Q_F)}{\rightleftharpoons} CS$.

(14) $A_{14}$: $F_j| \equiv \#(c)$.

(15) $A_{15}$: $F_i| \equiv CS| \Rightarrow (a, c, s'_F, RID'_W)$.

(16) $A_{16}$: $W_i| \equiv \#(a)$.

*3.2.5. Detailed Proof.* From $M_1$, we can obtain $S_1: CS \triangleleft \left\{V_1: \langle a, I\ D_W \rangle_{RPW_W}, V_2, RID_W\right\}$. After simplification, it becomes $S_2: CS \triangleleft \left\{\langle a, ID_W \rangle_{RPW_W}\right\}$. Based on $A_1$ and $S_2$, using the M-M rule, we obtain $S_3: CS| \equiv W_i \mid \sim (a, ID_W)$. Based on further derivation, we obtain $S_4: CS| \equiv W_i \mid \sim a$. According to $A_2$ and $S_4$, using the N-V rule, we obtain $S_5: CS| \equiv W_i| \equiv a$. Additionally, based on $A_3$ and $S_5$, using the jurisdiction rule, we obtain $S_6: CS \mid \equiv a$.

From $M_2$, we can obtain $S_7: CS \triangleleft \left\{V_3: \langle b, ID_F \rangle_{Q_F}, V_4, RID_F\right\}$. After simplification, it becomes $S_8: CS \triangleleft \{\langle b, ID_F \rangle_{Q_F}\}$. According to $A_4$ and $S_8$, using the M-M rule, we obtain $S_9: CS \mid \equiv F_j \mid \sim (b, ID_F)$. Based on further derivation, we obtain $S_{10}: CS \mid \equiv F_j \mid \sim b$. Based on $A_5$ and $S_{10}$, using the N-V rule, we obtain $S_{11}: CS| \equiv F_j| \equiv b$. Based on $A_6$ and $S_{11}$, using the jurisdiction rule, we have $S_{12}: CS| \equiv b$. Because $SK = h(a \| b \| c \| RID'_W \| RID_F)$ and based on $A_7, A_8, S_6$, and $S_{12}$, we have $S_{13}: CS| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ ($G_3$). Based on $A_2$ and $S_{13}$, using the S-K rule, we have $S_{14}: CS| \equiv W_i| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ ($G_6$). According to $A_5$ and $S_{14}$, using the S-K rule, we have $S_{15}: CS| \equiv F_j| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ ($G_7$).

From $M_3$, we can obtain $S_{16}: W_i \triangleleft \{E_1: \langle b, c, s'_W, RID'_W, RID_F \rangle_{h(ID_W \| RPW_W)}\}$. According to $A_9$ and $S_{16}$, using the M-M rule, we obtain $S_{17}: W_i| \equiv CS| \sim (b, c, s'_W, RID'_W, RID_F)$. Based on $A_{10}$ and using the freshness rule, we have $S_{18}: W_i| \equiv \#(b, c, s'_W, RID'_W, RID_F)$. Based on $S_{17}$ and $S_{18}$, using the N-V rule, we obtain $S_{19}: W_i| \equiv CS| \equiv (b, c, s'_W, RID'_W, RID_F)$. Based on $A_{11}$ and $S_{19}$, using the jurisdiction rule, we have $S_{20}: W_i| \equiv (b, c, s'_W, RID'_W, RID_F)$. Based on further derivation, we obtain $S_{21}: W_i| \equiv b$, $S_{22}: W_i| \equiv c$, $S_{23}: W_i \; | \; \equiv RID'_W$, and $S_{24}: W_i| \equiv RID_F$. Because $SK = h(a\|b\|c\|RID'_W\|RID_F)$ and using $S_{21} - S_{24}$, we have $S_2: W_i| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ $(G_1)$. Based on $A_{12}$ and $S_{25}$, using the S-K rule, we obtain $S_{26}: W_i| \equiv F_j| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ $(G_4)$.

From $M_4$, we can obtain $S_{27}: F_j \lhd \{E_2: \langle a, c, s'_F, RID'_F \rangle_{(ID_F\|Q_F)}\}$. Based on $A_{13}$ and $S_{27}$, using the M-M rule, we obtain $S_{28}: F_j| \equiv CS| \sim (a, c, s'_F, RID'_W)$. According to $A_{14}$, using the freshness rule, we have $S_{29}: F_j| \equiv \#(a, c, s'_F, RID'_W)$. Based on $S_{28}$ and $S_{29}$, using the N-V rule, we obtain $S_{30}: F_j| \equiv \#(a, c, s'_F, RID'_W)$. Based on $A_{15}$ and $S_{30}$, using the jurisdiction rule, we have $S_{31}: F_j| \equiv \#(a, c, s'_F, RID'_W)$. Based on further derivation, we obtain $S_{32}: F_j| \equiv a$, $S_{33}: F_j| \equiv c$, and $S_{34}: F_j| \equiv RID'_W$. Because $SK = h(a\|b\|c\|RID'_W\|RID_F)$ and using $S_{32} - S_{34}$, we have $S_3: F_j| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ $(G_2)$. According to $A_{16}$ and $S_{35}$, using the S-K rule, we obtain $S_{36}: F_j| \equiv W_i| \equiv W_i \overset{SK}{\leftrightarrow} F_j$ $(G_5)$.

