Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 5290579, 17 pages
https://doi.org/10.1155/2017/5290579

Research Article

A Privacy Protection User Authentication and Key Agreement Scheme Tailored for the Internet of Things Environment: PriAuth

Yuwen Chen, José-Fernán Martínez, Pedro Castillejo, and Lourdes López

Departamento de Ingeniería Telemática y Electrónica (DTE), Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST), Universidad Politécnica de Madrid (UPM), C/Nikola Tesla, s/n, 28031 Madrid, Spain

Correspondence should be addressed to Yuwen Chen; yuwen.chen@upm.es

Received 6 July 2017; Revised 29 October 2017; Accepted 7 November 2017; Published 24 December 2017

Academic Editor: Anton Kos

In a wearable sensor-based deployment, sensors are placed over the patient to monitor their body health parameters. Continuous physiological information monitored by wearable sensors helps doctors have a better diagnostic and a suitable treatment. When doctors want to access the patient's sensor data remotely via network, the patient will authenticate the identity of the doctor first, and then they will negotiate a key for further communication. Many lightweight schemes have been proposed to enable a mutual authentication and key establishment between the two parties with the help of a gateway node, but most of these schemes cannot enable identity confidentiality. Besides, the shared key is also known by the gateway, which means the patient's sensor data could be leaked to the gateway. In PriAuth, identities are encrypted to guarantee confidentiality. Additionally, Elliptic Curve Diffie–Hellman (ECDH) key exchange protocol has been adopted to ensure the secrecy of the key, avoiding the gateway access to it. Besides, only hash and XOR computations are adopted because of the computability and power constraints of the wearable sensors. The proposed scheme has been validated by BAN logic and AVISPA, and the results show the scheme has been proven as secure.

1. Introduction

As sensors become widespread in their usage regarding health monitoring scenarios, a significant amount of personal sensitive data like blood pressure, pulse, or electrocardiogram readings will be monitored. These sensors could be interconnected to compose a Wireless Body Area Network (WBAN). With different sensors gathering patient's data and continually sending these data to doctors or to a remote monitoring station for further analysis, it is necessary to make sure that these data are transferred confidentially. The usual way is to encrypt them first before they are sent. The proposal presented in this paper, named PriAuth, aims to help the patient and the doctor build a shared key for encrypting health parameters.

Because only appointed doctors are allowed to access the patient's data, the patient and the doctor have to authenticate each other first. A workable way is to introduce a gateway to help the patient authenticating the legitimacy of the doctor and vice versa. After authentication, the two parties will build a shared key for further communication.

When a doctor wants to read patient's data, he sends a request to the patient. The patient forwards this request together with his own identification information to the gateway. The gateway checks whether the patient and the doctor are legitimate, and if any of them is not regarded as such then the scheme is aborted. Only when they are all legitimate, the gateway sends the authentication result to the patient. Once the patient has become aware of the legitimacy of the doctor, he sends the authentication result to the doctor as well. Based on the authentication result, the patient and the doctor can build a shared key, which is used for encrypting confidential information sent between them.

There are many research results focusing on the authentication and key agreement problems; while most of them could ensure the safety of the data, this is not enough, as there is also a need to protect privacy.

In the authentication process, the patient and the doctor have to send their identities and some other related information to the gateway. It has to be ensured that the patient's identity should not be leaked. Of course, a patient is usually unwilling to leak his identity information, because if the patient's identity is leaked, the health history and status of the patient will be freely available for anyone in the system, regardless of the patient wishes.

On the other hand, when a doctor sends his identity to the gateway for authentication, we have to make sure that the doctor's identity is kept confidential, too (e.g., when an adversary eavesdrops the identity of the doctor and finds out the doctor's major is dermatology according to the identity of the doctor, there is a great chance that the patient has a skin related problem). Therefore, it is also necessary to keep the doctor's identity confidential in order to protect the privacy of the patient. In PriAuth, Elliptic Curve Cryptography (ECC) is adopted as the method used to protect the identities of the data transmission participants, which is similar to [15–21].

After the gateway finishes the authentication process, the gateway will send the authentication result to the patient and the doctor. Based on the authentication result, the patient and the doctor could build a shared key. In some traditional schemes, the gateway could learn the key shared from the authentication information it gets from the patient and the doctor. This means the patient's personal health data could be leaked to the gateway. It is necessary to prevent the gateway learning this key. In PriAuth, Elliptic Curve Diffie–Hellman (ECDH) key exchange protocol is adopted to ensure the shared key secrecy between the patient and doctor. Besides, only hash and XOR operations are adopted, which is suitable for the wearable sensors.

PriAuth has been validated by BAN logic and AVISPA. BAN logic is one of the most prevalent methods that help determine whether the exchanged information is trustworthy, secure against eavesdropping. BAN logic is also adopted to prove the security of the schemes by [22–24]. AVISPA (Automated Validation of Internet Security Protocols and Applications) is a tool for the automated validation of Internet security-sensitive protocols and applications, which has been widely adopted by [24–26], and so forth.

This paper is organized as follows: Section 2 is related works; Section 3 is the preliminary knowledge. In Section 4, we introduce PriAuth; Section 5 provides the BAN logic validation. Section 6 includes AVISPA verification. Section 7 is the security analysis part. Section 8 provides a comparison with other schemes. Section 9 is the validation part. Section 10 concludes with a summary of the contributions.

2. Related Works

In several papers of the researched literature, the authors use different acronyms; user and sensor are the most commonly used, which equals to doctor and sensor in our scheme. Thus, from now on, we will use user and sensor instead of doctor and patient. D. Wang and P. Wang provide overviews of some of the schemes described in [27, 28]. Farash et al. use a single shared key between all the users or sensors to encrypt the identities [13]. All the sensors use the same key $h(X_{\mathrm{GWN}} \parallel 1)$ to encrypt the sensor identity, using XOR method where SID_j is the sensor identity and T_2 is a timestamp.

$$\mathrm{ESID}_j = \mathrm{SID}_j \oplus h\left(h\left(X_{\mathrm{GWN}} \parallel 1\right) \parallel T_2\right), \tag{1}$$

where $h(X_{\mathrm{GWN}} \parallel 1)$ is a key that is shared by all the sensors, so malicious or curious sensors could learn the identity of sensor SID_j. As ESID_j, T_2 are sent via a public channel. A malicious or curious sensor with identity SID_k can eavesdrop sensor SID_j to get ESID_j, T_2. In order to get the sensor id SID_j, SID_k could decrypt ESID_j using the same key $h(X_{\mathrm{GWN}} \parallel 1)$:

$$\begin{aligned} &\mathrm{ESID}_j \oplus h\left(h\left(X_{\mathrm{GWN}} \parallel 1\right) \parallel T_2\right) \\ &\quad = \left\{\mathrm{SID}_j \oplus h\left(h\left(X_{\mathrm{GWN}} \parallel 1\right) \parallel T_2\right)\right\} \\ &\qquad \oplus h\left(h\left(X_{\mathrm{GWN}} \parallel 1\right) \parallel T_2\right) = \mathrm{SID}_j. \end{aligned} \tag{2}$$

Lu et al. use a random identity TID_i to protect identity privacy [10]. But as the identity is a fixed value, a user could be tracked by an adversary. Schemes [29–32] use a similar method, but all these procedures are prone to suffer from tractability attack.

In scheme proposed by Wu et al., every time the gateway gives a new $\mathrm{PID}_{\mathrm{newMU}}$ for the user [4]. But in this case, there is a potential loss of synchronization problem: if the adversary blocks the $\mathrm{PID}_{\mathrm{newMU}}$ from being sent to the user, then the two parties may lose their synchronization. Das et al. protect the identity of the user by generating a new masked identity every time in a similar way, but this scheme suffers from loss of synchronization problem, too [33].

Jung et al. use the similar method with the scheme [13] of Farash et al. [6]. The key to encrypt the identity of a single user is the same for all the users. This scheme has the same problem that has been discussed. What a user sends to the gateway node is as follows: $\mathrm{DID}_i = h(\mathrm{ID}_i \parallel R_1)$, $k = h(\mathrm{DID}_i \parallel v^* \parallel T_1)$, $A_i = E_k(\mathrm{DID}_i \parallel R_1 \parallel T_1)$, so other users could learn DID_i by decrypting A_i with the same key v^*. Besides, this scheme has the same inner side attacker problem, a detailed analysis is shown in Section 7.4.

