Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.022667

check for updates

ARTICLE

PSAP-WSN: A Provably Secure Authentication Protocol for 5G-Based Wireless Sensor Networks

Xuanang Li[1], Shuangshuang Liu[1], Saru Kumari[2] and Chien-Ming Chen[1,*]

[1]Shandong University of Science and Technology, Qingdao, 266590, China

[2]Department of Mathematics, Chaudhary Charan Singh University, Meerut, 250004, India

*Corresponding Author: Chien-Ming Chen. Email: chienmingchen@ieee.org

Received: 20 March 2022 Accepted: 27 May 2022

ABSTRACT

Nowadays, the widespread application of 5G has promoted rapid development in different areas, particularly in the Internet of Things (IoT), where 5G provides the advantages of higher data transfer rate, lower latency, and widespread connections. Wireless sensor networks (WSNs), which comprise various sensors, are crucial components of IoT. The main functions of WSN include providing users with real-time monitoring information, deploying regional information collection, and synchronizing with the Internet. Security in WSNs is becoming increasingly essential because of the across-the-board nature of wireless technology in many fields. Recently, Yu et al. proposed a user authentication protocol for WSN. However, their design is vulnerable to sensor capture and temporary information disclosure attacks. Thus, in this study, an improved protocol called PSAP-WSN is proposed. The security of PSAP-WSN is demonstrated by employing the ROR model, BAN logic, and ProVerif tool for the analysis. The experimental evaluation shows that our design is more efficient and suitable for WSN environments.

KEYWORDS

5G; wireless sensor networks; IoT; authentication protocol

1 Introduction

Historically, communication modes have evolved constantly, progressing through flying pigeons, post stations, wireless telegrams, fixed telephones, and mobile phones. Currently, most countries enjoy excellent Internet communication. Humans can control objects around them, as well as distant objects. Consequently, the Internet of Things (IoT) [1–3] emerged. In 1990, the world's first IoT device, Xerox's vending machine, appeared. In 1999, Professor Kevin Ashton of the Massachusetts Institute of Technology first proposed the definition of the IoT [4]. IoT now controls distant things from theory to practice. However, the slow transmission speed of information in the IoT, high latency, and limited support for connected devices are significant problems. 5G has emerged to solve these problems [5,6], providing higher data transfer rates, lower latency, and more connections to facilitate the efficient application of IoT worldwide [7]. Currently, IoT has been deployed in various applications [8–10].

In the last two or three decades, people's lives have continuously improved with the vigorous development of the Internet. Expectations for quality of life have generally increased. However, traditional electronic devices cannot meet the growing needs of people. With the rapid development of IoT, sensors joined IoT to form wireless sensor networks (WSNs) [11–13], meeting people's needs for work, production, study, entertainment, and other aspects. Sensors are ubiquitous in everyday life. As shown in Fig. 1, different types of sensors are deployed in homes, hospitals, schools, and other environments. In hospitals, patients are equipped with sensors to self-monitor physiological indicators, and doctors can remotely analyze these data to provide timely medical services to patients. Sensors are placed in schools or homes to collect temperature, carbon monoxide, or pyroelectric data.

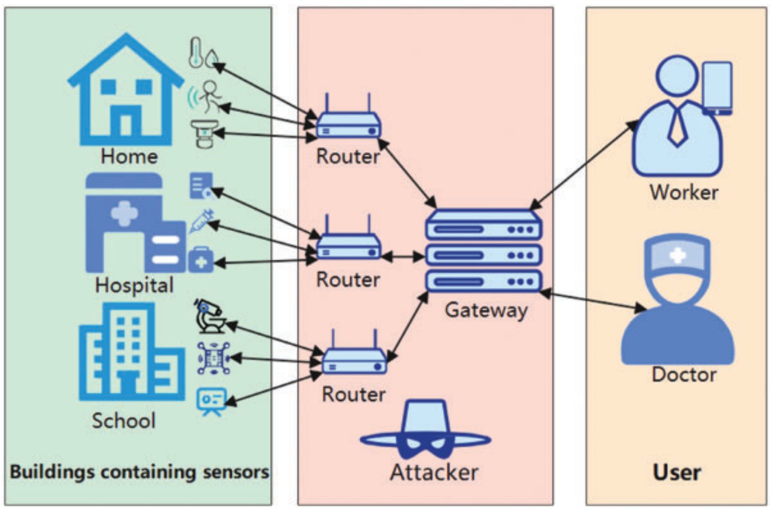


Figure 1: Wireless sensor network environment

Although WSNs make people's lives more efficient and convenient, they also create security problems [14–16]. For example, in 2016, a massive network outage in the eastern United States was caused by hackers who exploited vulnerabilities in communication protocols through a distributed denial-of-service attack [17,18]. Therefore, security is a significant problem that must be solved in WSNs [19,20]. In a typical WSN, two vital security issues must be carefully considered. First, because all sensing data are transmitted through a public channel, the data must be encrypted. Second, all members in a WSN should authenticate each other before sending data [21,22]. Many authentication protocols have been proposed to overcome these two security issues [23–25].

Recently, Yu et al. [26] proposed an authentication protocol called SLUA-WSN, declaring that it is secure against various attacks. Nevertheless, their design remains insecure against temporary information disclosure and sensor capture attacks [26]. To address these vulnerabilities, in this study, a novel authentication protocol, called PSAP-WSN, is proposed. To demonstrate that PSAP-WSN is secure and addresses the vulnerability issues, the ROR model, BAN logic, and ProVerif tools, which are three effective methods for proving the security of an authentication protocol, were employed. In addition, a performance evaluation was conducted to demonstrate that PSAP-WSN is suitable for WSN environments.

The remainder of this paper is organized as follows. In Sections 2 and 3, related work and Yu et al.'s protocol are described, respectively. In Section 4, it is demonstrated that Yu et al.'s protocol is insecure.

In Section 5, new solutions are proposed. In Sections 6 and 7, a security analysis and performance evaluation are provided, respectively.

2 Related Work

5G requires powerful security and privacy solutions because it connects all aspects of a communication network. Various security mechanisms have been proposed for 5G applications. In 2019, Lu et al. [27] recognized the crucial challenges of security and privacy in 5G vehicle-to-everything. In 2020, Liu et al. [28] proposed a federated learning framework to make 5G environments secure. In 2021, Afaq et al. [29] recognized essential security issues in 5G networks. Then, Yahaya et al. [30] proposed a privacy handover scheme for SDN-based 5G networks. In 2022, Yahaya et al. [30] provided an energy trading model for a 5G-deployed smart community based on blockchain technology.

Various authentication protocols have been proposed for WSNs. In 2015, Chang et al. [31] proposed an authentication protocol for protecting user privacy. However, some parameters of their protocols are not protected. Anonymity and backward confidentiality attacks may occur when users lose their smart cards. In 2017, Lu et al. [32] presented a three-factor authentication protocol with anonymity. In 2019, Mo et al. [33] analyzed Lu et al.'s protocol and concluded that it did not provide three-factor security. Therefore, an improved protocol was proposed. In 2020, Yu et al. [26] indicated that their protocol [33] was insecure against camouflage and session key exposure attacks. In addition, this protocol [33] does not provide anonymity. In 2020, Almuhaideb et al. [34] analyzed Yu et al.'s protocol and noted loopholes. Security problems occur if an adversary obtains both random numbers and sensitive information stored in a smart card. However, we believe that this attack is not reasonable because an adversary should simultaneously obtain two types of secret information.

3 Revisit SLUA-WSN

Here, Yu et al.'s design, which consists of sensor registration, user registration, and login and authentication phases, is revisited. The symbols and notations used are listed in Table 1.

Table 1: Notation definitions in SLUA-WSN

Notations	Definitions
U_i	The i_{th} user
S_j	The j_{th} sensor node
SID_j	S_j's identity
GWN	Gateway
ID_i	U_i's identity
PW_i	Password of of U_i
BIO_i	Biometric of U_i
R_g, R_u, R_s	Random numbers
K_{GWN}	Master key of GWN
$\mathcal{A}$	Attacker
X_j	Secret key of S_j
SK	Session key
T_i	The i_{th} timestamp
$\|\|$	Concatenation
$\oplus$	exclusive-or operation

(Continued)

Table 1 (continued)

Notations	Definitions
$h(\cdot)$	One-way hash function

3.1 Sensor Registration Phase

Assuming that a sensor S_j desires to enter a WSN, S_j must register with the gateway GWN first. GWN selects identity SID_j for S_j and calculates $X_j = h(SID_j||K_{GWN})$. Subsequently, GWN transmits $\{SID_j, X_j\}$ to S_j.