### 3.3. *ProVerif.*

The formal analysis method has become one of the main protocol analyses in cryptography. ProVerif [35, 36] is a common formal analysis tool that uses logic programming language rules and an automatic reasoning algorithm to determine whether a given event can occur. Therefore, ProVerif verifies protocol confidentiality and supports operations such as hashing, symmetric encryption, and decryption. According to the specific process of the proposed protocol, we use ProVerif for simulation reasoning. The entire simulation process is divided into the declaration, process, event, query, and main function parts.

First, as shown in Figure 5, we define the public channel, secure channel, constants, variables, and constituent functions. Second, as shown in Figure 6, we declare the queries and the events: Wearable Device Started, Wearable Device Authed, and Wearable Device AcCloud Server indicate that $W_i$ starts authentication, $W_i$ completes authentication, and $W_i$ passes the authentication of $CS$, respectively. Fog Node AcCloud Server indicates that fog node $F_j$ has passed the authentication of $CS$. Moreover, Cloud Server AcWearable Device and Cloud Server AcFog Node indicate that $CS$ has passed the authentication of $W_i$ and $F_j$, respectively.

Third, as shown in Figure 7, we define the process and main function, which includes three processes: $W_i$, $F_j$, and $CS$. After all operations are completed, we run the ProVerif function and obtain the following results.

(1) Query not attacker $(SK_w)$ is true.

(2) Query not attacker $(SK_f)$ is true.

(3) Query not attacker $(SK_c)$ is true.

(4) Query
inj − event (WearableDeviceAuthed) == > inj − event (Wearable Device Started) is true.

(5) Query
inj − event (Cloud Server AcFog Node) ==
> inj − event (Cloud Server AcWearable Device) is true.

(6) Query
inj − event (Fog Node AcCloud Server) ==
> inj − event (CloudServer AcFog Node) is true.

(7) Query
inj − event (Wearable Device AcCloud Sever) ==
> inj − event (Fog Node AcCloud Sever) is true.

```
(***************** channel *****************)
free ch :channel. (*  public channel  *)
free sch: channel  [private]. (*  secure channel,  used for registering *)
(***************** shared keys***************)
free SKw:bitstring [private].
free SKf:bitstring [private].
free SKc:bitstring [private].
(*****************constants  *****************)
free  s:bitstring [private].
(*********** functions & reductions & equations**********)
fun h(bitstring) :bitstring. (hash  function)
fun mult(bitstring,bitstring) :bitstring. (scalar  multiplication  operation)
fun senc(bitstring,bitstring):bitstring.  (symmetric  encryption)
reduc forall m:bitstring, key:bitstring; sdec(senc(m,key),key)=m.
fun con(bitstring,bitstring):bitstring.  (concatenation  operation)
reduc forall m:bitstring n:bitstring; getmess(con(m,n))=m.
fun xor(bitstring,bitstring):bitstring.  (XOR operation)
equation forall m:bitstring, n:bitstring; xor(xor(m,n),n)=m.
```

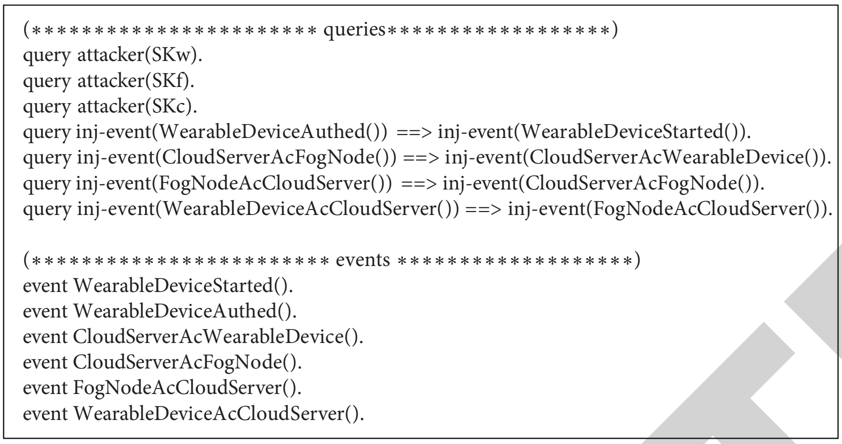
Figure 5: Definitions.

Results (1)–(3) show that the security of the session key is not threatened. Results (4)–(7) show that each process of the three entities is successfully initiated and terminated, and they ensure the correctness of each step of the protocol. Therefore, the proposed protocol has complete authentication steps and good session-key security.