Rabin cryptosystem with quadratic residue problem is used to encrypt a message [11, 34]. Assume $n = pq$, where p and q are two large primes. If $y = x^2 \mod n$ has a solution, that is, there exists a square root for y, then y is called a quadratic residue mod$n$. The set of all quadratic residue numbers in $[1, n-1]$ is denoted by QR_n. The quadratic residue problem states that, for $y \in \mathrm{QR}_n$, it is hard to find x without the knowledge of p and q due to the difficulty of factoring n [35]; this is a kind of public-key encryption method.

Chatterjee and Das provide a similar methodology of protecting the identity of the user. They use the ECC based public key methods [15]. Besides, they try to combine the authentication scheme with an attributed based access control scheme. He et al. use a similar method, while they use exponentiation operations instead [36].

We summarize some of them in Table 1. From the table, it can be inferred that privacy is a problem that has not drawn enough attention from the researchers. In some schemes,

Table 1: Comparison of protection of privacy.

Schemes	Sensor anonymity	User anonymity	Shared key privacy
Choi et al. [1]	×	×	√
Shi and Gong [2]	×	×	√
Chang and Le [3, Scheme 1]	×	×	×
Chang and Le [3, Scheme 2]	×	×	√
Wu et al. [4]	√	×	√
Das et al. [5]	√	×	√
Jung et al. [6]	√	×	×
Fan et al. [7]	×	×	×
Amin and Biswas [8]	×	×	×
Nam et al. [9]	×	×	√
Lu et al. [10]	√	√	×
Zhao et al. [11]	√	×	×
Hou et al. [12]	×	×	×
Farash et al. [13]	×	×	×
Turkanović et al. [14]	×	×	×
PriAuth	√	√	√

all the users share the same key to encrypt their identities, this means the encrypted identity could be decrypted by a malicious or curious user using the same key [5, 6, 10, 13]. Some of the schemes fail to enable the anonymity of the user or sensor, such as [37–39]. We adopt the ECC based method to enable the anonymity, which is similar to [15–21] because "ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security" [40]. The gateway has a public key that is known by every user; all the identities are encrypted by an XOR method with a new key which is generated from gateway's public key before the identities are sent to the gateway. Thus, only the gateway could learn the identities.

As for the shared key between user and sensor, in some schemes, the gateway knows the shared key in schemes [6–8, 11–14], while, in some others, the gateway does not know the key, they use Diffie–Hellman (DH) anonymous key agreement protocol to build the shared key [1, 2, 4, 5, 9, 30]. As we have discussed, the gateway is not allowed to know the shared key in order to prevent a curious gateway from eavesdropping the sensor data.

3. Preliminary

Elliptic Curve Cryptography (ECC) is a public-key cryptography approach based on the algebraic structure of elliptic curves over finite fields. For current cryptographic purposes, an elliptic curve is a plane curve over a finite field (rather than the real numbers) which consists of the points satisfying the following:

$$y^2 = x^3 + ax + b. \tag{3}$$

In order to use ECC, all parties must agree on all the domain parameters of the elliptic curve $\{p, a, b, G, n, h\}$:

$F(p)$: the finite field over p, where p is a prime and represents the size of the finite field

(a, b): the parameters of elliptic curves $y^2 = x^3 + ax + b$ over $F(p)$

$G(x_p, y_p)$: generator point, but $G \neq 0$

n: the order of the base point G

h: cofactor, an integer, $h = F(p)/n$

Elliptic Curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol that allows two parties; each has an elliptic curve based public, private key pair, to establish a shared secret over an insecure channel. Suppose Alice wants to establish a shared key with Bob, but the channel available for them is not safe. Initially, the domain parameters (p, a, b, G, n, h) must be agreed upon. Also, each party must have a key pair suitable for elliptic curve cryptography, consisting of a private key d (a randomly selected integer in the interval $[1, n-1]$) and a public key Q (where $Q = dG$, that is, the result of adding G together d times).

Alice's private key and public key are (d_A, Q_A); Bob's key pair is (d_B, Q_B). Alice computes $d_A Q_B$ while Bob computes $d_B Q_A$. So the shared key between them is $d_A Q_B = d_B Q_A$, because

$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A. \tag{4}$$

4. Privacy Enhanced Scheme: PriAuth

The structure model of our scheme is depicted in Figure 1. A gateway is introduced to help user and sensor authenticate each other. We suppose this gateway is trustworthy.

4.1. Symbols Used in the PriAuth. Before the scheme begins, GWN (gateway node) generates the parameters for ECC encryption (p, a, b, G, n, h). After that, GWN generates its public-key pair (d_g, Q_g); besides, GWN generates a secret key X_{GWN}. The symbols are summarized in Table 2.

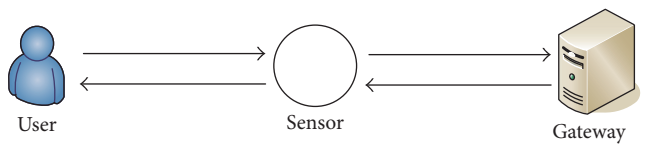

Figure 1: The structure of the model.

Table 2: Symbols used in the PriAuth.

Symbols	Meaning
GWN	Gateway node
U_i	The ith user
S_j	The jth sensor node
ID_i	The ith user's identity
SID_j	The jth sensor's identity
$\|$	String connector, connect two strings together
$\oplus$	XOR operation
X_{GWN}	GWN's secret value, master key
$X_{\mathrm{GWN\text{-}}S_j}$	Shared key between S_j and GWN
(d_g, Q_g)	The private key and public key of GWN
G	The generator of ECC
SK, SK′	Shared key between user U_i and S_j
T_1, T_2	Timestamp
h	Hash function

4.2. Registration Phase of the Sensor. The registration messages of the sensor in registration phase are sent via the public channel. Sensor S_j conducts the following steps for registration:

(1) It creates a random number r_j and gets the timestamp T_1.

(2) It covers its password with r_j, $MN_j = r_j \oplus X_{\mathrm{GWN\text{-}}S_j}$ and generates a hash value $MP_j = h(X_{\mathrm{GWN\text{-}}S_j} \| r_j \| \mathrm{SID}_j \| T_1)$.

(3) It sends $\{\mathrm{SID}_j, MP_j, MN_j, T_1\}$ to GWN via a public channel.

After GWN receives S_j's registration message $\{\mathrm{SID}_j, MP_j, MN_j, T_1\}$. GWN has to check the freshness of the message by T_1, if the message is not fresh, GWN abandons the message. Then GWN computes $r_j' = MN_j \oplus X_{\mathrm{GWN\text{-}}S_j}$. GWN checks if MP_j equals $h(X_{\mathrm{GWN\text{-}}S_j} \| r_j' \| \mathrm{SID}_j \| T_1)$. If they are not equal, GWN abandons the message. GWN continues the sensor registration phase in the following steps. The registration phase is described in Table 3.

(1) GWN computes $x_j = h(\mathrm{SID}_j \| X_{\mathrm{GWN}})$, $e_j = x_j \oplus h(\mathrm{SID}_j \| X_{\mathrm{GWN\text{-}}S_j})$.

(2) GWN gets the timestamp T_2 and gets the hash value $f_j = h(x_j \| X_{\mathrm{GWN\text{-}}S_j} \| T_2)$.

(3) GWN sends $\{e_j, f_j, T_2, p, a, b, G, n, h, Q_g\}$ to sensor S_j.

After receiving the message, S_j first checks the freshness of T_2, then computes $x_j = e_j \oplus h(\mathrm{SID}_j \| X_{\mathrm{GWN\text{-}}S_j})$, and checks if $f_j = h(x_j \| X_{\mathrm{GWN\text{-}}S_j} \| T_2)$; if they are equal, S_j stores $\{x_j, p, a, b, G, n, h, Q_g\}$ in its memory.

4.3. Registration Phase of the User. User U_i chooses a random number r_i and computes $MP_i = h(r_i \| \mathrm{ID}_i \| \mathrm{PW}_i)$. U_i then sends $\{\mathrm{ID}_i, MP_i\}$ to GWN via a secure channel.

After receiving the user registration message $\{\mathrm{ID}_i, MP_i\}$, GWN computes $d_i = h(\mathrm{ID}_i \| X_{\mathrm{GWN}})$, $f_i = d_i \oplus MP_i$. Finally, GWN sends $\{f_i, p, a, b, G, n, h, Q_g\}$ to U_i.