3.2 User Registration Phase

1. U_i enters his ID_i, PW_i and BIO_i and then calculates $Gen(BIO_i) = (R_i, P_i)$ and $MPW_i = h(PW_i||R_i)$, where Gen is a fuzzy extractor operation and U_i transmits $\{ID_i, MPW_i\}$ to GWN.
2. GWN generates R_g and calculates $MID_i = h(ID_i||h(K_{GWN}||R_g))$, $X_i = h(MID_i||R_g||K_{GWN})$, $Q_i = h(MID_i||MPW_i) \oplus X_i$ and $W_i = h(MPW_i||X_i)$. GWN deposits R_g in its own database and further issues a smart card storing $\{MID_i, Q_i, W_i\}$ to U_i.

3.3 Login and Authentication Phase

1. With the smart card, U_i inputs ID_i, PW_i, and BIO_i, and obtains $R_i = Rep(BIO_i, P_i)$, where Rep is another fuzzy extractor operation. U_i then calculates $MPW_i = h(PW_i||R_i)$, $X_i = h(MID_i||MPW_i) \oplus Q_i$, and $W_i^* = h(MPW_i||X_i)$ and verifies whether W_i^* is equal to W_i. If it is equal, U_i generates R_u and T_1 and calculates $M_1 = X_i \oplus R_u$, $CID_i = (ID_i||SID_j) \oplus h(MID_i||R_u||X_i)$, and $M_{UG} = h(ID_i||R_u||X_i||T_1)$. Now, U_i transmits $\{M_1, MID_i, CID_i, M_{UG}, T_1\}$ to GWN.
2. GWN examines the freshness of T_1 and obtains M_{UG}^* by calculating $X_i = h(MID_i||R_g||K_{GWN})$, $R_u = M_1 \oplus X_i$, $(ID_i||SID_j) = CID_i \oplus h(MID_i||R_u||X_i)$. GWN compares M_{UG}^* with the received M_{UG}. If they are equal, GWN calculates $M_2 = (R_u||R_g) \oplus h(SID_j||X_j||T_2)$ and $M_{GS} = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$ and then transmits $\{M_2, MID_i, M_{GS}, T_2\}$ to S_j.
3. S_j examines the freshness of T_2 and calculates $(R_u||R_g) = M_2 \oplus h(SID_j||X_j||T_2)$, $M_{GS}^* = h(MID_i||SID_j||R_u||R_g||X_j||T_2)$. S_j checks whether M_{GS}^* and the received M_{GS} are equal. Next, S_j generates R_s and T_3, calculates $M_3 = R_s \oplus h(R_u||SID_j||X_j||T_3)$, $M_{SG} = h(R_s||R_g||SID_j||X_j||T_3)$, and finally calculates the session key $SK = h(R_u||R_s)$ and $M_{SU} = h(SK||R_s||R_u||SID_j||MID_i)$. Now, S_j transmits $\{M_3, M_{SG}, M_{SU}, T_3\}$ to GWN.
4. GWN calculates $R_s = M_3 \oplus h(R_u||SID_j||X_j||T_3)$ and $M_{SG}^* = h(R_s||R_g||SID_j||X_j||T_3)$ after checking the freshness of T_3. GWN then checks whether M_{SG}^* and the received M_{SG} are equal. Next, GWN computes $MID_i^{new} = h(ID_i||h(K_{GWN}||R_g))$, $X_i^{new} = h(MID_i^{new}||R_g||K_{GWN})$, $M_4 = (MID_i^{new}||X_i^{new}||R_s||R_g) \oplus h(MID_i||X_i||T_4)$, and $M_{GU} = h(R_u||R_g||MID_i||X_i||T_4)$. Thereafter, GWN transmits $\{M_4, M_{SU}, M_{GU}, T_4\}$ to U_i.
5. U_i first examines the freshness of T_4 and calculates $(MID_i^{new}||X_i^{new}||R_s||R_g) = M_4 \oplus h(MID_i||X_i||T_4)$ and $M_{GU}^* = h(R_u||R_g||MID_i||X_i||T_4)$. In addition, U_i verifies whether M_{GU}^* is equal to the received M_{GU}. If they are equal, U_i obtains the session key $SK = h(R_u||R_s)$.

4 Attacks on the SLUA-WSN Protocol

This section analyzes the SLUA-WSN protocol [26]. The adversary model utilized in this study is presented, demonstrating that SLUA-WSN is insecure against sensor node capture and temporary information leakage attacks.

4.1 Adversary Model

The Dolev-Yao (DY) model [35] is a widely used and reasonable adversary model for analyzing authentication protocols [36]. Under the DY model, the protocol can be thoroughly and reasonably cryptanalyzed. Therefore, the DY model was used as the adversary model with $\mathcal{A}$ utilized to denote an attacker; the detailed attack capability is described below:

1. $\mathcal{A}$ can intercept/modify/delete messages submitted via a public channel.
2. $\mathcal{A}$ can steal temporary variables used in the process of an authentication protocol.
3. $\mathcal{A}$ can crack parameters stored in a smart card [37], implying that, once the user's smart card is stolen, sensitive parameters in this smart card will also be compromised by $\mathcal{A}$.
4. $\mathcal{A}$ can capture the sensor and obtain the information stored in it.

4.2 Sensor Node Capture Attack

According to the DY model, after capturing a sensor, $\mathcal{A}$ can capture the sensitive parameters stored therein. Various authentication protocols have considered this attack [38–41].

Assume that $\mathcal{A}$ captures a sensor S_j, and then $\mathcal{A}$ performs the following steps:

1. $\mathcal{A}$ obtains $\{SID_j, X_j\}$ stored in S_j.
2. $\mathcal{A}$ intercepts $\{M_1, M_2, M_4, MID_i, CID_i, M_{GS}, M_{UG}, T_1, T_2, T_4\}$ via a public channel.
3. $\mathcal{A}$ obtains $(R_u||R_g)$ by computing $M_2 \oplus h(SID_j||X_j||T_2)$.
4. With R_u and M_1, $\mathcal{A}$ can have X_i.
5. Now, $\mathcal{A}$ will have R_s by computing $M_4 \oplus h(MID_i||X_i||T_4)$.
6. Eventually, $\mathcal{A}$ can have SK because $SK = h(R_u||R_s)$.

Evidently, the SLUA-WSN protocol [26] cannot effectively resist sensor node capture attacks.

4.3 Temporary Information Leakage Attack

As mentioned in the adversary model, $\mathcal{A}$ steals temporary variables during the authentication process. Various authentication protocols have considered this attack [41–43].

Suppose that $\mathcal{A}$ obtains $\{R_u\}$, which is a temporary variable in this protocol. The following steps are then performed:

1. $\mathcal{A}$ intercepts $\{M_1, M_4, MID_i, T_4\}$ via a public channel.
2. $\mathcal{A}$ obtains X_i by computing $R_u \oplus M_1$.
3. $\mathcal{A}$ obtains $(MID_i^{new}||X_i^{new}||R_s||R_g)$ by computing $M_4 \oplus h(MID_i||X_i||T_4)$.
4. Eventually, $\mathcal{A}$ obtains SK because $SK = h(R_u||R_s)$.

5 PSAP-WSN

This section describes, in detail, the proposed PSAP-WSN, which consists of the pre-processing, user registration, login, and authentication phases. The symbols used in PSAP-WSN are listed in Table 2.

Table 2: Notations used in PSAP-WSN

Notations	Definitions
U_i	i_{th} user
S_j	j_{th} sensor node
$SUID_j$	Identity of S_j
GWN	Gateway node
UID_i	U_i's identity
UPW_i	U_i's password
$UBIO_i$	Biometric features of U_i
K_G	Master key of GWN
R_n, R_u, R_g, R_s	Random numbers
$Gen(\cdot)/Rep(\cdot)$	Generation/reproduction process of fuzzy extractor
$ENC_{PU}()/DES_{PR}()$	Public and private key encryption and decryption of gateway node
SK	Session keys produced by U_i, S_j
T_i	The i_{th} timestamp
$\mathcal{A}$	The attacker
$h(\cdot)$	One-way hash function
$x\|\|y$	Concatenation
$\oplus$	Exclusive-or operation

5.1 Pre-Processing Phase

GWN has to prepare some parameters for the sensors before they are deployed. This phase does not significantly differ from the SLUA-WSN protocol [26]. Fig. 2 illustrates this process. The detailed steps are as follows:

(1) GWN chooses the unique $SUID_j$ for S_j and uses its own key K_G to calculate $UA_j = h(SUID_j||K_G)$. Then, GWN submits $\{SUID_j, UA_j\}$ to S_j.