### 3.4. *Informal Proof*

#### 3.4.1. *Insider Attacks.*

Suppose $\mathscr{A}$ obtains $\{P_W, R_W, RID_W, s_W\}$ in $W_i$'s memory and calculates the session key. Then, $\mathscr{A}$ needs to obtain the key, $h(ID_W\|RPW_W)$, used for symmetric decryption between $W_i$ and $CS$, where $RPW_W = h(ID_W\|PW_W\|r_W)$. Because $PW_W$ and $r_W$ are unknown, $\mathscr{A}$ cannot calculate the session key. Suppose $\mathscr{A}$ obtains $\{P_F, RID_F, s_F\}$ in $F_j$'s memory and calculates the session key. Then, $\mathscr{A}$ needs to obtain key $h(ID_F\|Q_F)$ used for symmetric decryption between $F_j$ and $CS$, where $Q_F = h(ID_F\|r_F)$. Because $r_F$ is unknown, $\mathscr{A}$ cannot calculate the session key. Suppose $\mathscr{A}$ obtains $\{RID_W, ID_W, s_W, RID_F, s_F\}$ in $CS$'s database and calculates the session key. Then, $\mathscr{A}$ needs to obtain the master key $s$ of $CS$, which is used to calculate the private information $a = V_1 \oplus h(ID_W\|s_W\|s)$ and the key $RID_W \oplus h(ID_W\|s_W\|s)$ for symmetric decryption between $W_i$ and $CS$ and the private information $b = V_3 \oplus h(s_F\|s)$ and the key $RID_F \oplus h(s_F\|s)$ for symmetric decryption between $F_j$ and $CS$. Because $s$ is known only by $CS$ and $\mathscr{A}$ cannot obtain it, the session key cannot be calculated. Therefore, the scheme is resistant to internal attacks.



```
(*********************** queries*****************)
query attacker(SKw).
query attacker(SKf).
query attacker(SKc).
query inj-event(WearableDeviceAuthed()) ==> inj-event(WearableDeviceStarted()).
query inj-event(CloudServerAcFogNode()) ==> inj-event(CloudServerAcWearableDevice()).
query inj-event(FogNodeAcCloudServer()) ==> inj-event(CloudServerAcFogNode()).
query inj-event(WearableDeviceAcCloudServer()) ==> inj-event(FogNodeAcCloudServer()).

(*********************** events ******************)
event WearableDeviceStarted().
event WearableDeviceAuthed().
event CloudServerAcWearableDevice().
event CloudServerAcFogNode().
event FogNodeAcCloudServer().
event WearableDeviceAcCloudServer().
```

Figure 6: Queries and events.

*3.4.2. Man-in-the-Middle Attacks.* Suppose $\mathscr{A}$ intercepts messages $M_1 = \{V_1, V_2, RID_W\}$ and $M_2 = \{V_1, V_2, RID_W, V_3, V_4, RID_F\}$ and forges them to pass the authentication of $CS$ and then intercepts $M_3 = \{E_1, E_2\}$ and $M_4 = \{E_1\}$ and forges $CS$ to pass the authentication of $F_j$ and $W_i$, respectively. First, assume that $\mathscr{A}$ forges the message from $W_i$. $CS$ determines the identity of $W_i$ by verifying $V_2 \overset{?}{=} h(a\|ID_W\|RID_W\|RID_F\|V_1)$, where $ID_W$ is stored in the registration phase, and $\mathscr{A}$ cannot be obtained in the authentication phase. Second, assume that $\mathscr{A}$ forges the message from $F_j$. $CS$ determines the identity of $F_j$ by verifying $V_4 \overset{?}{=} h(b\|s_F\|RID_W\|RID_F\|V_3)$, where $s_F$ is stored in the registration phase, and $\mathscr{A}$ cannot be obtained in the authentication phase. In other words, $\mathscr{A}$ cannot pass the verification at the $CS$ end and cannot continue to intercept $M_3$ and $M_4$. Therefore, the proposed scheme successfully resists man-in-the-middle attacks.

*3.4.3. Replay Attacks.* $\mathscr{A}$ attempts to replay messages $M_1 = \{V_1, V_2, RID_W\} M_2 = \{V_1, V_2, RID_W, V_3, V_4, RID_F\}$, $M_3 = \{E_1, E_2\}$, and $M_4 = \{E_1\}$ in the public channel. $M_1$ and $M_2$ are updated with random numbers $a$ and $b$, respectively, and $\mathscr{A}$ cannot obtain $a$ and $b$. $M_3$ and $M_4$ are updated by random numbers $a, b$, and $c$, and $\mathscr{A}$ cannot replay. Even if $\mathscr{A}$ replays these messages, it will cause the session to terminate. Therefore, the proposed scheme can resist replay attacks.

*3.4.4. Anonymity and Untraceability.* In the proposed protocol, the identities of $W_i$ and $F_j$ are not transmitted directly to the public channel. Moreover, $ID_W$ and $ID_F$ are protected by $h(ID_W\|PW_W\|r_W)$ and $h(ID_F\|r_F)$, respectively. Therefore, $\mathscr{A}$ cannot know the real identities of $W_i$ and $F_j$ during the entire authentication process and cannot trace them by intercepting information. Therefore, the proposed scheme provides device anonymity and fog node anonymity.