After receiving $\{f_i, p, a, b, G, n, h, Q_g\}$, U_i inserts the previously selected random nonce r_i into it, now what in the smart card is $\{MP_i, f_i, r_i, p, a, b, G, n, h, Q_g\}$. The registration phase is described in Table 4.

4.4. Login and Authentication Phase. If user U_i wants to access a sensor's data, U_i has to login first. This login process is completed by the smart card SC. A user inserts his smart card SC into a card reader and inputs his identity ID_i' and password PW_i'. SC computes a temporary version $MP_i' = h(r_i \| \mathrm{ID}_i' \| \mathrm{PW}_i')$ using the inserted PW_i', ID_i' and the stored value r_i. Then SC compares MP_i' with MP_i in the smart card. If they are equal, SC acknowledges the legitimacy of U_i.

After user U_i passes through the verification, then SC prepares for the authentication process. SC computes $d_i = f_i \oplus MP_i'$ using MP_i' in login phase. SC chooses a random number $k_1 \in [1, n-1]$ and gets the timestamp T_1. SC then computes the following data:

$$A = k_1 \cdot G$$
$$K_{ug} = h(T_1 \| k_1 \cdot Q_g)$$
$$M_1 = (\mathrm{ID}_i, \mathrm{SID}_j) \oplus K_{ug}$$
$$M_2 = h(A \| M_1 \| d_i \| T_1)$$

Then SC sends Message 1 $= \{A, M_1, M_2, T_1\}$ to sensor S_j via a public channel.

After receiving $\{A, M_1, M_2, T_1\}$ from U_i, sensor S_j first checks the freshness of T_1 and S_j abandons the message if T_1 is not fresh and otherwise goes to the next step. S_j chooses a random number $k_2 \in [1, n-1]$ and gets the timestamp T_2. S_j then computes the following data:

$$B = k_2 \cdot G$$
$$M_3 = h(B \| M_2 \| x_j \| T_2)$$

S_j sends Message 2 $= \{A, M_1, M_2, T_1, B, M_3, T_2\}$ to GWN via a public channel.

After receiving the message $\{A, M_1, M_2, T_1, B, M_3, T_2\}$, GWN first checks the freshness of T_1 and T_2, if T_1 or T_2 is not fresh, GWN abandons the message; otherwise GWN completes the following steps:

(1) GWN computes $K_{ug}' = h(T_1 \| d_g \cdot A)$.

(2) GWN gets ID_i' and SID_j' by $(\mathrm{ID}_i', \mathrm{SID}_j') = M_1 \oplus K_{ug}'$.

(3) GWN computes d_i' by $d_i' = h(\mathrm{ID}_i' \| X_{\mathrm{GWN}})$.

(4) GWN computes x_j' by $x_j' = h(\mathrm{SID}_j' \| X_{\mathrm{GWN}})$.

Table 3: Registration phase of the sensor.

Sensor	Gateway
$\mathrm{SID}_j, X_{\mathrm{GWN\text{-}}S_j}$	*master key* X_{GWN} *for each sensor stores* $\mathrm{SID}_j, X_{\mathrm{GWN\text{-}}S_j}$
random number r_j	
gets timestamp T_1	
$MN_j = r_j \oplus X_{\mathrm{GWN\text{-}}S_j}$	
$MP_j = h(X_{\mathrm{GWN\text{-}}S_j} \parallel r_j \parallel \mathrm{SID}_j \parallel T_1)$	
$\xrightarrow{\{\mathrm{SID}_j, MP_j, MN_j, T_1\}}$	
	checks if T_1 *is fresh*
	$r_j' = MN_j \oplus X_{\mathrm{GWN\text{-}}S_j}$
	$MP_j =? h\left(X_{\mathrm{GWN\text{-}}S_j} \parallel r_j \parallel \mathrm{SID}_j \parallel T_1\right)$
	gets timestamp T_2
	$x_j = h\left(\mathrm{SID}_j \parallel X_{\mathrm{GWN}}\right)$
	$e_j = x_j \oplus h\left(\mathrm{SID}_j \parallel X_{\mathrm{GWN\text{-}}S_j}\right)$
	$f_j = h\left(x_j \parallel X_{\mathrm{GWN\text{-}}S_j} \parallel T_2\right)$
checks if: T_2 *is fresh*	$\xleftarrow{\{e_j, f_j, T_2, p, a, b, G, n, h, Q_g\}}$
$x_j = e_j \oplus h(\mathrm{SID}_j \parallel X_{\mathrm{GWN\text{-}}S_j})$	
$f_j =? h(x_j \parallel X_{\mathrm{GWN\text{-}}S_j} \parallel T_2)$	
stores $\{x_j, p, a, b, G, n, h, Q_g\}$	

Table 4: Registration phase of the user.

User	Gateway
$\mathrm{ID}_i, \mathrm{PW}_i$	*master key* X_{GWN}
random number r_i	
$MP_i = h(r_i \parallel \mathrm{ID}_i \parallel \mathrm{PW}_i)$	
$\xrightarrow{\{\mathrm{ID}_i, MP_i\}}$	
	$d_i = h(\mathrm{ID}_i \parallel X_{\mathrm{GWN}})$
	$f_i = d_i \oplus MP_i$
inserts into the smart card	$\xleftarrow{\{f_i, p, a, b, G, n, h, Q_g\}}$
$\{MP_i, f_i, r_i, p, a, b, G, n, h, Q_g\}$	

(5) GWN uses d_i', A, M_1 and T_1 to check if $M_2 = h(A \parallel M_1 \parallel d_i' \parallel T_1)$. If they are equal, the procedure goes to next step; otherwise it terminates here.

(6) GWN uses x_j', B, M_2 and T_2 to check if $M_3 = h(B \parallel M_2 \parallel x_j' \parallel T_2)$. If they are equal, the procedure goes to next step; otherwise it terminates here.

(7) GWN calculates the following messages:

$$M_4 = h(A \parallel x_j \parallel M_3 \parallel B \parallel T_2)$$

$$M_5 = h(B \parallel d_i \parallel M_2 \parallel A \parallel T_1)$$

(8) GWN sends Message 3 = $\{M_4, M_5\}$ to sensor S_j.

After receiving the message $\{M_4, M_5\}$, sensor S_j does the following calculations:

(1) S_j uses A getting from user to checks if $M_4 = h(A \parallel x_j \parallel M_3 \parallel B \parallel T_2)$. If they are equal, the procedure goes to next step; otherwise it terminates here.

(2) S_j calculates the shared key SK between U_i and S_j: $\mathrm{SK} = h(k_2 \cdot A) = h(k_1 \cdot k_2 \cdot G)$.

(3) S_j sends Message 4 = $\{B, M_5\}$ to user U_i

After U_i receives the message $\{B, M_5\}$, U_i goes to the following steps. The whole process is in Table 5.

(1) U_i uses B getting from S_j to check if $M_5 = h(B \parallel d_i \parallel M_2 \parallel A \parallel T_1)$; if they are equal, the procedure goes to next step; otherwise it terminates here.

(2) U_i calculates the shared key SK' between U_i and S_j: $\mathrm{SK}' = h(k_1 \cdot B) = h(k_1 \cdot k_2 \cdot G)$.

4.5. Password Change Phase. If a user wants to change his password, he has to be authenticated by the smart card first. We state the password change process in Table 6, which is a summary of the steps:

(1) A user U_i inserts his smart card SC into a card reader and inputs their identity and password: $\mathrm{ID}_i, \mathrm{PW}_i$.

(2) SC computes $h(r_i \parallel \mathrm{ID}_i \parallel \mathrm{PW}_i)$ using password ID_i, PW_i, and the stored r_i.

(3) SC compares $h(r_i \parallel \mathrm{ID}_i \parallel \mathrm{PW}_i)$ with the stored version of MP_i in the smart card; if they are equal, SC acknowledges the legitimacy of user U_i.

Table 5: Login and authentication phase.