(2) S_j stores them in its local memory.

GWN		S_j
Distribute a unique $SUID_j$ for S_j $UA_j = h(SUID_j\|\|K_G)$	$\xrightarrow{\{SUID_j, UA_j\}}$	Store $SUID_j$ and UA_j in memory.

Figure 2: Pre-processing phase

5.2 User Registration Phase

All users need to register with GWN before entering the network. Assume that U_i desires to join this network; then, the user registration phase is initiated. In Fig. 3, the procedure followed in this phase is displayed. The detailed steps are as follows. Note that this phase is executed through a secure channel.

1. U_i inputs UID_i, UPW_i and $UBIO_i$ and computes $Gen(UBIO_i) = \langle UR_i, UP_i \rangle$. U_i then calculates $MUPW_i = h(UPW_i||UR_i)$ and encrypts $MUPW_i$ with GWN's public key PU. Thereafter, U_i sends $\{UR_i, UID_i, S\}$ to GWN.
2. GWN obtains $MUPW_i$ by decrypting S with his private key PR. Further, GWN generates R_n and calculates $MUID_i = h(h(K_G||R_n)||UID_i)$, $UA_i = h(K_G||R_n||MUID_i)$, $UB_i = UA_i \oplus h(MUID_i||MUPW_i)$, and $UC_i = h(MUPW_i||UA_i)$. GWN issues a smart card to U_i, which stores UB_i, UC_i, and $MUID_i$. GWN also stores R_n, UR_i and S in its database.

U_i		GWN
Read the user's biological information $UBIO_i$ Enter UID_i and UPW_i $Gen(UBIO_i) =< UR_i, UP_i >$ $MUPW_i = h(UPW_i\|\|UR_i)$ $S = ENC_{PU}(MUPW_i)$		
	$\xrightarrow{\{UR_i, UID_i, S\}}$	
		$MUPW_i = DES_{PR}(S)$ Generate R_n $MUID_i = h(h(K_G\|\|R_n)\|\|UID_i)$ $UA_i = h(K_G\|\|R_n\|\|MUID_i)$ $UB_i = UA_i \oplus h(MUID_i\|\|MUPW_i)$ $UC_i = h(MUPW_i\|\|UA_i)$ Store UB_i, UC_i and $MUID_i$ in a smart card Store R_n, UR_i and S in its database
	$\xleftarrow{\{Smart\ card\}}$	
Receive a smart card.		

Figure 3: User registration phase

5.3 Login and Authentication Phase

This phase is performed when the user is expected to connect to a specific sensor. Fig. 4 illustrates this process. Suppose that U_i wishes to connect to S_j; the following steps are then executed:

(1) U_i inserts his smart card and inputs $UBIO_i$, UID_i, and UP_i. U_i then computes $UR_i = Rep(BIO_i, UPW_i)$, $MUPW_i = h(UPW_i||UR_i)$, $UA_i = h(MUID_i||MUPW_i) \oplus UB_i$, and $UC_i' = h(MUPW_i||UA_i)$. The smart card checks whether UC_i equals UC_i'. Subsequently, U_i generates R_u and T_1 and calculates $M_1 = MUPW_i \oplus UID_i \oplus UA_i \oplus R_u$, $CUID_i = h(MUID_i||UA_i||R_u) \oplus (SUID_i||UID_i)$, and $K_{UG} = h(UA_i||R_u||UID_i||T_1)$. Now, U_i transmits $\{M_1, MUID_i, CUID_i, K_{UG}, T_1\}$ to GWN.

(2) GWN checks the freshness of T_1. GWN calculates $UA_i = h(MUID_i||R_n||K_G)$, $R_u = UA_i \oplus M_1 \oplus UID_i \oplus MUPW_i$, $(UID_i||SUID_j) = CUID_i \oplus h(MUID_i||UA_i||R_u)$, and $K_{UG}' = h(UA_i||R_u||UID_i||T_1)$. Now, GWN verifies whether K_{UG}' is equal to the K_{UG} that GWN received. If they are the same, GWN further calculates $M_2 = (R_u||R_g) \oplus h(SUID_j||UA_j||T_2)$ and $K_{GS} = h(MUID_i||SUID_j||R_u||R_g||UA_j||T_2)$ and then sends $\{M_2, MUID_i, M_{GS}, T_2\}$ to S_j.

U_i	GWN	S_j
Read the user's biological information $UBIO_i$ Enter UID_i and UPW_i $UR_i = Rep(UBIO_i, UP_i)$ $MUPW_i = h(UPW_i \| \| UR_i)$ $UA_i = h(MUID_i \| \| MUPW_i) \oplus UB_i$ $UC_i' = h(MUPW_i \| \| UA_i)$ Check $UC_i' \stackrel{?}{=} UC_i$ Generate R_u, T_1 $M_1 = MUPW_i \oplus UID_i \oplus UA_i \oplus R_u$ $CUID_i = h(MUID_i \| \| UA_i \| \| R_u)$ $\oplus (SUID_i \| \| UID_i)$ $K_{UG} = h(UA_i \| \| R_u \| \| UID_i \| \| T_1)$		
$\xrightarrow{\{M_1, MUID_i, CUID_i, K_{UG}, T_1\}}$		
	Checks T_1 $UA_i = h(MUID_i \| \| R_n \| \| K_G)$ $R_u = UA_i \oplus M_1 \oplus UID_i \oplus MUPW_i$ $(UID_i \| \| SUID_j) = CUID_i \oplus$ $h(MUID_i \| \| UA_i \| \| R_u)$ $K_{UG}' = h(UA_i \| \| R_u \| \| UID_i \| \| T_1)$ $K_{UG}' \stackrel{?}{=} K_{UG}$ Generate R_g, T_2 $M_2 = (R_u \| \| R_g) \oplus h(SUID_j \| \| UA_j \| \| T_2)$ $K_{GS} = h(MUID_i \| \| SUID_j \| \| R_u \| \| R_g \| \| UA_j \| \| T_2)$ $\xrightarrow{\{M_2, MUID_i, M_{GS}, T_2\}}$	
		Checks T_2 $(R_u \| \| R_g) = h(SUID_j \| \| UA_j \| \| T_2) \oplus M_2$ $K_{GS}' = h(MUID_i \| \| SUID_j \| \| R_u$ $\| \| R_g \| \| UA_j \| \| T_2)$ $K_{GS}' \stackrel{?}{=} K_{GS}$ Generate R_s, T_3 $N = ENC_{PU}(R_s)$ $K_{SG} = h(R_s \| \| R_g \| \| SUID_j \| \| UA_j \| \| T_3)$ $SK = h(R_u \| \| R_s)$ $K_{SU} = h(SK \| \| R_s \| \| R_u$ $\| \| SUID_j \| \| MUID_i)$ $\xleftarrow{\{N, K_{SG}, K_{SU}, T_3\}}$
	Checks T_3 $R_s = DES_{PR}(N)$ $K_{SG}' = h(R_s \| \| R_g \| \| SUID_j \| \| UA_j \| \| T_3)$ $K_{SG}' \stackrel{?}{=} K_{SG}$ Generate T_4 $MUID_i^{new} = h(UID_i \| \| h(K_G \| \| R_g))$ $UA_i^{new} = h(MUID_i^{new} \| \| R_g \| \| K_g)$ $M_4 = h(MUID_i \| \| UA_i \| \| T_4) \oplus$ $(MUID_i^{new} \| \| UA_i^{new} \| \| R_g)$ $K_{GU} = h(R_u \| \| R_g \| \| MUID_i \| \| UA_i \| \| T_4)$ $MUPW_i = DES_{PR}(S)$ $UC_i = h(MUPW_i \| \| UA_i)$ $M_{KU} = R_s \oplus UC_i \oplus UR_i$ $\xleftarrow{\{M_{KU}, M_4, K_{SU}, K_{GU}, T_4\}}$	
Checks T_4 $(MUID_i^{new} \| \| UA_i^{new} \| \| R_g)$ $= M_4 \oplus h(MUID_i \| \| UA_i \| \| T_4)$ $R_s = M_{KU} \oplus UC_i \oplus UR_i$ $K_{GU}' = h(R_u \| \| R_g \| \| MUID_i \| \| UA_i \| \| T_4)$ $K_{GU}' \stackrel{?}{=} K_{GU}$ $SK = h(R_u \| \| R_s)$ $K_{SU}' = h(SK \| \| R_s \| \| R_u \| \| SUID_j \| \| MUID_i)$ $K_{SU}' \stackrel{?}{=} K_{SU}$ $UB_i^{new} = h(MUID_i^{new} \| \| MUPW_i) \oplus UA_i^{new}$ $UC_i^{new} = h(MUPW_i \| \| UA_i^{new})$ Update $MUID_i^{new}, UB_i^{new}, UC_i^{new}$ to $MUID_i, UB_i, UC_i$		

Figure 4: Login and authentication phase

(3) S_j confirms the freshness of T_2 and computes $(R_u||R_g) = h(SUID_j||UA_j||T_2) \oplus M_2$, $K'_{GS} = h(MUID_i||SUID_j||R_u||R_g||UA_j||T_2)$. Now, S_j verifies the correctness K'_{GS}. Then, S_j generates R_s and T_3 and calculates $N = ENC_{PU}(R_s)$, $K_{SG} = h(R_s||R_g||SUID_j||UA_j||T_3)$, $SK = h(R_u||R_s)$ and $K_{SU} = h(SK||R_s||R_u||SUID_j||MUID_i)$. Eventually, S_j transfers $\{N, K_{SG}, K_{SU}, T_3\}$ to GWN.