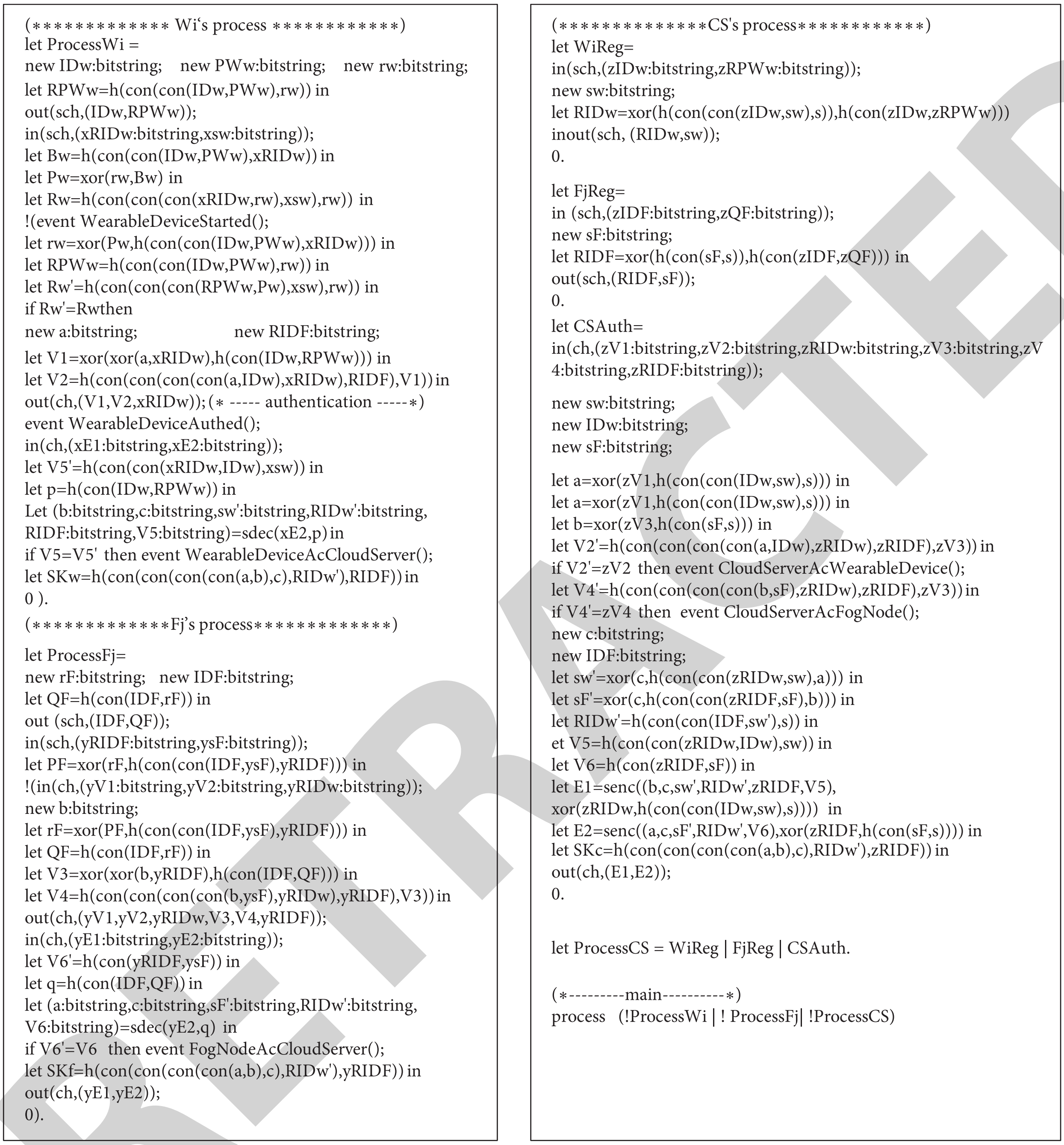
*3.4.5. Clogging Attacks.* $\mathscr{A}$ attempts to launch clogging attacks by forging request message $M_2 = \{V_1, V_2, RID_W, V_3, V_4, RID_F\}$. $\mathscr{A}$ can select random numbers $a, b$ and calculate $V_1 = a \oplus RID_W \oplus h(ID_W \| RPW_W)$, $V_2 = h(a\|ID_W\|RID_W\|RID_F\|V_1)$, $V_3 = b \oplus RID_F \oplus h(ID_F \| Q_F)$, and $V_4 = h(b\|s_F\|RID_W\|RID_F\|V_3)$. To calculate these four verification values, $\mathscr{A}$ also needs $ID_W$, $RPW_W$, $ID_F$, $r_F$, and $s_F$. However, the identity information $(ID_W, ID_F)$ is confidential, and $RPW_W$ is only known by $W_i$, and $r_F, s_F$ are updated in each communication. In other words, $\mathscr{A}$ cannot construct $V_1, V_2, V_3, V_4$ and make them pass the verification of $CS$. Therefore, our scheme can resist clogging attacks.