User	Sensor	Gateway
$\mathrm{ID}_i, \mathrm{PW}_i, d_i$	SID_j, x_j	d_g, Q_g
User: inserts SC *into terminal*		
User: input ID_i' *and* PW_i'		
SC: $MP_i' = h(r_i \parallel \mathrm{ID}_i' \parallel \mathrm{PW}_i')$		
SC: $d_i = f_i \oplus MP_i'$		
SC: *random* k_1, $A = k_1 \cdot G$		
SC: *gets timestamp* T_1		
SC: $K_{ug} = h(T_1 \parallel k_1 \cdot Q_g)$		
SC: $M_1 = (\mathrm{ID}_i, \mathrm{SID}_j) \oplus K_{ug}$		
SC: $M_2 = h(A \parallel M_1 \parallel d_i \parallel T_1)$		
$\xrightarrow{\{A, M_1, M_2, T_1\}}$		
	checks the freshness of T_1	
	random k_2, $B = k_2 \cdot G$	
	gets timestamp T_2	
	$M_3 = h(B \parallel M_2 \parallel x_j \parallel T_2)$	
	$\xrightarrow{\{A, M_1, M_2, T_1, B, M_3, T_2\}}$	
		checks the freshness of T_1, T_2
		$K_{ug}' = h(T_1 \parallel d_g \cdot A)$
		$(\mathrm{ID}_i', \mathrm{SID}_j') = M_1 \oplus K_{ug}'$
		$d_i' = h(\mathrm{ID}_i' \parallel X_{\mathrm{GWN}})$
		$x_j' = h(\mathrm{SID}_j' \parallel X_{\mathrm{GWN}})$
		checks if: $M_2 = h(A \parallel M_1 \parallel d_i' \parallel T_1)$
		checks if: $M_3 = h(B \parallel M_2 \parallel x_j' \parallel T_2)$
		$M_4 = h(A \parallel x_j \parallel M_3 \parallel B \parallel T_2)$
		$M_5 = h(B \parallel d_i \parallel M_2 \parallel A \parallel T_1)$
	checks if: $M_4 = h(A \parallel x_j \parallel M_3 \parallel B \parallel T_2)$	$\xleftarrow{\{M_4, M_5\}}$
	$\mathrm{SK} = h(k_2 \cdot A) = h(k_1 \cdot k_2 \cdot G)$	
Checks if: $M_5 = h(B \parallel d_i \parallel M_2 \parallel A \parallel T_1)$	$\xleftarrow{\{B, M_5\}}$	
$\mathrm{SK}' = h(k_1 \cdot B) = h(k_1 \cdot k_2 \cdot G)$		

Table 6: Password change phase of the user.

User
User: inserts SC *into terminal*
User: inserts ID_i *and* PW_i
SC: *check if* $MP_i =? h(r_i \parallel \mathrm{ID}_i \parallel \mathrm{PW}_i)$
SC: $d_i = f_i \oplus MP_i$
User: inputs a new password PW_i'
SC: $MP_i' = h(r_i \parallel \mathrm{ID}_i \parallel \mathrm{PW}_i')$
SC: $f_i' = d_i \oplus MP_i'$
SC: *changes* f_i *with* f_i'

(4) SC computes $d_i = f_i \oplus MP_i$ using the stored values f_i and the user password MP_i.

(5) User U_i inputs the new password PW_i'.

(6) SC uses this new PW_i' to update the stored version of f_i with $f_i' = d_i \oplus MP_i'$.

5. Security Analysis Using BAN Logic

5.1. Some Basic Knowledge of BAN Logic. A security analysis of PriAuth using Burrows-Abadi-Needham logic (BAN logic) [41] is conducted in this part. With the help of BAN logic, we can determine whether the exchanged information is trustworthy and secure against eavesdropping. First, some symbols and primary postulates used in BAN logic are described in Tables 7 and 8.

Table 7: Symbols of BAN logic.

Symbol	Meaning
$P \mid\equiv X$	P believes X
$P \triangleleft X$	P sees/receives X
$P \mid\sim X$	P once said X (or P sent X)
$P \mid\Rightarrow X$	P controls X
$\#(X)$	X is fresh
$P \overset{k}{\leftrightarrow} Q$	P and Q communicate using shared key K
$\overset{k}{\rightarrow} Q$	K is the public key of Q
$\{X\}_k$	Message X is encrypted by K
$\{X\}_{k^{-1}}$	Message X is encrypted by private key K

5.2. The Premise and Proof Goals of PriAuth. U_i, S_j, and GWN are used as the user, sensor, and the gateway. Suppose GWN is trustworthy, if GWN believes that U_i has said message X and GWN believes that X is fresh, GWN would send X to S_j. If S_j believes X is fresh and S_j believes GWN once said X, then S_j believes U_i said X. This could be translated into BAN logic

Table 8: Some primary BAN logic postulates.

Rule	BAN Logic form
$\triangleleft$ rule	$\frac{P \mid\equiv \xrightarrow{k} P, P \triangleleft \{X\}_k}{P \triangleleft X}, \frac{P \mid\equiv P \xleftrightarrow{k} Q, P \triangleleft \{X\}_k}{P \triangleleft X}, \frac{P \mid\equiv \xrightarrow{k} Q, P \triangleleft \{X\}_{k^{-1}}}{P \triangleleft X}$
$\mid\sim$ introduction rule	$\frac{P \mid\equiv \xrightarrow{k} Q,\ P \triangleleft \{X\}_{k^{-1}}}{P \mid\equiv Q \mid\sim X}, \frac{P \mid\equiv P \xleftrightarrow{k} Q, P \triangleleft \{X\}_k}{P \mid\equiv Q \mid\sim X}$
$\mid\sim$ elimination rule	$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$
#()-introduction	$\frac{P \text{ creates } X}{P \mid\equiv \# X}$
Jurisdiction or control rule	$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$
$\xleftrightarrow{k}$ introduction rule	$\frac{P \mid\equiv \#(k), P \mid\equiv Q \mid\equiv X}{P \mid\equiv P \xleftrightarrow{k} Q}$
Freshness rule	$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$
Elimination of multipart messages rule	$\frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X}, \frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X}, \frac{P \mid\equiv (X, Y)}{P \mid\equiv X}, \frac{P \triangleleft (X, Y)}{P \triangleleft X}, \frac{P \mid\equiv \#(X, Y)}{P \mid\equiv \#(X)}$

like (postulate A). According to the "$\mid\sim$ elimination rule," (postulate A) could be simplified as (postulate B). It is the same as the message that sensor S_j sends to GWN. If GWN believes S_j once said another message X (the same notion is used for simplification), and GWN believes X is fresh, GWN would send X to U_i. If U_i believes X is fresh and U_i believes GWN once said X, then U_i believes S_j said X. In the same way, we can get (postulate C).

$$\frac{\mathrm{GWN} \mid\equiv \#(X), \mathrm{GWN} \mid\equiv U_i \mid\sim X, S_j \mid\equiv \#(X), S_j \mid\equiv \mathrm{GWN} \mid\sim X}{S_j \mid\equiv U_i \mid\sim X} \quad \text{(postulate A)}$$

$$\frac{\mathrm{GWN} \mid\equiv U_i \mid\equiv X, S_j \mid\equiv \mathrm{GWN} \mid\equiv X}{S_j \mid\equiv U_i \mid\sim X} \quad \text{(postulate B)}$$

$$\frac{\mathrm{GWN} \mid\equiv S_j \mid\equiv X, U_i \mid\equiv \mathrm{GWN} \mid\equiv X}{U_i \mid\equiv S_j \mid\sim X} \quad \text{(postulate C)}$$

The proof goals of PriAuth in BAN logic form are in the way described below. These goals could ensure U_i and S_j to agree on a shared key SK.

$$\begin{aligned} &(1)\ U_i \mid\equiv U_i \xleftrightarrow{\mathrm{SK}} S_j \\ &(2)\ S_j \mid\equiv U_i \xleftrightarrow{\mathrm{SK}} S_j. \end{aligned} \tag{5}$$

5.3. Preparation for Proof. Before the proof begins, messages have to be transformed into an idealized form, the messages of PriAuth in idealized form in BAN logic are given in Table 9 ($K_{ug} = h(T_1 \parallel k_1 \cdot Q_g)$). At the same time, some assumptions have to be made, so (postulate B) and (postulate C) are included as assumptions A11 and A12. The assumptions are listed in Table 10.

5.4. The Proof of PriAuth. The whole proof of the proposal is in Appendix A. It has been divided into 3 parts related to Message 2, Message 3, and Message 4 separately. The two goals of the scheme are proved at the Message 3 and Message 4. The proof results show that PriAuth is secured under BAN logic.