(4) GWN confirms the freshness of T_3 and computes $R_s = DES_{PR}(N)$ and $K'_{SG} = h(R_s||R_g||SUID_j||UA_j||T_3)$. Then, GWN confirms the correctness of K'_{SG}. After that, GWN calculates $MUID_i^{new} = h(UID_i||h(K_G||R_g))$, $UA_i^{new} = h(MUID_i^{new}||R_g||K_g)$, $M_4 = h(MUID_i||UA_i||T_4) \oplus (MUID_i^{new}||UA_i^{new}||R_g)$, $K_{GU} = h(R_u||R_g||MUID_i||UA_i||T_4)$, $MUPW_i = DES_{PR}(S)$, $UC_i = h(MUPW_i||UA_i)$ and $M_{KU} = R_s \oplus UC_i \oplus UR_i$. Now, GWN sends $\{M_{KU}, M_4, K_{SU}, K_{GU}, T_4\}$ to U_i.

(5) U_i checks the freshness of T_4 and calculates $(MUID_i^{new}||UA_i^{new}||R_g) = M_4 \oplus h(MUID_i||UA_i||T_4)$, $R_s = M_{KU} \oplus UC_i \oplus UR_i$ and $K'_{GU} = h(R_u||R_g||MUID_i||UA_i||T_4)$. U_i then verifies the correctness of K'_{GU}. After that, U_i calculates $SK = h(R_u||R_s)$ and $K'_{SU} = h(SK||R_s||R_u||SUID_j||MUID_i)$, and then checks the correctness of K'_{SU}. Furthermore, U_i calculates $UB_i^{new} = h(MUID_i^{new}||MUPW_i) \oplus UA_i^{new}$ and $UC_i^{new} = h(MUPW_i||UA_i^{new})$, and then replaces $MUID_i, UB_i, UC_i$ with $MUID_i^{new}, UB_i^{new}, UC_i^{new}$.

Finally, U_i and S_j both have $SK = h(R_u||R_s)$ as a session key.

6 Security Analysis

This section demonstrates that PSAP-WSN is provably secure against different attacks, using BAN logic, ROR model, and ProVerif tool.

6.1 BAN Logic

Ban Logic Rules

Message-meaning rule (R1) $\dfrac{U \mid\equiv U \stackrel{K}{\longleftrightarrow} G, U \lhd \{M\}_K}{U \mid\equiv G \mid\sim M}$. $\dfrac{U \mid\equiv U \stackrel{N}{\rightleftharpoons} G, U \lhd \langle M\rangle_N}{U \mid\equiv G \mid\sim M}$.

Nonce-verification rule (R2) $\dfrac{U \mid\equiv \sharp(M), U \mid\equiv G \mid\sim M}{U \mid\equiv G \mid\equiv M}$.

Jurisdiction rule (R3) $\dfrac{U \mid\equiv G \mid\Rightarrow M, U \mid\equiv G \mid\equiv M}{U \mid\equiv M}$.

Freshness rule (R4) $\dfrac{U \mid\equiv \sharp(M)}{U \mid\equiv \sharp(M, N)}$.

Belief rule (R5) $\dfrac{U \mid\equiv M, U \mid\equiv N}{U \mid\equiv (M, N)}$.

Session key rule (R6) $\dfrac{U \mid\equiv \sharp(M), U \mid\equiv G \mid\equiv M}{U \mid\equiv U \stackrel{K}{\longleftrightarrow} G}$.

Goals

G1 $U \mid\equiv U \stackrel{SK}{\longleftrightarrow} G$.

G2 $G \mid\equiv U \stackrel{SK}{\longleftrightarrow} G$.

G3 $U \mid\equiv G \mid\equiv U \stackrel{SK}{\longleftrightarrow} G$.

G4 $G \mid\equiv U \mid\equiv U \xleftrightarrow{SK} G$.

G5 $S \mid\equiv S \xleftrightarrow{SK} G$.

G6 $G \mid\equiv S \xleftrightarrow{SK} G$.

G7 $S \mid\equiv G \mid\equiv S \xleftrightarrow{SK} G$.

G8 $G \mid\equiv S \mid\equiv S \xleftrightarrow{SK} G$.

6.1.1 Idealizing Communication

$M_{sg1} U \to G : \{M_1, MUID_i, CUID_i, K_{UG}, T_1\}$.

$M_{sg2} G \to S : \{M_2, MUID_i, K_{GS}, T_2\}$.

$M_{sg3} S \to G : \{N, K_{SG}, K_{SU}, T_3\}$.

$M_{sg4} G \to U : \{M_{KU}, M_4, K_{SU}, K_{GU}, T_4\}$.

Initial state assumptions

A1 $U \mid\equiv U \overset{UA_i}{\rightleftharpoons} G$.

A2 $G \mid\equiv U \overset{UA_i}{\rightleftharpoons} G$.

A3 $G \mid\equiv \sharp(R_u, R_s)$.

A4 $G \mid\equiv U \mid\Rightarrow (R_u)$.

A5 $G \mid\equiv \sharp(R_s)$.

A6 $U \mid\equiv \sharp(R_u, R_s)$.

A7 $U \mid\equiv G \mid\Rightarrow (R_u, R_s)$.

A8 $S \mid\equiv G \overset{UA_j, SUID_j}{\rightleftharpoons} S$.

A9 $G \mid\equiv S \overset{UA_j, SUID_j}{\rightleftharpoons} G$.

A10 $S \mid\equiv \sharp(R_u, R_s)$.

A11 $S \mid\equiv G \mid\Rightarrow (R_u, R_s)$.

Detailed steps

With M_{sg1} and using the seeing rule, we obtain

S1 : $G \triangleleft \{\langle R_u \rangle_{UA_i}, \langle MUID_i, CUID_i, K_{UG}, T_1 \rangle\}$

Using S1, R1, and A2, we obtain

S2 : $G \mid\equiv U \mid\sim (R_u)$

Using S2, under the assumption of A3 and nonce verification postulate R2, S3 can be obtained.

S3 : $G \mid\equiv U \mid\equiv (R_u)$

With A4, R3, and S3, we obtain

S4 : $G \mid\equiv (R_u)$

Similarly, we obtain

S5 : $G \mid\equiv (R_s)$

Because $SK = h(R_u||R_s)$, using S4 and S5, we obtain

S6 : $G \mid\equiv U \stackrel{SK}{\longleftrightarrow} G\ (G2)$.

With A3, A5, and R4, we obtain

S7 : $G \mid\equiv U \mid\equiv U \stackrel{SK}{\longleftrightarrow} G\ (G4)$.

In addition, using M_{sg4}, we obtain

S8 : $U \triangleleft \{\langle R_s\rangle_{UA_i}, M_{KU}, K_{SU}, M4, T_4\}$.

By using A1, and R1 we obtain

S9 : $U \mid\equiv G \mid\sim (R_s)$

With S9, A6, and R2, we obtain

S10 : $U \mid\equiv G \mid\equiv (R_s)$

Using A7, S9, and R3, we obtain

S11 : $U \mid\equiv (R_s)$.

thus,

S12 : $U \mid\equiv (R_u)$.

Because $SK = h(R_u||R_s)$, using S11 and S12, we obtain

S13 : $U \mid\equiv U \stackrel{SK}{\longleftrightarrow} G\ (G1)$

With S13, A6, and R4, we obtain

S14 : $U \mid\equiv G \mid\equiv U \stackrel{SK}{\longleftrightarrow} G\ (G3)$.