## 4. Performance Evaluation

The proposed scheme and five related protocols are analyzed for performance evaluation. These five schemes were proposed by Jia et al. [24], Wazid et al. [25], Chen et al. [26], Wu et al. [28], and Ali et al. [29].

*4.1. Security Evaluation.* Table 3 presents the security evaluation. $SA_1 - SA_{11}$, respectively, represent insider attacks, offline password-guessing attacks, impersonation attacks, clogging attacks, user anonymity, user untraceability, fog node anonymity, replay attacks, man-in-the-middle attacks, perfect forward security, and known session-specific temporary information attacks. Note that clogging attacks [26] mean that an adversary can force a legitimate user to process a fake request sent by him disguised as a legitimate user, resulting in resource clogging. "$\surd$" indicates that it can resist this attack. "$\chi$" indicates that the attack cannot be resisted. According to Table 3, we can see that Jia et al.'s scheme [24] and Wazid et al.'s scheme [25] cannot provide user anonymity and user untraceability. In addition, the scheme in [24] cannot resist impersonation attacks and known session-specific temporary information attacks and cannot provide fog-node anonymity. The scheme in [25] cannot resist clogging attacks. The schemes in [26, 28, 29] and our scheme have good security.

*4.2. Computation Cost Evaluation.* The evaluation environment was a Windows 10 operating system with an Intel (R) Core (TM) i5-8500 CPU at 3.00 Hz, and the memory was 8G. The development software is IntelliJ idea version 2019.3,



```
(************* Wi‘s process ************)
let ProcessWi =
new IDw:bitstring;    new PWw:bitstring;    new rw:bitstring;
let RPWw=h(con(con(IDw,PWw),rw)) in
out(sch,(IDw,RPWw));
in(sch,(xRIDw:bitstring,xsw:bitstring));
let Bw=h(con(con(IDw,PWw),xRIDw)) in
let Pw=xor(rw,Bw) in
let Rw=h(con(con(con(xRIDw,rw),xsw),rw)) in
!(event WearableDeviceStarted();
let rw=xor(Pw,h(con(con(IDw,PWw),xRIDw))) in
let RPWw=h(con(con(IDw,PWw),rw)) in
let Rw'=h(con(con(con(RPWw,Pw),xsw),rw)) in
if Rw'=Rwthen
new a:bitstring;                    new RIDF:bitstring;
let V1=xor(xor(a,xRIDw),h(con(IDw,RPWw))) in
let V2=h(con(con(con(con(a,IDw),xRIDw),RIDF),V1)) in
out(ch,(V1,V2,xRIDw)); (* ----- authentication -----*)
event WearableDeviceAuthed();
in(ch,(xE1:bitstring,xE2:bitstring));
let V5'=h(con(con(xRIDw,IDw),xsw)) in
let p=h(con(IDw,RPWw)) in
Let (b:bitstring,c:bitstring,sw':bitstring,RIDw':bitstring,
RIDF:bitstring,V5:bitstring)=sdec(xE2,p) in
if V5=V5'  then event WearableDeviceAcCloudServer();
let SKw=h(con(con(con(con(a,b),c),RIDw'),RIDF)) in
0 ).
(*************Fj’s process*************)

let ProcessFj=
new rF:bitstring;   new IDF:bitstring;
let QF=h(con(IDF,rF)) in
out (sch,(IDF,QF));
in(sch,(yRIDF:bitstring,ysF:bitstring));
let PF=xor(rF,h(con(con(IDF,ysF),yRIDF))) in
!(in(ch,(yV1:bitstring,yV2:bitstring,yRIDw:bitstring));
new b:bitstring;
let rF=xor(PF,h(con(con(IDF,ysF),yRIDF))) in
let QF=h(con(IDF,rF)) in
let V3=xor(xor(b,yRIDF),h(con(IDF,QF))) in
let V4=h(con(con(con(con(b,ysF),yRIDw),yRIDF),V3)) in
out(ch,(yV1,yV2,yRIDw,V3,V4,yRIDF));
in(ch,(yE1:bitstring,yE2:bitstring));
let V6'=h(con(yRIDF,ysF)) in
let q=h(con(IDF,QF)) in
let (a:bitstring,c:bitstring,sF':bitstring,RIDw':bitstring,
V6:bitstring)=sdec(yE2,q)  in
if V6'=V6   then event FogNodeAcCloudServer();
let SKf=h(con(con(con(con(a,b),c),RIDw'),yRIDF)) in
out(ch,(yE1,yE2));
0).
```