6. AVISPA Verification

AVISPA (Automated Validation of Internet Security Protocols and Applications) is "*a push-button tool for the automated validation of Internet security-sensitive protocols and applications*" [42]. Recently, many papers have used this method as a way to authenticate their protocols, like [24–26]. HLPSL (High Level Protocols Specification Language) is a role-based language that is used to describe security protocols and specifying their intended security properties, as well as a set of tools to formally validate them. We write the protocol in HLPSL and test the protocol. The code is in Appendix B. The goal of PriAuth is to create a key that is shared by a user and a sensor. The validation result of the protocol is in Table 11. Considering all these testing activities, it could be concluded that our protocol is safe. PriAuth can protect the privacy of the user identity, sensor identity, and the key between the user and sensor.

Table 9: The idealization form of the message.

Message	Flow	Idealized form
1	$U_i \longrightarrow S_j$	$\left\{A, \left\{\mathrm{ID}_i, \mathrm{SID}_j\right\}_{K_{ug}}, \left\{A, \left\{\mathrm{ID}_i, \mathrm{SID}_j\right\}_{K_{ug}}, T_1\right\}_{d_i}, T_1\right\}$
2	$S_j \longrightarrow \mathrm{GWN}$	$\left\{A, \left\{\mathrm{ID}_i, \mathrm{SID}_j\right\}_{K_{ug}}, \left\{A, \left\{\mathrm{ID}_i, \mathrm{SID}_j\right\}_{K_{ug}}, T_1\right\}_{d_i}, T_1, B, \left\{B, M_2, T_2\right\}_{x_j}, T_2\right\}$
3	$\mathrm{GWN} \longrightarrow S_j$	$\left\{\left\{A, M_3, B, T_2\right\}_{x_j}, \left\{B, M_2, A, T_1\right\}_{d_i}\right\}$
4	$S_j \longrightarrow U_i$	$\left\{B, \left\{B, M_2, A, T_1\right\}_{d_i}\right\}$

Table 10: Some assumptions.

Number	Assumptions
A1	$\mathrm{GWN} \mid\equiv \#(A)$
A2	$\mathrm{GWN} \mid\equiv \#(B)$
A3	$S_j \mid\equiv \#(A)$
A4	$U_i \mid\equiv \#(B)$
A5	$U_i \mid\equiv \mathrm{GWN} \overset{d_i}{\longleftrightarrow} U_i$
A6	$\mathrm{GWN} \mid\equiv \mathrm{GWN} \overset{d_i}{\longleftrightarrow} U_i$
A7	$U_i \mid\equiv \mathrm{GWN} \overset{K_{ug}}{\longleftrightarrow} U_i$
A8	$\mathrm{GWN} \mid\equiv \mathrm{GWN} \overset{K_{ug}}{\longleftrightarrow} U_i$
A9	$S_j \mid\equiv \mathrm{GWN} \overset{x_j}{\longleftrightarrow} S_j$
A10	$\mathrm{GWN} \mid\equiv \mathrm{GWN} \overset{x_j}{\longleftrightarrow} S_j$
A11	$\dfrac{\mathrm{GWN} \mid\equiv U_i \mid\equiv X, S_j \mid\equiv \mathrm{GWN} \mid\equiv X}{S_j \mid\equiv U_i \mid\sim X}$
A12	$\dfrac{\mathrm{GWN} \mid\equiv S_j \mid\equiv X, U_i \mid\equiv \mathrm{GWN} \mid\equiv X}{U_i \mid\equiv S_j \mid\sim X}$
A13	$S_j \mid\equiv U_i \mapsto A$
A14	$U_i \mid\equiv S_j \mapsto B$

7. Security and Privacy Analysis

In this section, we conduct a security comparison of the schemes that has been depicted as Table 12. For the scheme in [3], we only consider the second situation.

7.1. Traceability Protection. Traceability means the adversary can track a user or a sensor according to their identities or masked identities like in the scheme [5, 10, 29–32]. Once some fixed information about the identities is used in a scheme, then this scheme could probably be tracked by an adversary. One possible solution is to update their masked identity every time like in the schemes shown in [4, 7]. But these kinds of solutions are vulnerable to loss of synchronization attack.

7.2. Synchronization Loss Attack. In order to protect the identity of the user, the gateway will generate a new identity for them when it is requested [4]. But if an adversary prevents this new identity from being received by the user, the user could not update his old identity while the gateway has updated its stored version of the user's identity. When the user logs in for the next time, this legitimate user will not be treated as a legal one anymore. A similar problem exists in the scheme [7].

7.3. Malicious Sensor Attack. Like in scheme [13], the gateway only checks the legitimacy of a sensor. If the sensor is a legitimate one, the gateway will reply some key information to the sensor, but the gateway does not check if the sensor is the one that the user wants to talk to. So a legitimate but malicious sensor could launch an attack.

When a user sends a request message $\{M_1, M_2, M_3, T_1\}$ to a sensor, an inner side legitimate sensor can intercept this message to generate its own $\{M_4', M_5', \mathrm{ESID}_j', T_2'\}$ and send this message to the gateway, as the gateway only checks the legitimacy of the sensor. Therefore, this inner side sensor will definitely be treated as a legal sensor. The gateway will send $\{M_6', M_7', M_8', M_9', T_3'\}$ to the sensor. Afterwards, the sensor will be able to send $\{M_6', M_8', M_{10}', T_3', T_4'\}$ to the user, and it will be treated as a legal sensor by the user, but the user will not check if this is the sensor he wants to talk to. In this way, the sensor could send false data to the user.

7.4. Inside User Attack. In scheme [6], all the users share a key v^*, so there is a potential risk. The message a gateway sends to the user is $D_i = E_k(\mathrm{DID}_i \parallel \mathrm{SID}_n \parallel \mathrm{SK} \parallel R_1 \parallel T_4)$, where $k = h(\mathrm{DID}_i \parallel v^* \parallel T_4)$, in which DID_i and T_4 are public message, and v^* is shared by all the legitimate users. This means any legitimate user could decrypt D_i to get the shared key SK.

7.5. User Impersonation Attack. In scheme [1], when a user asks to access a sensor's data, he could send his request $M_1 = \{\mathrm{ID}_u, \mathrm{ID}_{S_n}, X, T_u, \alpha, \omega\}$ to the sensor.

$$\begin{aligned} X' &= r_u \times P, \\ X &= r_u \times K_u, \\ \omega &= h\left(\mathrm{ID}_u \parallel h\left(\mathrm{ID}_{S_n} \parallel h\left(X \oplus Y\right)\right) \parallel T_u\right), \\ \alpha &= h\left(\mathrm{ID}_u \parallel \mathrm{ID}_{S_n} \parallel X \parallel X' \parallel T_u \parallel \omega\right). \end{aligned} \tag{6}$$

ID_u, K_u, P, and ID_{S_n} are sent publicly; r_u is a random number generated by the user, whereas T_u is a timestamp. Only $h(X \oplus Y)$ is regarded as secret information between the user and the gateway. $h(X \oplus Y)$ is shared by all the users; other legitimate users, say a legitimate user with ID_u', could easily generate a request the same as M_1, and then ID_u' will be treated as ID_u by the gateway.

8. Comparison

8.1. Computational Performance. The normal way to compute the execution time of the protocol is to calculate protocol's

Table 11: Simulation results.

CL-AtSe back-end	OFMC
SUMMARY	% OFMC
SAFE	% Version of 2006/02/13
	SUMMARY
DETAILS	SAFE
BOUNDED_NUMBER_OF_SESSIONS	DETAILS
TYPED_MODEL	BOUNDED_NUMBER_OF_SESSIONS
	PROTOCOL
PROTOCOL	
	/home/iotdev/avispa/avispa-1.1/testsuite/results/usg.if
/home/iotdev/avispa/avispa-1.1/testsuite/results/usg.if	GOAL
GOAL	as_specified
As Specified	BACKEND
	OFMC
BACKEND	COMMENTS
CL-AtSe	STATISTICS
	parseTime: 0.00 s
STATISTICS	searchTime: 0.05 s
Analysed: 14 states	visitedNodes: 24 nodes
Reachable: 4 states	depth: 4 plies
Translation: 0.00 seconds	
Computation: 0.00 seconds	

Table 12: Security feature comparison.