By considering the message M_{sg2}, we obtain

S15 : $S \triangleleft \{\langle R_u, R_s\rangle_{UA_j}, MUID_i, K_{GS}\}$

Using S15, R1, and A8, we obtain

S16 : $S \mid\equiv G \mid\sim (R_u, R_s)$

Using S16, under the assumption of A10 and the nonce verification postulate R2, S17 can be obtained.

S17 : $S \mid\equiv G \mid\equiv (R_u, R_s)$

Using A11, R3, and S17, we obtain

S18 : $S \mid\equiv (R_u, R_s)$

Because $SK = h(R_u||R_s)$, using S18, we obtain

S19 : $S \mid\equiv G \stackrel{SK}{\longleftrightarrow} S\ (G5)$

Using S19, A10, and R4, we obtain

S20 : $S \mid\equiv G \mid\equiv S \stackrel{SK}{\longleftrightarrow} G\ (G7)$.

By considering message M_{sg3}, we obtain

S21 : $G \triangleleft \{\langle R_u, R_s\rangle_{SUID_j}, K_{GU}\}$

Using S21, R1, and A9, we obtain

S22 : $G \mid\equiv S \mid\sim (R_u, R_s)$

Using S22, under the assumption of A3, A5, and nonce verification postulate R2, S23 can be obtained.

S23 : $G \mid\equiv U \mid\equiv (R_u, R_s)$

Using A4, R3, and S2, we obtain

S24 : $G \mid\equiv (R_u, R_s)$

Because $SK = h(R_u||R_s)$, using S24, we obtain

S25 : $G \mid\equiv S \stackrel{SK}{\longleftrightarrow} G\ (G6)$

Using S25, A3, A5, and R4, we obtain

S26 : $S \mid\equiv G \mid\equiv S \stackrel{SK}{\longleftrightarrow} G\ (G8)$.

6.2 ROR Model

The well-known real-or-random (ROR) model [44] was used to demonstrate that PSAP-WSN is provably secure. The ROR model has been widely used in numerous studies. The PSAP-WSN has three entities: U_i, GWN, and S_j. In the proof, we define $R = \{H_u^x, H_G^y, H_s^z\}$, where H_u^x, H_G^y, and H_s^z denote the *x-th* U_i, *y-th* GWN, and *z-th* S_j, respectively. In addition, $\mathcal{A}$ as an attacker can perform the following operations:

Execute(R): With *Execute*(R), $\mathcal{A}$ can obtain messages transmitted by U_i, GWN, and S_j through a public channel.

Send(R, M): $\mathcal{A}$ can receive or send messages transmitted between entities via *Send*(R, M).

Reveal(R): By performing *Reveal*(R), $\mathcal{A}$ can access the session key generated between various entities.

Hash$(String)$: Using *Hash*$(String)$, $\mathcal{A}$ can calculate the hash value of a fixed string.

Test(O): During the execution of the game, it is necessary to flip coin C to determine the probability that $\mathcal{A}$ can obtain SK. If C equals 1, the correct painting key is obtained; if it equals 0, a string with the same length as the painting key is obtained.

Theorem 1: Using $Adv_P^{\mathcal{A}}$ as the main function for $\mathcal{A}$ the SK between the communicators is obtained. q_h and q_s represent the number of *Hash* and *Send* queries, respectively, and H and B represent the range that can be accommodated by the hash function and the space size of the user password dictionary. The advantage of using a function to crack SK is that $Adv_P^{\mathcal{A}} \leq q_h^2/|H| + 2q_s/|B|$.

Security proof

Proof: To prove Theorem 1, four games $Game_i(i = 0, 1, 2, 3)$ were created. Among them, the $\mathcal{A}$ that wins the game can be identified as $Adv_{Game_i}^{\mathcal{A}}$, and the probability of $\mathcal{A}$ winning the game is $Pr[Adv_{Game_i}^{\mathcal{A}}]$.

$Game_0$: In the first game, $\mathcal{A}$ does not perform any operation except for selecting bit b; therefore, the result of $\mathcal{A}$ against the protocol is $Adv_P^{\mathcal{A}} = |2Adv_{Game_0}^{\mathcal{A}} - 1|$.

$Game_1$: In the second game, $\mathcal{A}$ performs the eavesdropping operations. $\mathcal{A}$ can intercept and eavesdrop on the information $\{M_1, MUID_i, CUID_i, K_{UG}, T_1\}$ and $\{N, K_{SG}, K_{SU}, T_3\}$ transmitted between communicators through a public channel. However, if $\mathcal{A}$ wants to obtain SK between the two communication parties by executing the *Test* operation, it must also know the random numbers R_u and R_s because $SK = h(R_u||R_s)$. Therefore, even if $\mathcal{A}$ executes the *Execute* operation, the probability of obtaining the session key is the same as in $Game_0$. Hence, $Pr[Adv_{Game_0}^{\mathcal{A}}] = Pr[Adv_{Game_1}^{\mathcal{A}}]$.

$Game_2$: The *Send* operation and *Hash* query were added to the previous game. During the execution of the game, we found that M_2, K_{UG}, and K_{SG} were protected by a hash function. If $\mathcal{A}$ wants to obtain SK, $\mathcal{A}$ must crack the hash function; however, $\mathcal{A}$ cannot successfully crack the hash function because of the collision of the hash function. Thus, a conclusion can be drawn from the birthday paradox $Pr[Adv_{Game_2}^{\mathcal{A}} - Adv_{Game_1}^{\mathcal{A}}] \leq q_h^2/2|H|$.

GM_3: During the operation of this game, $\mathcal{A}$ attempts to estimate UID_i. In addition, $\mathcal{A}$ cracked SK between U_i and S_j by intercepting the messages transmitted by the communicator through a public channel. However, random number R_u can only be obtained using U_i's password, because $R_u = UA_i \oplus M_1 \oplus UID_i \oplus MUPW_i$. In the proposed protocol, $\mathcal{A}$ can only send a limited number of send requests to crack SK. Thus,

$Pr[Adv_{Game_3}^{\mathcal{A}} - Adv_{Game_2}^{\mathcal{A}}] \leq q_s/|B|$.

After executing the above four games, $\mathcal{A}$ can only win the game by guessing the correct bit B; thus,

$Pr[Adv_{Game_3}^{\mathcal{A}}] = 1/2$.

By sorting the above formulae, we obtain

$$\begin{aligned}
1/2Adv_P^{\mathcal{A}} &= |Adv_{Game_0}^{\mathcal{A}} - 1/2| \\
&= |Pr[Adv_{Game_1}^{\mathcal{A}}] - Pr[Adv_{Game_3}^{\mathcal{A}}]| \\
&\leq |Pr[Adv_{Game_1}^{\mathcal{A}}] - Pr[Adv_{Game_2}^{\mathcal{A}}]| + |Pr[Adv_{Game_2}^{\mathcal{A}}] - Pr[Adv_{Game_3}^{\mathcal{A}}]| \\
&= q_h^2/2|H| + q_s/|B|
\end{aligned}$$

Subsequently, we obtain

$Adv_P^{\mathcal{A}} \leq q_h^2/|H| + 2q_s/|B|$. Therefore, it is proven that Theorem 1 is valid.