(a1) Process

```
(*************CS's process************)
let WiReg=
in(sch,(zIDw:bitstring,zRPWw:bitstring));
new sw:bitstring;
let RIDw=xor(h(con(con(zIDw,sw),s)),h(con(zIDw,zRPWw)))
inout(sch, (RIDw,sw));
0.

let FjReg=
in (sch,(zIDF:bitstring,zQF:bitstring));
new sF:bitstring;
let RIDF=xor(h(con(sF,s)),h(con(zIDF,zQF))) in
out(sch,(RIDF,sF));
0.
let CSAuth=
in(ch,(zV1:bitstring,zV2:bitstring,zRIDw:bitstring,zV3:bitstring,zV
4:bitstring,zRIDF:bitstring));

new sw:bitstring;
new IDw:bitstring;
new sF:bitstring;

let a=xor(zV1,h(con(con(IDw,sw),s))) in
let a=xor(zV1,h(con(con(IDw,sw),s))) in
let b=xor(zV3,h(con(sF,s))) in
let V2'=h(con(con(con(con(a,IDw),zRIDw),zRIDF),zV3)) in
if V2'=zV2  then event CloudServerAcWearableDevice();
let V4'=h(con(con(con(con(b,sF),zRIDw),zRIDF),zV3)) in
if V4'=zV4  then  event CloudServerAcFogNode();
new c:bitstring;
new IDF:bitstring;
let sw'=xor(c,h(con(con(zRIDw,sw),a))) in
let sF'=xor(c,h(con(con(zRIDF,sF),b))) in
let RIDw'=h(con(con(IDF,sw'),s)) in
et V5=h(con(con(zRIDw,IDw),sw)) in
let V6=h(con(zRIDF,sF)) in
let E1=senc((b,c,sw',RIDw',zRIDF,V5),
xor(zRIDw,h(con(con(IDw,sw),s))))  in
let E2=senc((a,c,sF',RIDw',V6),xor(zRIDF,h(con(sF,s)))) in
let SKc=h(con(con(con(con(a,b),c),RIDw'),zRIDF)) in
out(ch,(E1,E2));
0.


let ProcessCS = WiReg | FjReg | CSAuth.

(*---------main----------*)
process   (!ProcessWi | ! ProcessFj| !ProcessCS)
```

(a2) Process

Figure 7: Processes.

Table 3: Security evaluation.

| | [24] | [25] | [26] | [28] | [29] | Our scheme |
|---|---|---|---|---|---|---|
| $SA_1$ | √ | √ | √ | √ | √ | √ |
| $SA_2$ | √ | √ | √ | √ | √ | √ |
| $SA_3$ | $\chi$ [28] | √ | √ | √ | √ | √ |
| $SA_4$ | √ | $\chi$ [29] | √ | √ | √ | √ |
| $SA_5$ | $\chi$ [28] | $\chi$ [29] | √ | √ | √ | √ |
| $SA_6$ | $\chi$ [28] | $\chi$ [29] | √ | √ | √ | √ |
| $SA_7$ | $\chi$ [28] | √ | √ | √ | √ | √ |
| $SA_8$ | √ | √ | √ | √ | √ | √ |
| $SA_9$ | √ | √ | √ | √ | √ | √ |
| $SA_{10}$ | √ | √ | √ | √ | √ | √ |
| $SA_{11}$ | $\chi$ [28] | √ | √ | √ | √ | √ |

Table 4: Computation cost evaluation.

| | [24] | [25] | [26] | [28] | [29] | Our scheme |
|---|---|---|---|---|---|---|
| User/WD | $5T_h + 2T_m + T_{\text{map}} + T_{ex} \approx 38.62$ | $21T_h + T_a + 2T_m + T_{fe} \approx 25.934$ | $8T_h + 2T_m + T_{fe} \approx 25.832$ | $7T_h + 3T_m + T_{\text{map}} + T_{ex} \approx 47.228$ | $12T_h + T_a + 4T_m + T_{fe} \approx 43.098$ | $7T_h + T_s \approx 8.228$ |
| Fog node | $4T_h + 2T_m + T_{\text{map}} + T_{ex} \approx 38.616$ | $16T_h + T_a + 3T_m \approx 25.914$ | $6T_h + 4T_m \approx 34.424$ | $4T_h + 3T_m + T_{\text{map}} + T_{ex} \approx 47.216$ | $10T_h + T_a + 5T_m \approx 43.09$ | $6T_h + T_s \approx 8.224$ |
| CS | $9T_h + 3T_m + T_{\text{map}} + T_{ex} \approx 47.236$ | — | $12T_h + 3T_m \approx 25.848$ | $10T_h + 4T_m + T_{\text{map}} + T_{ex} \approx 55.84$ | — | $10T_h + 2T_s \approx 16.44$ |
| Device | — | $17T_h \approx 0.068$ | — | — | $10T_h \approx 0.04$ | — |
| Total | 124.472 ms | 51.916 ms | 86.104 ms | 150.284 ms | 86.218 ms | 32.892 ms |