Security feature	[1]	[3, Scheme 2]	[7]	[9]	PriAuth
User anonymity	×	×	√	√	√
Sensor anonymity	×	×	×	×	√
Shared key privacy	√	√	√	√	√
Traceability of user	×	×	√	√	√
Traceability of sensor	×	×	×	×	√
Loss of synchronization	√	√	×	√	√
Malicious sensor attack	√	√	√	√	√
User impersonation attack	×	√	√	√	√
Sensor impersonation attack	√	√	√	√	√
Replay attack	√	√	×	√	√
Inside user attack	√	√	√	√	√

computational costs of different operations, and the operations' execution time is measured by simulation [3–14]. The execution time of XOR operation is very small compared to an elliptic curve point multiplication or hash operation; we neglect it when computing the time approximately [3]. We use the famous MIRACL++ Library [43] (example code can be found at [44]). The experiment is conducted in Visual C++ 2017 on a 64-bit Windows 7 operating system, 3.5 GHz processor, 8 GB memory. The hash function is the SHA-1; the symmetric encryption/decryption function is AES with a 128-bit long key of the MR_PCFB1 form (using one string to encrypt another string, the same hash function is called to get the hashed form of the key string). The elliptic curve encryption scheme is ECC-160. The results are shown in Table 13. T_{mac} is the time for HMAC with SHA-1 operation, according to [9] $T_{\text{mac}} \approx T_H$. The final result is in Table 14.

8.2. Communication Performance. The sum of each variable length in bytes which a sensor node and a gateway node need while performing authentication process is calculated for comparison of the communication cost. The identity or password is 8-byte long [13]. The sizes of the general hash function's output and timestamp are 20 bytes and 4 bytes, respectively [45]. The random point of ECC-160 is 20 bytes. The result is shown in Table 15. The byte length of the AES encryption result is treated as byte length of the original data for approximation.

Table 13: Computation time of different operations.

Operations	Time	Experiment times
T_H: *one way hash function*	0.0394 ms	1000000
$T_{E/D}$: *symmetric encryption/decryption*	0.5728 ms	100000
T_{MUL}: *scalar multiplication in ECC-160*	3.66 ms	2733

9. Validation

LifeWear project intends to improve the quality of human life by using wearable equipment and applications for everyday use [46]. The main objective of LifeWear is the development of modern physiological monitoring to inspect human health parameters, like blood pressure, pulse, or the electrocardiogram of a patient in different environments. With real-time data of these health parameters, medical staffs can take actions instantly, which can greatly improve the quality of a treatment.

Since medical parameters are sent from patients to medical staffs, data security and patient's privacy are a must. In order to ensure the data confidentiality, all the data must be encrypted before they are sent. The proposed scheme helps the patients and medical staff building a shared key. This key will be used to encrypt the health parameters of the patient. In order to protect the privacy of the patient, all the identities are encrypted before they are sent as well. Since wearable sensors have only limited computability, we introduce a gateway to provide the patients and medical staff the shared key to be used in the system.

LifeWear project also makes use of a middleware solution able to hide heterogeneity and interoperability problem. This middleware is composed of four abstraction layers related to the functionalities covered in each of them, namely, hardware abstraction layer, low and high services, cross-layer services, and service composition platform.

The hardware abstraction layer includes the IoT hardware platform, the operating system, and the networking stack. It offers an easy way to port the solution to other hardware platforms. The low and high service layers define the software components needed to abstract the underlying network heterogeneity, thus providing an integrated, distributed environment to simplify programming tasks by means of a set of generic services, along with an access point to the management functions of the sensor network services. The upper layer is the service composition platform, designed to build applications using services offered by the lower layers. The cross-layer services are offered to both high and low level services in order to provide inner service composition. The proposal presented in this paper (PriAuth) has been deployed as a service inside this layer. The security service can be used by the upper layer (service composition) to compose newly secured services, based on the services presented in the lower layers.

The architecture has been deployed over a commercial IoT node solution called SunSPOT platform, manufactured by Oracle. Main characteristics of SunSPOT hardware platform are as follows:

(a) Processor: ARM 920T CPU (400 MHz, 32 bits)

(b) Memory: 1 Mb RAM, 8 Mb Flash memory

(c) Network: Chipcon 2420 radio with integrated antenna (IEEE 802.15.4 at 2.4 GHz)

(d) Data: USB interface, mini-USB connector

(e) Power supply: 3.6 V rechargeable 750 mAh Li-Ion battery

10. Conclusions

Privacy will be a big concern as more and more IoT equipment is applied into the medical scenarios. In this paper, we propose an authentication and key agreement scheme tailored for Wireless Sensor Networks. We focus on the privacy problems during the authentication process. Our scheme not only ensures the security of the data but also protects the identity privacy of the users and sensors. The shared key between the user and sensor is built by means of the Elliptic Curve Diffie–Hellman method, which could ensure forward privacy. The proposed scheme has been verified with BAN logic and AVISPA, which are the two most commonly used tools to validate the security of the communication scheme. Simulation results show that our scheme is feasible and secure. Furthermore, experiment results show that our scheme is comparable with the related works in terms of computation cost and more efficient in communication cost.

As part of our work in the LifeWear project, we focus on privacy problems during the authentication and key establishment processes. In future, we will pay more attention to authentication scheme without the help of the gateway.

Appendix

A. The Proof of PriAuth Using BAN Logic

The proof starts at Message 2. From Message 2 onwards, we can prove that GWN believes U_i once said A and GWN believes S_j once said B.

(1) According to Message 2, we get

$$\text{GWN} \lhd \left\{A, \left\{\text{ID}_i, \text{SID}_j\right\}_{K_{ug}}, \left\{A, \left\{\text{ID}_i, \text{SID}_j\right\}_{K_{ug}}, T_1\right\}_{d_i}, T_1, B, \left\{B, M_2, T_2\right\}_{x_j}, T_2\right\}. \tag{A.1}$$

Table 14: Computation cost of the login and authentication.

Schemes	User	Sensor	Gateway	Total	Total (ms)
Choi et al. [1]	$7T_H + 3T_{\text{MUL}}$	$4T_H + 2T_{\text{MUL}}$	$4T_H + 1T_{\text{MUL}}$	$15T_H + 6T_{\text{MUL}}$	22.551
Chang and Le [3, Scheme 2]	$7T_H + 2T_{\text{MUL}}$	$5T_H + 2T_{\text{MUL}}$	$9T_H$	$21T_H + 4T_{\text{MUL}}$	15.4674
Fan et al. [7]	$13T_H + 2T_{\text{MUL}}$	$4T_H + 2T_{\text{MUL}}$	$14T_H$	$31T_H + 4T_{\text{MUL}}$	15.8614
Nam et al. [9]	$3T_H + 1T_{E/D} + 1T_{\text{mac}} + 3T_{\text{MUL}}$	$1T_H + 2T_{\text{mac}} + 2T_{\text{MUL}}$	$2T_H + 2T_{E/D} + 3T_{\text{mac}} + 1T_{\text{MUL}}$	$6T_H + 3T_{E/D} + 6T_{\text{mac}} + 6T_{\text{MUL}}$	24.1512
PriAuth	$5T_H + 3T_{\text{MUL}}$	$3T_H + 2T_{\text{MUL}}$	$7T_H + 1T_{\text{MUL}}$	$15T_H + 6T_{\text{MUL}}$	22.551

Table 15: Communication comparison.

Schemes	M1	M2	M3	M4	Total bytes	Compared*
Choi et al. [1]	80	124	44	68	316	+64
Chang and Le [3, Scheme 2]	64	84	64	44	256	+4
Fan et al. [7]	128	68	60	100	356	+104
Nam et al. [9]	52	104	40	56	252	0
PriAuth	64	108	40	40	252	0

*Compared** means compared with our scheme; M1, M2, M3, and M4 mean Messages 1, 2, 3, and 4.

(2) According to (A.1) and "','-elimination rule"

$$\text{GWN} \lhd \left\{A, \left\{\text{ID}_i, \text{SID}_j\right\}_{K_{ug}}, \text{SID}_j, T_1\right\}_{d_i}, \tag{A.2}$$

$$\text{GWN} \lhd \{B, M_2, T_2\}_{x_j}. \tag{A.3}$$

(3) According to (A.2), A6, and "|~ introduction rule"

$$\text{GWN} \mid\equiv U_i \mid\sim \left\{A, \left\{\text{ID}_i, \text{SID}_j\right\}_{K_{ug}}, \text{SID}_j, T_1\right\}. \tag{A.4}$$

(4) According to (A.3), A10, and "|~ introduction rule"

$$\text{GWN} \mid\equiv S_j \mid\sim \{B, M_2, T_2\}. \tag{A.5}$$

(5) According to (A.4) and "','-elimination rule"

$$\text{GWN} \mid\equiv U_i \mid\sim A. \tag{A.6}$$

(6) According to (A.5) and "','-elimination rule"

$$\text{GWN} \mid\equiv S_j \mid\sim B. \tag{A.7}$$

(7) According to A1, (A.6), and "|~ elimination rule"

$$\text{GWN} \mid\equiv U_i \mid\equiv A. \tag{A.8}$$

(8) According to A2, (A.7), and "|~ elimination rule"

$$\text{GWN} \mid\equiv S_j \mid\equiv B. \tag{A.9}$$

The following content is the analysis of Message 3. From it, we can prove that S_j believes GWN believes A. Based on assumption A11, we can get that S_j believes U_i believes A; this process is shown at (A.10)~(A.17). Equations (A.18)~(A.20) prove the first goal of the scheme.