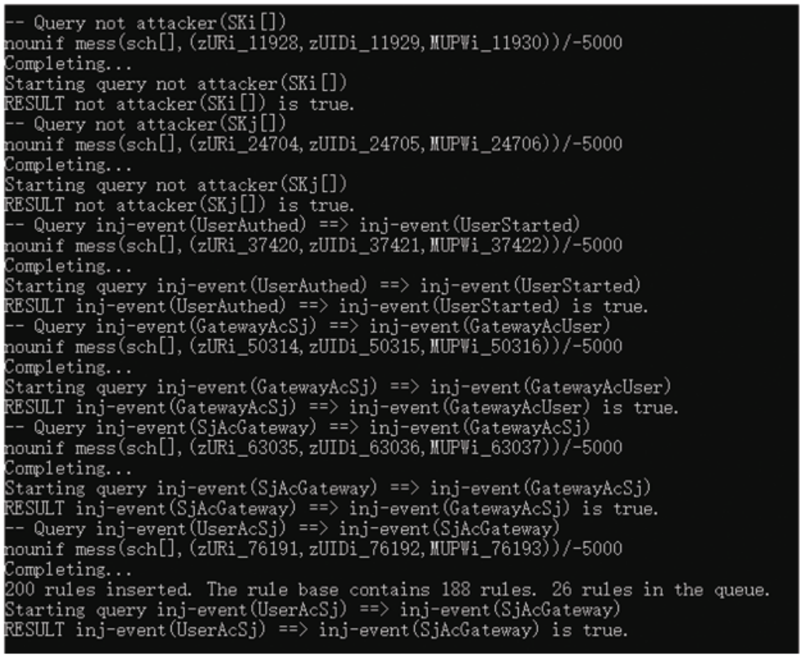
6.3 *ProVerif*

To further verify the security of the proposed PSAP-WSN, a well-known verification tool called ProVerif [45,46] was used. In this simulation, we define *ch* as a public channel and *sch* as a secure channel. *SKi* and *SKj* represent the session keys established by the user and the sensor node, respectively. In addition, *PR* and *KG* represent the gateway's private and master keys, respectively. The simulation contained five events: UserStarted(), UserAuthed(), GatewayAcUser(), SjAcGateway(), and UserAcSj(). The defined parameters and function codes are presented in detail in Fig. 5.

```
(*  channel*)
free ch :channel. (*  public channel *)
free sch: channel [private]. (*  secure channel, used for registering *)
(*  shared keys *)
free SKi : bitstring  [private].
free SKj : bitstring  [private].
(* constants *)
free PR:bitstring [private].(* the GWN's secret key *)
free KG:bitstring [private].
(*  functions  &  reductions & equations  *)
fun h(bitstring) :bitstring. (*  hash function *)
fun mult(bitstring,bitstring) :bitstring. (* scalar multiplication operation *)
fun add(bitstring,bitstring):bitstring. (* Addition operation *)
fun sub(bitstring,bitstring):bitstring. (* Subtraction operation *)
fun mod(bitstring,bitstring):bitstring. (* modulus operation *)
fun con(bitstring,bitstring):bitstring. (* concatenation operation *)
reduc forall m:bitstring, n:bitstring; getmess(con(m,n))=m.
fun senc(bitstring,bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring. (*  XOR operation *)
equation forall m:bitstring, n:bitstring; xor(xor(m,n),n)=m.
fun Gen(bitstring):bitstring. (* Generator operation *)
fun Rep(bitstring,bitstring):bitstring.
(*  queries  *)
query attacker(SKi).
query attacker(SKj).
query inj-event(UserAuthed()) ==> inj-event(UserStarted()).
query inj-event(GatewayAcSj()) ==> inj-event(GatewayAcUser()).
query inj-event(SjAcGateway()) ==> inj-event(GatewayAcSj()).
query inj-event(UserAcSj()) ==> inj-event(SjAcGateway()).
(* event *)
event UserStarted().
event UserAuthed().
event GatewayAcUser().
event GatewayAcSj().
event SjAcGateway().
event UserAcSj().
```

Figure 5: Definition, queries, and events in the ProVerif tool

The results for ProVerif are shown in Fig. 6. We can see "Result not attacker (ski []) is true," "RESULT not attacker(SKj[]) is true," "RESULT inj-event(UserAuthed) ==> inj-event(UserStarted) is true," "RESULT inj-event(GatewayAcSj) ==> inj-event(GatewayAcUser) is true," "RESULT inj-event(Sj-"AcGateway) ==> inj-event(GatewayAcSj) is true," and "RESULT inj-event(UserAcSj) ==> inj-event (SjAcGateway) is true." The results show that PSAP-WSN can pass the Proverif tool.



```
-- Query not attacker(SKi[])
nounif mess(sch[],(zURi_11928,zUIDi_11929,MUPWi_11930))/-5000
Completing...
Starting query not attacker(SKi[])
RESULT not attacker(SKi[]) is true.
-- Query not attacker(SKj[])
nounif mess(sch[],(zURi_24704,zUIDi_24705,MUPWi_24706))/-5000
Completing...
Starting query not attacker(SKj[])
RESULT not attacker(SKj[]) is true.
-- Query inj-event(UserAuthed) ==> inj-event(UserStarted)
nounif mess(sch[],(zURi_37420,zUIDi_37421,MUPWi_37422))/-5000
Completing...
Starting query inj-event(UserAuthed) ==> inj-event(UserStarted)
RESULT inj-event(UserAuthed) ==> inj-event(UserStarted) is true.
-- Query inj-event(GatewayAcSj) ==> inj-event(GatewayAcUser)
nounif mess(sch[],(zURi_50314,zUIDi_50315,MUPWi_50316))/-5000
Completing...
Starting query inj-event(GatewayAcSj) ==> inj-event(GatewayAcUser)
RESULT inj-event(GatewayAcSj) ==> inj-event(GatewayAcUser) is true.
-- Query inj-event(SjAcGateway) ==> inj-event(GatewayAcSj)
nounif mess(sch[],(zURi_63035,zUIDi_63036,MUPWi_63037))/-5000
Completing...
Starting query inj-event(SjAcGateway) ==> inj-event(GatewayAcSj)
RESULT inj-event(SjAcGateway) ==> inj-event(GatewayAcSj) is true.
-- Query inj-event(UserAcSj) ==> inj-event(SjAcGateway)
nounif mess(sch[],(zURi_76191,zUIDi_76192,MUPWi_76193))/-5000
Completing...
200 rules inserted. The rule base contains 188 rules. 26 rules in the queue.
Starting query inj-event(UserAcSj) ==> inj-event(SjAcGateway)
RESULT inj-event(UserAcSj) ==> inj-event(SjAcGateway) is true.
```

Figure 6: Operation results

6.4 Security Requirement Analysis

Next, it is demonstrated that PSAP-WSN is secure against the following attacks.

6.4.1 Sensor Node Capture Attack

Because a sensor node is unattended, it is easily obtained by $\mathcal{A}$ to analyze the internal parameters. Assume $\mathcal{A}$ obtains $SUID_j$ and UA_j after capturing S_j. However, to obtain SK, $\mathcal{A}$ must know R_u and R_s simultaneously. R_u can be obtained through $(R_u||R_g) = h(SUID_j||UA_i||T_2) \oplus M_2$, where T_2 and M_2 are submitted via a public channel. Unfortunately, R_s is a temporary random number; therefore, the PSAP-WSN can resist this attack.

6.4.2 Temporary Information Disclosure Attack

This attack assumes that $\mathcal{A}$ can obtain a random number in PSAP-WSN if R_u is leaked, but UA_i and UID_i are not obtained. Only $UID_i \oplus UA_i$ can be acquired, but other operations cannot be further performed. If R_g is leaked, but other parameters have not been analyzed, $\mathcal{A}$ cannot carry out the next operation. Thus, the PSAP-WSN can resist this type of attack.

6.4.3 Impersonation Attack

$\mathcal{A}$ can impersonate a user to send messages to GWN, but $\mathcal{A}$ cannot generate a request message $M_1, MUID_i, CUID_i, K_{UG}$. This is because $\mathcal{A}$ cannot obtain the user identity, biometrics, and random numbers; thus, PSAP-WSN can resist this attack.

6.4.4 Replay Attack

Suppose $\mathcal{A}$ performs a replay attack. However, when $\mathcal{A}$ attempts to send a request M_1, $MUID_i$, $CUID_i$, K_{UG}, T_1, GWN verifies the freshness of the timestamp T_1. Simultaneously, PSAP-WSN uses UA_i, R_u, and UID_i to hash T_1. For these reasons, it is concluded that PSAP-WSN can resist this attack.

6.4.5 Anonymity and Untraceability

In our design, neither UID_i is transferred, nor are there any devices to store UID_i. In addition, one-way hash function processing is performed for the places where UID_i is required; therefore, $\mathcal{A}$ cannot analyze UID_i in various ways. The user parameters $MUID_i$, UB_i, UC_i are updated after each authentication round. $\mathcal{A}$ cannot use the current information to infer previously transmitted information and cannot track the user; therefore, the proposed protocol can ensure anonymity and untraceability.

6.5 Security Comparisons

The proposed PSAP-WSN was compared with similar protocols. The primary attacks included A1: sensor node capture attack; A2: privileged insider attack; A3: temporary information disclosure attack; A4: impersonation attack; A5: replay attack; and A6: anonymity and untraceability attacks. The results in Table 3 confirm that PSAP-WSN provides sufficient security advantages compared with other protocols.

Table 3: Communication overhead comparison

Protocols	A1	A2	A3	A4	A5	A6
Ours	√	√	√	√	√	√
Wu et al. [47]	√	√	√	×	×	√
Wang et al. [48]	√	×	√	√	√	√
Li et al. [49]	√	√	√	×	×	√
Li et al. [50]	√	×	√	×	×	√
Lu et al. [32]	√	√	√	×	√	√
Mo et al. [33]	√	√	√	×	√	×
Yu et al. [26]	×	×	×	√	√	√
Almuhaideb et al. [34]	√	√	√	√	√	√

7 Performance Evaluation

This section evaluates the performance by experimentally calculating the computation and communication overhead.