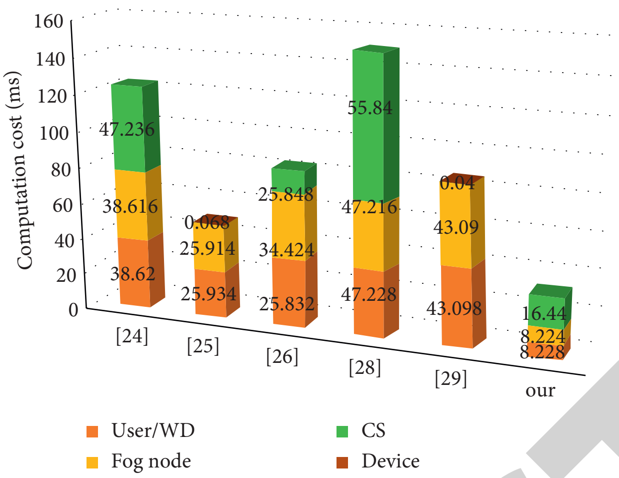


Figure 8: Computation cost comparison.

Table 5: Communication cost evaluation.

| | Rounds | Cost (bits) |
|---|---|---|
| [24] | 4 | $6C_p + 9C_h + 5C_t = 5696$ |
| [25] | 3 | $4C_p + 10C_h + 3C_t = 4800$ |
| [26] | 4 | $10C_p + 9C_h + 5C_t = 7744$ |
| [28] | 4 | $6C_p + 9C_h + 5C_t = 5696$ |
| [29] | 3 | $4C_a + 10C_h + 3C_t = 4800$ |
| Our scheme | 4 | $9C_h + 3C_s = 3072$ |

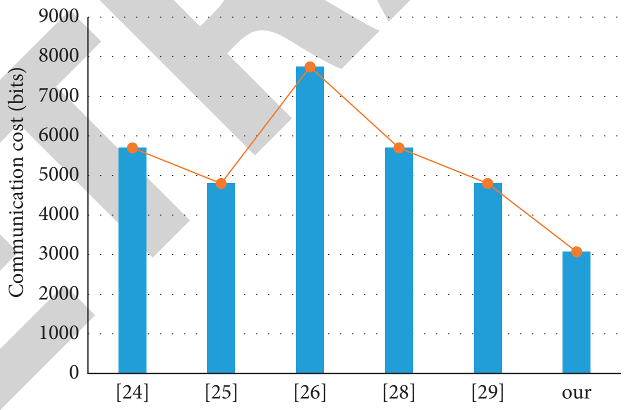


Figure 9: Communication cost comparison.

which is based on the calls of the Java pairing library, signature library, and symmetric encryption/decryption function. Table 4 presents the computation cost evaluation of the AKA phase. $T_h$ represents the time of general hash operation, $T_a$ represents the operation time of point addition, $T_m$ represents the operation time of scalar multiplication of elliptic curve, $T_{fe}$ represents the operation time of fuzzy function, $T_s$ represents the operation time of symmetric encryption and decryption, $T_{\text{map}}$ represents the operation time of bilinear pair, and $T_{ex}$ represents the operation time of exponential operation. It should be noted that $T_h = 0.004$ ms, $T_a = 0.05$ ms, $T_m = 8.6$ ms, $T_{fe} = T_m$, $T_s = 8.2$ ms, $T_{\text{map}} = 10.6$ ms, and $T_{ex} = 10.8$ ms. According to Table 4, it is evident that the computational cost of our proposed scheme is far less than that of the other five schemes. Figure 8 shows the advantages of the proposed scheme in terms of computational cost.

*4.3. Communication Cost Evaluation.* Assume that the point of the elliptic curve occupies 512 bits, the hash operation and symmetric encryption and decryption operation occupy 256 bits, and the timestamp occupies 64 bits. Table 5 presents the communication cost evaluation of the AKA phase, where $C_p$, $C_h$, $C_s$, and $C_t$ represent the point, hash operation, symmetric encryption and decryption operation, and

Table 6: Performance optimization ratio.

| | Computational performance speed (times) | Communication performance speed (times) |
|---|---|---|
| Our scheme | 1 | 1 |
| [24] | 3.784 | 1.854 |
| [25] | 1.578 | 1.563 |
| [26] | 2.618 | 2.521 |
| [28] | 4.569 | 1.854 |
| [29] | 2.621 | 1.563 |

timestamp, respectively. According to Table 5, the proposed protocol has the lowest communication cost. Figure 9 shows the advantages of the proposed scheme in terms of communication cost.

After evaluating our scheme and the other four related schemes in terms of security, computation cost, and communication cost, it is obvious that our scheme has great advantages in these three aspects at the same time. Our scheme not only ensures security but also has the least computation cost and communication cost. Table 6 shows the ratio of other related schemes and the proposed scheme in terms of computational performance and communication performance. According to Table 6, [24–26, 28, 29] are, respectively, 378.4%, 157.8%, 261.8%, 456.9%, and 262.1% of the proposed scheme in terms of computational performance and 185.4%, 156.3%, 252.1%, 185.4%, and 156.3% of the proposed scheme in terms of communication performance. Therefore, our scheme has good advantages in performance.