(9) Based on Message 3,

$$S_j \lhd \left\{\{A, M_3, B, T_2\}_{x_j}, \{B, M_2, A, T_1\}_{d_i}\right\}. \tag{A.10}$$

(10) According to (A.10) and "','-elimination rule"

$$S_j \lhd \left\{\{A, M_3, B, T_2\}_{x_j}\right\}. \tag{A.11}$$

(11) According to (A.11), A9, and "|~ introduction rule"

$$S_j \mid\equiv \text{GWN} \mid\sim \{A, M_3, B, T_2\}. \tag{A.12}$$

(12) According to (A.12) and "',"-elimination rule"

$$S_j \mid\equiv \text{GWN} \mid\sim A. \tag{A.13}$$

(13) According to A3, (A.13), and "|~ elimination rule"

$$S_j \mid\equiv \text{GWN} \mid\equiv A. \tag{A.14}$$

(14) According to A11, (A.8), (A.14), we get

$$S_j \mid\equiv U_i \mid\sim A. \tag{A.15}$$

(15) According to A3, (A.15), and "|~ elimination rule"

$$S_j \mid\equiv U_i \mid\equiv A. \tag{A.16}$$

(16) According to A13, (A.16), and "jurisdiction or control rule"

$$S_j \mid\equiv A. \tag{A.17}$$

(17) As k_2 is randomly created by S_j, according to "#()-introduction"

$$S_j \mid\equiv \#(k_2). \tag{A.18}$$

(18) According to (A.18), A3, A5, and "#()-promotion rule"

$$S_j \mid\equiv \#(\text{SK}) \quad \text{SK} = h(k_2 \cdot A). \tag{A.19}$$

(19) According to (A.19), (A.17), and "$\overset{k}{\longleftrightarrow}$ introduction rule"

$$S_j \mid\equiv S_j \overset{\text{SK}}{\longleftrightarrow} U_i. \tag{A.20}$$

The following is the analysis of Message 4, where it is proven that U_i believes GWN and believes B, based on assumption A12, so we can infer that U_i believes S_j believes B; this procedure is shown at (A.21)~(A.28). Equations (A.29)~(A.31) prove the first goal of the scheme. Until now, the two goals of the scheme have been proved at (A.20) and (A.31), so it can be claimed that this protocol is feasible and safe.

(20) Based on Message 4,

$$U_i \lhd \left\{B, \{B, M_2, A, T_1\}_{d_i}\right\}. \tag{A.21}$$

```
role user (Ui, Sj, GW : agent,
           Kdi: symmetric_key,
           Kug: symmetric_key,
           H: hash_func,
           P: text,
           SND_US,RCV_US: channel (dy))
played_by Ui
def=
  local State : nat,
  T1,K1,Na,Nb,SIDj,IDi,SK : text
  const user_sensor_sk,sc_user_id:protocol_id
  init  State := 0
  transition
  (1) State = 0  RCV_US(start)=|>
   State' := 2  /\ T1' := new()
                /\ K1' := new()
                /\ Na' := exp(P,K1')
                /\ SND_US(Na'
                          .xor((IDi.SIDj),Kug)
                          .H(Na'.xor((IDi.SIDj),Kug).Kdi.T1')
                          .T1')
                /\ secret(IDi,sc_user_id,{Ui, GW})
                /\ secret(IDi,sc_sensor_id,{Ui, GW})
  (2) State = 2  /\ RCV_US(Nb'
                           .H(Nb'.Kdi.H(Na.xor((IDi.SIDj),Kug).Kdi.T1).Na.T1))=|>
   State' := 4  /\ SK' := H(exp(Nb',K1))
                /\ witness(Ui,Sj,user_sensor_sk,SK')
                /\ request(Ui,Sj,user_sensor_sk,SK')
end role
```

Box 1

(21) According to (A.21) and "','-elimination rule"

$$U_i \triangleleft \left\{\left\{B, M_2, A, T_1\right\}_{d_i}\right\}. \tag{A.22}$$

(22) According to (A.22), A7, and "|~ introduction rule"

$$U_i \mid\equiv \mathrm{GWN} \mid\sim \left\{B, M_2, A, T_1\right\}. \tag{A.23}$$

(23) According to (A.23) and "','-elimination rule"

$$U_i \mid\equiv S_j \mid\sim B. \tag{A.24}$$

(24) According to A4, (A.23), and "|~ elimination rule"

$$U_i \mid\equiv \mathrm{GWN} \mid\equiv B. \tag{A.25}$$

(25) According to A12, (A.9), and (A.25), we get

$$U_i \mid\equiv S_j \mid\sim B. \tag{A.26}$$

(26) According to A4, (A.26), and "|~ elimination rule"

$$U_i \mid\equiv S_j \mid\equiv B. \tag{A.27}$$

(27) According to A14, (A.27), and "jurisdiction or control rule"

$$U_i \mid\equiv B. \tag{A.28}$$

(28) As k_2 is randomly created by U_i, according to "#()-introduction"

$$U_i \mid\equiv \#\left(k_1\right). \tag{A.29}$$

(29) According to (A.29), A4, A6, and "#()-promotion rule"

$$U_i \mid\equiv \#\left(\mathrm{SK}\right) \quad \mathrm{SK} = h\left(k_1 \cdot B\right). \tag{A.30}$$

(30) According to (A.30), (A.27), and "$\overset{k}{\longleftrightarrow}$ introduction rule"

$$U_i \mid\equiv S_j \overset{\mathrm{SK}}{\longleftrightarrow} U_i. \tag{A.31}$$

B. The HLPSL Code for PriAuth

The ECC public-key pair of the gateway is (d_g, Q_g). At the beginning of this protocol usage, every user generates a random number $k_1 \in [1, n-1]$ and calculates $A = k_1 \cdot G$, so we could treat (k_1, A), as the ECC key pair of this user, and we send A to the gateway. Now the two parties could calculate a shared key $k_1 \cdot Q_g = d_g \cdot A$. Thus, at the beginning of the scheme, we declare $K_{ug} = h(T_1 \parallel k_1 \cdot Q_g)$ to be a symmetric key between the two.

For the role of the user, see Box 1. For the role of the sensor, see Box 2. For the role of the gateway, see Box 3.

```
role sensor (Ui, Sj, GW : agent,
             Kxj: symmetric_key,
             H: hash_func,
             P: text,
             SND_US,RCV_US,SND_SG,RCV_SG: channel(dy))
played_by Sj
def=
  local State : nat,
  T1,T2,K2, Na,Nb,SK : text,
  Y,X,Z : message
  const user_sensor_sk:protocol_id
  init  State := 1
  transition
    (1) State  = 1  /\ RCV_US(Na'.Y'.Z'.T1') =|>
        State' := 3  /\ T2' := new()
                     /\ K2' := new()
                     /\ Nb' := exp(P,K2')
                     /\ SND_SG( Na'
                                .Y'
                                .Z'
                                .T1'
                                .Nb'
                                .H(Nb'.Z'.Kxj.T2')
                                .T2' )
    (2) State = 2  /\ RCV_SG( H(Na.Kxj.H(Nb.Z.Kxj.T2).T2)
                              .X' ) =|>
        State' := 4  /\ SK' := H(exp(Na,K2))
                     /\ witness(Sj,Ui,user_sensor_sk,SK')
                     /\ request(Sj,Ui,user_sensor_sk,SK')
                     /\ SND_US(Nb
                               .X')
end role
```