7.1 Computation Comparisons

The three different types of devices used in the comparisons included the OPPO-R9 mobile phone, MI10-UTAR mobile phone, and ASUS-A456U notebook to represent the user, gateway, and sensor, respectively. The running times of the different functions for each device are listed in Table 4. In our experiment, the running times of symmetric encryption and asymmetric encryption were almost the same. In the experiment mentioned in [47], the running time of T_R (rep operation) is nearly equal to T_m. Therefore, this setting was adopted in our experiment.

Table 4: Running time on different devices

Operation	OPPO-R9	MI10-UTAR	ASUS-A456U
Point multiplication execution (T_m)	15 ms	14 ms	27 ms
Symmetric encryption/decryption (T_s)	0.19 ms	0.1 ms	0.19 ms
Asymmetric encryption/decryption (T_{as})	0.1 ms	0.05 ms	0.09 ms
Hash (T_h)	0.005 ms	0.0025 ms	0.004 ms

The experimental results are presented in Table 5. As shown in the Table 5, the running times of the user, gateway, and sensor node were 15.055, 0.0825, and 0.11 ms, respectively. Although the running time of our design was not always optimal, the overall ranking was relatively high. In addition, the difference was also quite small. Most importantly, these protocols have better running times and are vulnerable to attacks. The results are illustrated in Fig. 7.

Table 5: Computational cost of the proposed protocol

Protocols	User	Gateway	Sensor Node
Ours	$11T_h + T_R$ (15.055 ms)	$13T_h + T_{as}$ (0.0825 ms)	$5T_h + T_{as}$ (0.11 ms)
Wu et al. [47]	$11T_h + T_R + 2T_m$ (40.055 ms)	$10T_h$ (0.025 ms)	$3T_h + 2T_m$ (54.012 ms)
Wang et al. [48]	$10T_h + T_R + 3T_m$ (60.05 ms)	$13T_h + T_m$ (14.0325 ms)	$6T_h + 2T_m$ (54.024 ms)
Li et al. [49]	$8T_h + T_R + 2T_m$ (45.04 ms)	$9T_h + T_m$ (14.0225 ms)	$4T_h$ (0.016 ms)
Li et al. [50]	$12T_h + 3T_m$ (45.06 ms)	$8T_h + T_m$ (14.02 ms)	$4T_h + 2T_m$ (54.016 ms)
Lu et al. [32]	$7T_h + T_R + 3T_m + T_s$ (60.225 ms)	$6T_h + T_m + T_s$ (14.115 ms)	$2T_h + 2T_m + 2T_s$ (54.388 ms)
Mo et al. [33]	$12T_h + T_R + 2T_m$ (45.06 ms)	$10T_h + T_s$ (0.125 ms)	$5T_h + 2T_m + T_s$ (54.21 ms)
Yu et al. [26]	$11T_h + T_R$ (15.055 ms)	$11T_h$ (0.0275 ms)	$6T_h$ (0.024 ms)

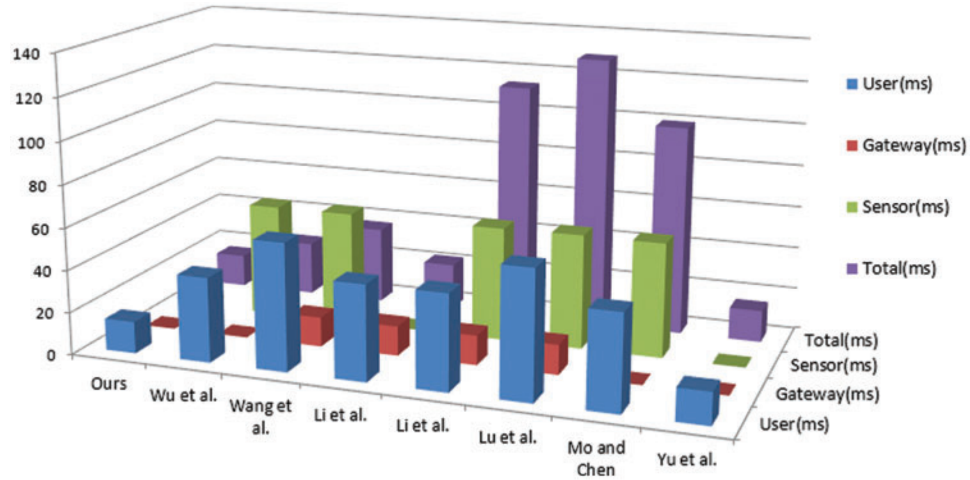


Figure 7: Running time

7.2 Communication Comparisons

Here, to discuss the communication overhead, the proposed protocol is compared with other related protocols. In the experiment, the settings in [26] were adopted, thereby assuming that the prime number, random nonce, identity, timestamp, and hash function are 160, 128, 32, 32, and 160 bits, respectively. The information exchanged in our proposed protocol includes, $M_1, MUID_i, CUID_i, K_{UG}, T_1$, $M_2, MUID_i, M_{GS}, T_2, N, K_{SG}, K_{SU}, T_3$, and $M_{KU}, M_4, K_{SU}, K_{GU}, T_4$, respectively, denoted by $(160+160+160+160+32=672$ bits), $(160+160+160+32=672$ bits), $(128+160+160+32=480$ bits), $(160+160+160+32=672$ bits). Table 6 lists the overhead for each protocol. It is observed that our design is not the best in terms of communication overhead, but the differences are not significant. However, the proposed method provides better security than these other protocols.

Table 6: Communication overhead comparison

Protocols	Communication overhead
Ours	2496 bits
Wu et al. [47]	3072 bits
Wang et al. [48]	2368 bits
Li et al. [49]	2496 bits
Li et al. [50]	2880 bits
Lu et al. [32]	2880 bits
Mo et al. [33]	3328 bits
Yu et al. [26]	2208 bits

8 Conclusions

In this paper, first, Yu et al.'s protocol was reviewed and cryptanalyzed, thereby determining that it is vulnerable to sensor node capture attacks and temporary information disclosure attacks. Therefore, the PSAP-WSN protocol was proposed. Subsequently, PSAP-WSN was demonstrated to be provably secure, using BAN logic, the ROR model, and the Proverif tool. In addition, an adversarial attack was simulated against the proposed PSAP-WSN. The performance evaluation indicates that the PSAP-WSN has reasonable communication and computation overhead and is suitable for WSNs.

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

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Khan, M. A., Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems, 82,* 395–411. DOI 10.1016/j.future.2017.11.022.
2. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29(7),* 1645–1660. DOI 10.1016/j.future.2013.01.010.
3. Huang, X., Xiong, H., Chen, J., Yang, M. (2021). Efficient revocable storage attribute-based encryption with arithmetic span programs in cloud-assisted Internet of Things. *IEEE Transactions on Cloud Computing*. DOI 10.1109/TCC.2021.3131686.
4. Ashton, K. (2009). That 'Internet of Things' thing. *RFID Journal, 22(7),* 97–114.
5. Chettri, L., Bera, R. (2019). A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet of Things Journal, 7(1),* 16–32. DOI 10.1109/JIoT.6488907.
6. Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., Mustaqim, M. (2020). Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5 G-IoT scenarios. *IEEE Access, 8,* 23022–23040. DOI 10.1109/Access.6287639.
7. Khan, M. A., Alzahrani, B. A., Barnawi, A., Al-Barakati, A., Irshad, A. et al. (2022). A resource friendly authentication scheme for space–air–ground–sea integrated maritime communication network. *Ocean Engineering, 250,* 110894. DOI 10.1016/j.oceaneng.2022.110894.
8. Chaudhry, S. A., Irshad, A., Nebhen, J., Bashir, A. K., Moustafa, N. et al. (2021). An anonymous device to device access control based on secure certificate for internet of medical things systems. *Sustainable Cities and Society, 75,* 103322. DOI 10.1016/j.scs.2021.103322.
9. Xiong, H., Chen, J., Mei, Q., Zhao, Y. (2020). Conditional privacy-preserving authentication protocol with dynamic membership updating for vanets. *IEEE Transactions on Dependable and Secure Computing, 19(3),* 2089–2104. DOI 10.1109/TDSC.2020.3047872.
10. Khan, M. A., Ullah, I., Alkhalifah, A., Rehman, S. U., Shah, J. A. et al. (2021). A provable and privacy-preserving authentication scheme for UAV-enabled intelligent transportation systems. *IEEE Transactions on Industrial Informatics, 18(5),* 3416–3425.
11. Chu, S. C., Dao, T. K., Pan, J. S., Nguyen, T. T. (2020). Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification. *EURASIP Journal on Wireless Communications and Networking, 2020(1),* 52.
12. Fan, F., Chu, S. C., Pan, J. S., Lin, C., Zhao, H. (2021). An optimized machine learning technology scheme and its application in fault detection in wireless sensor networks. *Journal of Applied Statistics*. DOI 10.1080/02664763.2021.1929089.