## 5. Conclusion

Researchers have proposed many AKA schemes based on fog computing. Some of these schemes are for the healthcare environment; however, these have low security and high cost consumption. Therefore, we propose a fog-driven secure authentication and key exchange scheme for wearable health monitoring systems. Using a formal analysis, BAN logic, ProVerif tools, and an informal analysis, we find that our scheme can resist known attack methods. The performance comparison with related protocols shows that the proposed scheme has significant advantages in terms of both computational and communication costs. Therefore, our scheme is more suitable for a wearable health monitoring system.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIOT," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713–11724, 2020.

[2] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, p. 1, 2020.

[3] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[4] T. N. Tu, "A fuzzy approach of large size remote sensing image clustering," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 4, pp. 187–198, 2020.

[5] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

[6] J. Wu, M. Xu, F.-F. Liu, M. Huang, L.-H. Ma, and Z.-M. Lu, "Solar wireless sensor network routing algorithm based on multi-objective particle swarm optimization," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 12, no. 1, pp. 1–11, 2021.

[7] K. K. Venkatasubramanian, A. Banerjee, and S. K. Gupta, "Ekg-based key agreement in body sensor networks," in *Proceedings of the IEEE INFOCOM Workshops*, pp. 1–6, Phoenix, AZ, USA, April 2008.

[8] J. C. Sriram, M. Shin, T. Choudhury, and D. Kotz, "Activity-aware ecg-based patient authentication for remote health monitoring," in *Proceedings of the 2009 International Conference on Multimodal Interfaces*, pp. 297–304, Cambridge, MA, USA, November 2009.

[9] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Pska: usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–67, 2009.

[10] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "Opfka: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *Proceedings of the 2013 IEEE INFOCOM*, pp. 2274–2282, Turin, Italy, April 2013.

[11] M. Masdari, S. Ahmadzadeh, and M. Bidaki, "Key management in wireless body area network: challenges and issues," *Journal of Network and Computer Applications*, vol. 91, pp. 36–51, 2017.

[12] X. Li, J. Niu, M. Karuppiah, S. Kumari, and F. Wu, "Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications," *Journal of Medical Systems*, vol. 40, no. 12, pp. 1–12, 2016.

[13] N. Radhakrishnan and M. Karuppiah, "An efficient and secure remote user mutual authentication scheme using smart cards for Telecare medical information systems," *Informatics in Medicine Unlocked*, vol. 16, Article ID 100092, 2019.

[14] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved SMS applied for QR

code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.

[15] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, "Hiding sensitive information in eHealth datasets," *Future Generation Computer Systems*, vol. 117, pp. 169–180, 2021.

[16] Z. Zhang, S. Chen, X. Sun, and Y. Liang, "Trajectory privacy protection based on spatial-time constraints in mobile social networks," *Journal of Network Intelligence*, vol. 6, no. 3, pp. 485–499, 2021.

[17] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2795–2805, 2017.

[18] 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.

[19] C.-M. Chen, B. Xiang, T.-Y. Wu, and K.-H. Wang, "An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks," *Applied Sciences*, vol. 8, no. 7, p. 1074, 2018.

[20] 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.

[21] M. Kompara, S. H. Islam, and M. Hölbl, "A robust and efficient mutual authentication and key agreement scheme with untraceability for wbans," *Computer Networks*, vol. 148, pp. 196–213, 2019.

[22] S. F. Aghili, H. Mala, M. Shojafar, and P. Peris-Lopez, "Laco: lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT," *Future Generation Computer Systems*, vol. 96, pp. 410–424, 2019.

[23] 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.

[24] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[25] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019.

[26] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, pp. 1–16, 2020.

[27] S. Shamshad, M. S. Obaidat, Minahil, U. Shamshad, S. Noor, and K. Mahmood, "On the security of authenticated key agreement scheme for fog-driven IoT healthcare system," in *Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems*, pp. 1760–1765, Gwailor, Indai, August 2021.

[28] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key Agreement scheme for fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, Article ID 6658041, 16 pages, 2021.

[29] Z. Ali, S. A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, and Y. B. Zikria, "A clogging resistant secure authentication scheme for fog computing services," *Computer Networks*, vol. 185, Article ID 107731, 2021.

[30] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, vol. 51, no. 4, pp. 557–594, 2004.

[31] Y. Yu, O. Taylor, R. Li, and B. Sunagawa, "An extended chaotic map-based authentication and key agreement scheme for multi-server environment," *Mathematics*, vol. 9, no. 8, p. 798, 2021.

[32] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[33] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London A. Mathematical and Physical Sciences*, vol. 426, pp. 233–271, 1871.

[34] Y. Luo, W. Zheng, and Y.-C. Chen, "An anonymous authentication and key exchange protocol in smart grid," *Journal of Network Intelligence*, vol. 6, no. 2, pp. 206–215, 2021.

[35] S. A. Chaudhry, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid"," *International Journal of Electrical Power and Energy Systems*, vol. 125, Article ID 106529, 2021.

[36] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, and S. K. H. Lslam, "Improved ECC-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, Article ID 6627956, 14 pages, 2021.