Box 2

```
role gateway (Ui, Sj, GW : agent,
              Kdi, Kxj: symmetric_key,
              Kug : symmetric_key,
              H: hash_func,
              SND_SG, RCV_SG: channel(dy))
played_by GW
def=
  local State : nat,
  T1,T2,Na,Nb,IDi,SIDj : text
  const sk_User_gwn,sk_sensor_gwn,sc_sensor_id,sc_user_id:protocol_id
  init  State := 5
  transition
   (1) State  = 5  /\ RCV_SG( Na'
                              .xor((IDi'.SIDj'),Kug)
                              .H(Na'.xor((IDi'.SIDj'),Kug).Kdi.T1')
                              .T1'
                              .Nb'
                              .H(Nb'.H(Na'.xor((IDi'.SIDj'),Kug).Kdi.T1').Kxj.T2')
                              .T2') =|>
       State' := 7  /\ SND_SG(
                    H(Na'.Kxj.H(Nb'.H(Na'.xor((IDi'.SIDj'),Kug).Kdi.T1').Kxj.T2').T2')
                                 .H(Nb'.Kdi.H(Na'.xor((IDi'.SIDj'),Kug).Kdi.T1').Na'.T1)
                                )
                    /\ secret(IDi',sc_user_id,{Ui, GW})
                    /\ secret(SIDj',sc_sensor_id,{Ui, GW})
end role
```

Box 3

```
role session(Ui, Sj, GW : agent,
             Kdi, Kxj, Kug: symmetric_key,
             H: hash_func,
             P: text
            )
def=
  local  SSU,RSU,
         SSG,RSG,
         SUS,RUS,
         SGS,RGS:channel(dy)
  composition
       user(Ui,Sj,GW,Kdi,Kug,H,P,SUS,RUS)
  /\  sensor(Ui,Sj,GW,Kxj,H,P,SSG,RSG,SSU,RSU)
  /\ gateway(Ui,Sj,GW,Kdi,Kxj,Kug,H,SGS,RGS)

end role
```

Box 4

```
role environment()
def=
  const ui, sj, gw : agent,
        kdi, kxj, kug, kig, kiig: symmetric_key,
        user_sensor_sk: protocol_id,
        h: hash_func,
        p: text
  intruder_knowledge={ui, sj, gw, kig, kiig, h, p}
  composition
           session(ui,sj,gw,  kdi,kxj,kug,h,p)
        /\ session(ui, i,gw,  kdi,kig,kug,h,p)
        /\ session( i,sj,gw,  kig,kxj,kiig,h,p)
end role
```

Box 5

```
goal
% Confidentiality (G12)
secrecy_of sc_sensor_id,sc_user_id

% Message authentication (G2)
authentication_on user_sensor_sk
end goal
```

Box 6

For the role of the session, see Box 4. For the role of the environment, see Box 5.

The role of the goal is divided into two parts. The first part is the "secrecy_of sc_sensor_id,sc_user_id"; this means we want to keep the identity of the user and sensor confidential between them and the gateway. The second part "authentication_on user_sensor_sk" means the authentication of the shared key between a user and a sensor (see Box 6).

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All the authors have contributed equally to this work.

Acknowledgments

The work presented in this paper has been supported by the LifeWear Project (funded by the Spanish Ministry of Industry, Energy and Tourism with Reference TSI-010400-2010-100). The work has also been supported by the Chinese Scholarship Council (CSC) with File no. 201507040027.

References

[1] Y. Choi, D. Lee, and J. Kim, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081–10106, 2014.

[2] W. B. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 730831, 7 pages, 2013.

[3] C.-C. Chang and H.-D. Le, "A Provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 357–366, 2016.

[4] F. Wu et al., "A Novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks," *Security and Communication Networks*, vol. 9, no. 16, pp. 3527–3542, 2016.

[5] A. K. Das et al., "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 16, pp. 3670–3687, 2016.

[6] J. Jung, J. Kim, Y. Choi, and D. Won, "An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 8, article 1299, 2016.

[7] W. Fan et al., "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2016.

[8] R. Amin and G. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, part 1, pp. 58–80, 2016.

[9] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, "A provably-secure ECC-based authentication scheme for wireless sensor networks," *Sensors*, vol. 14, no. 11, pp. 21023–21044, 2014.

[10] Y. Lu, L. Li, H. Peng, and Y. Yang, "An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 837, 2016.

[11] D. Zhao, H. Peng, L. Li, and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 247–269, 2014.

[12] J. L. Hou et al., "Novel Authentication Schemes for IoT Based Healthcare Systems, Novel Authentication Schemes for IoT

Based Healthcare Systems," *International Journal of Distributed Sensor Networks*, Article ID e183659, 2015.

[13] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.

[14] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.

[15] S. Chatterjee and A. K. Das, "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Security and Communication Networks*, vol. 8, no. 9, pp. 1752–1771, 2015.

[16] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer-to-Peer Networking and Applications*, vol. 9, no. 1, pp. 171–192, 2016.

[17] Q. Jiang, N. Kumar, J. Ma, J. Shen, D. He, and N. Chilamkurti, "A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks," *International Journal of Network Management*, vol. 27, no. 3, Article ID e1937, 2017.

[18] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.

[19] J. Nam, K.-K. R. Choo, S. Han, M. Kim, J. Paik, and D. Won, "Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation," *PLoS ONE*, vol. 10, no. 4, Article ID e0116709, 2015.

[20] J. Moon, H. Yang, Y. Lee, and D. Won, "Improvement of user authentication scheme preserving uniqueness and anonymity for connected health care," in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication (IMCOM '17)*, Japan, January 2017.

[21] A. G. Reddy, A. K. Das, E.-J. Yoon, and K.-Y. Yoo, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.

[22] N. Saxena, B. J. Choi, and R. Lu, "Authentication and authorization scheme for various user roles and devices in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 907–921, 2016.

[23] H. Ning, H. Liu, and L. T. Yang, "Aggregated-proof based hierarchical authentication scheme for the internet of things," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 657–667, 2015.

[24] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[25] A. Rossi, S. Pierre, and S. Krishnan, "Secure route optimization for MIPv6 using enhanced CGA and DNSSEC," *IEEE Systems Journal*, vol. 7, no. 3, pp. 351–362, 2013.

[26] V. Odelu, A. K. Das, and A. Goswami, "SEAP: secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 30–38, 2016.

[27] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.

[28] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[29] P. Kumar, A. Gurtov, M. Ylianttila, S.-G. Lee, and H. J. Lee, "A strong authentication scheme with user privacy for wireless sensor networks," *ETRI Journal*, vol. 35, no. 5, pp. 889–899, 2013.

[30] M. K. Khan and S. Kumari, "An improved user authentication protocol for healthcare services via wireless medical sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 347169, 10 pages, 2014.

[31] J. Moon, Y. Choi, J. Jung, and D. Won, "An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards," *PLoS ONE*, vol. 10, no. 12, Article ID e0145263, 2015.

[32] M. Alizadeh et al., "Cryptanalysis and improvement of a secure password authentication mechanism for seamless handover," *PLOS One*, vol. 10, no. 11, Article ID e0142716, 2015.

[33] A. K. Das, A. K. Sutrala, V. Odelu, and A. Goswami, "A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks," *Wireless Pers Commun*, pp. 1–35, 2016.

[34] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[35] K. H. Rosen, *Elementary number theory and its applications*, Addison-Wesley Publishing Company, Advanced Book Program, Reading, MA, Second edition, 1988.

[36] D. He, N. Kumar, M. K. Khan, and J.-H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 811–817, 2013.

[37] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1646–1656, 2012.

[38] A. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2933, 2017.

[39] Y. Chung, S. Choi, Y. S. Lee, N. Park, and D. Won, "An enhanced lightweight anonymous authentication scheme for a scalable localization roaming service in wireless sensor networks," *Sensors*, vol. 16, no. 10, article 1653, 2016.

[40] Commercial National Security Algorithm Suite and Quantum Computing FAQ U.S. National Security Agency, January 2016.

[41] M. Burrows, M. Abad, and M. Needham, "A logic of authentication," *Proceedings of the Royal Society A Mathematical, Physical and Engineering Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[42] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Computer Aided Verification: International Conference on Computer Aided Verification*, vol. 3576, pp. 281–285, Springer, Berlin, Germany, 2005.

[43] 2017, https://www.miracl.com/.

[44] 2017, https://github.com/miracl/MIRACL.

[45] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[46] J. Rodríguez-Molina, J.-F. Martínez, P. Castillejo, and L. López, "Combining wireless sensor networks and semantic middleware for an internet of things-based sportsman/woman monitoring application," *Sensors*, vol. 13, no. 2, pp. 1787–1835, 2013.