13. Xue, X., Chen, J. (2019). Using compact evolutionary tabu search algorithm for matching sensor ontologies. *Swarm and Evolutionary Computation, 48,* 25–30. DOI 10.1016/j.swevo.2019.03.007.
14. Reddy, G. T., Kaluri, R., Reddy, P. K., Lakshmanna, K., Koppu, S. et al. (2019). A novel approach for home surveillance system using IoT adaptive security. *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur-India. DOI 10.2139/ssrn.3356525.
15. Kumar, S. A., Vealey, T., Srivastava, H. (2016). Security in Internet of Things: Challenges, solutions and future directions. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 5772–5781. Koloa, HI, USA.
16. Jian, M. S., Wu, J. M. T. (2021). Hybrid Internet of Things (IoT) data transmission security corresponding to device verification. *Journal of Ambient Intelligence and Humanized Computing*.
17. Abidoye, A. P., Obagbuwa, I. C. (2018). DDOS attacks in WSNS: Detection and countermeasures. *IET Wireless Sensor Systems, 8(2),* 52–59. DOI 10.1049/iet-wss.2017.0029.
18. Kaushal, K., Sahni, V. (2016). Early detection of DDOS attack in WSN. *International Journal of Computer Applications, 134(13),* 14–18. DOI 10.5120/ijca2016908117.
19. Soni, P., Pal, A. K., Islam, S. H. (2019). An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system. *Computer Methods and Programs in Biomedicine, 182,* 105054. DOI 10.1016/j.cmpb.2019.105054.
20. Modares, H., Salleh, R., Moravejosharieh, A. (2011). Overview of security issues in wireless sensor networks. *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*, pp. 308–311. Langkawi, Malaysia.
21. Chen, C. M., Li, Z., Chaudhry, S. A., Li, L. (2021). Attacks and solutions for a two-factor authentication protocol for wireless body area networks. *Security and Communication Networks, 2021,* 3116593. DOI 10.1155/2021/3116593.
22. Azrour, M., Mabrouki, J., Guezzaz, A., Farhaoui, Y. (2021). New enhanced authentication protocol for Internet of Things. *Big Data Mining and Analytics, 4(1),* 1–9. DOI 10.26599/BDMA.2020.9020010.
23. Wu, T. Y., Yang, L., Lee, Z., Chu, S. C., Kumari, S. et al. (2021). A provably secure three-factor authentication protocol for wireless sensor networks. *Wireless Communications and Mobile Computing, 2021,* 5537018. DOI 10.1155/2021/5537018.
24. Shafiq, A., Ayub, M. F., Mahmood, K., Sadiq, M., Kumari, S. et al. (2020). An identity-based anonymous three-party authenticated protocol for IoT infrastructure. *Journal of Sensors, 2020,* 1–17.
25. Chen, C. M., Xiang, B., Wang, K. H., Yeh, K. H., Wu, T. Y. (2018). A robust mutual authentication with a key agreement scheme for session initiation protocol. *Applied Sciences, 8(10),* 1789. DOI 10.3390/app8101789.
26. Yu, S., Park, Y. (2020). SLUA-WSN: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks. *Sensors, 20(15),* 4143. DOI 10.3390/s20154143.
27. Lu, R., Zhang, L., Ni, J., Fang, Y. (2019). 5G vehicle-to-everything services: Gearing up for security and privacy. *Proceedings of the IEEE, 108(2),* 373–389.
28. Liu, Y., Peng, J., Kang, J., Iliyasu, A. M., Niyato, D. et al. (2020). A secure federated learning framework for 5G networks. *IEEE Wireless Communications, 27(4),* 24–31. DOI 10.1109/MWC.7742.
29. Afaq, A., Haider, N., Baig, M. Z., Khan, K. S., Imran, M. et al. (2021). Machine learning for 5G security: Architecture, recent advances, and challenges. *Ad Hoc Networks, 123,* 102667. DOI 10.1016/j.adhoc.2021.102667.
30. Yahaya, A. S., Javaid, N., Ullah, S., Khalid, R., Javed, M. U. et al. (2022). A secure and efficient energy trading model using blockchain for a 5G-deployed smart community. *Wireless Communications and Mobile Computing, 2022,* 1–27.

31. Chang, I. P., Lee, T. F., Lin, T. H., Liu, C. M. (2015). Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors, 15(12),* 29841–29854. DOI 10.3390/s151229767.
32. Lu, Y., Xu, G., Li, L., Yang, Y. (2019). Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wireless Networks, 25(4),* 1461–1475. DOI 10.1007/s11276-017-1604-0.
33. Mo, J., Chen, H. (2019). A lightweight secure user authentication and key agreement protocol for wireless sensor networks. *Security and Communication Networks, 2019,* 2136506. DOI 10.1155/2019/2136506.
34. Almuhaideb, A. M., Alqudaihi, K. S. (2020). A lightweight three-factor authentication scheme for whsn architecture. *Sensors, 20(23),* 6860. DOI 10.3390/s20236860.
35. Dolev, D., Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory, 29(2),* 198–208. DOI 10.1109/TIT.1983.1056650.
36. Chen, C. M., Liu, S. (2021). Improved secure and lightweight authentication scheme for next-generation IoT infrastructure. *Security and Communication Networks, 2021,* 6537678. DOI 10.1155/2021/6537678.
37. Kocher, P., Jaffe, J., Jun, B. (1999). Differential power analysis. *Annual International Cryptology Conference*, pp. 388–397. Santa Barbara, California, USA.
38. Agadakos, I., Chen, C. Y., Campanelli, M., Anantharaman, P., Hasan, M. et al. (2017). Jumping the air gap: Modeling cyber-physical attack paths in the Internet-of-Things. *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, pp. 37–48. Dallas, Texas, USA.
39. Jokhio, S. H., Jokhio, I. A., Kemp, A. H. (2012). Node capture attack detection and defence in wireless sensor networks. *IET Wireless Sensor Systems, 2(3),* 161–169. DOI 10.1049/iet-wss.2011.0064.
40. Bharathi, M. V., Tanguturi, R. C., Jayakumar, C., Selvamani, K. (2012). Node capture attack in wireless sensor network: A survey. *2012 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–3. Tamilnadu, India.
41. Wang, C., Wang, D., Tu, Y., Xu, G., Wang, H. (2020). Understanding node capture attacks in user authentication schemes for wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing, 19(1),* 507–523.
42. Jiang, Q., Zeadally, S., Ma, J., He, D. (2017). Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access, 5,* 3376–3392. DOI 10.1109/ACCESS.2017.2673239.
43. Abbasinezhad-Mood, D., Ostad-Sharif, A., Nikooghadam, M., Mazinani, S. M. (2019). A secure and efficient key establishment scheme for communications of smart meters and service providers in smart grid. *IEEE Transactions on Industrial Informatics, 16(3),* 1495–1502. DOI 10.1109/TII.9424.
44. Abdalla, M., Fouque, P. A., Pointcheval, D. (2005). Password-based authenticated key exchange in the three-party setting. *International Workshop on Public Key Cryptography*, Les Diablerets, Switzerland: Springer.
45. Blanchet, B. (2013). Automatic verification of security protocols in the symbolic model: The verifier proverif. *Foundations of Security Analysis and Design VII*, pp. 54–87. Bertinoro, Italy, Springer.
46. Cheval, V., Cortier, V., Turuani, M. (2018). A little more conversation, a little less action, a lot more satisfaction: Global states in proVerif. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 344–358. Oxford, UK.
47. Wu, F., Xu, L., Kumari, S., Li, X. (2018). An improved and provably secure three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Networking and Applications, 11(1),* 1–20. DOI 10.1007/s12083-016-0485-9.
48. Wang, C., Xu, G., Sun, J. (2017). An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks. *Sensors, 17(12),* 2946. DOI 10.3390/s17122946.

49. Li, X., Niu, J., Kumari, S., Wu, F., Sangaiah, A. K. et al. (2018). A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments. *Journal of Network and Computer Applications, 103,* 194–204. DOI 10.1016/j.jnca.2017.07.001.
50. Li, X., Peng, J., Obaidat, M. S., Wu, F., Khan, M. K. et al. (2019). A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Systems Journal, 14(1),* 39–50. DOI 10.1109/JSYST.4267003.