RESEARCH ARTICLE

Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments

Chi-Tung Chen[1], Cheng-Chi Lee[2,3]*, Iuon-Chang Lin[3,4]

[1] Department of Distribution Management (Information Management), National Chin-Yi University of Technology, Taichung, Taiwan, China, [2] Department of Library and Information Science, Research and Development Center for Physical Education, Health, and Information Technology, Fu Jen Catholic University, New Taipei City, Taiwan, China, [3] Department of Photonics and Communication Engineering, Asia University, Taichung, Taiwan, China, [4] Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan, China

* cclee@mail.fju.edu.tw

OPEN ACCESS

Citation: Chen C-T, Lee C-C, Lin I-C (2020) Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments. PLoS ONE 15(4): e0232277. https://doi.org/10.1371/journal.pone.0232277

Editor: He Debiao, Wuhan University, CHINA

Received: February 14, 2020

Accepted: April 10, 2020

Published: April 30, 2020

Data Availability Statement: All relevant data are within the paper.

Funding: This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 108-2410-H-030-074. All funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

Abstract

In the Internet of Things (IoT), numerous devices can interact with each other over the Internet. A wide range of IoT applications have already been deployed, such as transportation systems, healthcare systems, smart buildings, smart factories, and smart cities. Wireless sensor networks (WSNs) play crucial roles in these IoT applications. Researchers have published effective (but not entirely secure) approaches for merging WSNs into IoT environments. In IoT environments, the security effectiveness of remote user authentication is crucial for information transmission. Computational efficiency and energy consumption are crucial because the energy available to any WSN is limited. This paper proposes a notably efficient and secure authentication scheme based on temporal credential and dynamic ID for WSNs in IoT environments. The Burrows–Abadi–Needham (BAN) logic method was used to validate our scheme. Cryptanalysis revealed that our scheme can overcome the security weaknesses of previously published schemes. The security functionalities and performance efficiency of our scheme are compared with those of previous related schemes. The result demonstrates that our scheme's security functionalities are quantitatively and qualitatively superior to those of comparable schemes. Our scheme can improve the effectiveness of authentication in IoT environments. Notably, our scheme has superior performance efficiency, low computational cost, frugal energy consumption, and low communication cost.

1. Introduction

Internet of Things (IoT) is an emerging technology, which is the extension of Internet connectivity into various devices such as sensors, vehicles, and mobile phones. These devices can interact with each other over the Internet [1]. A wide range of applications connecting objects that can communicate with each other have been deployed; applications include transportation systems, healthcare systems, smart buildings, smart factories, and smart cities [1, 2].

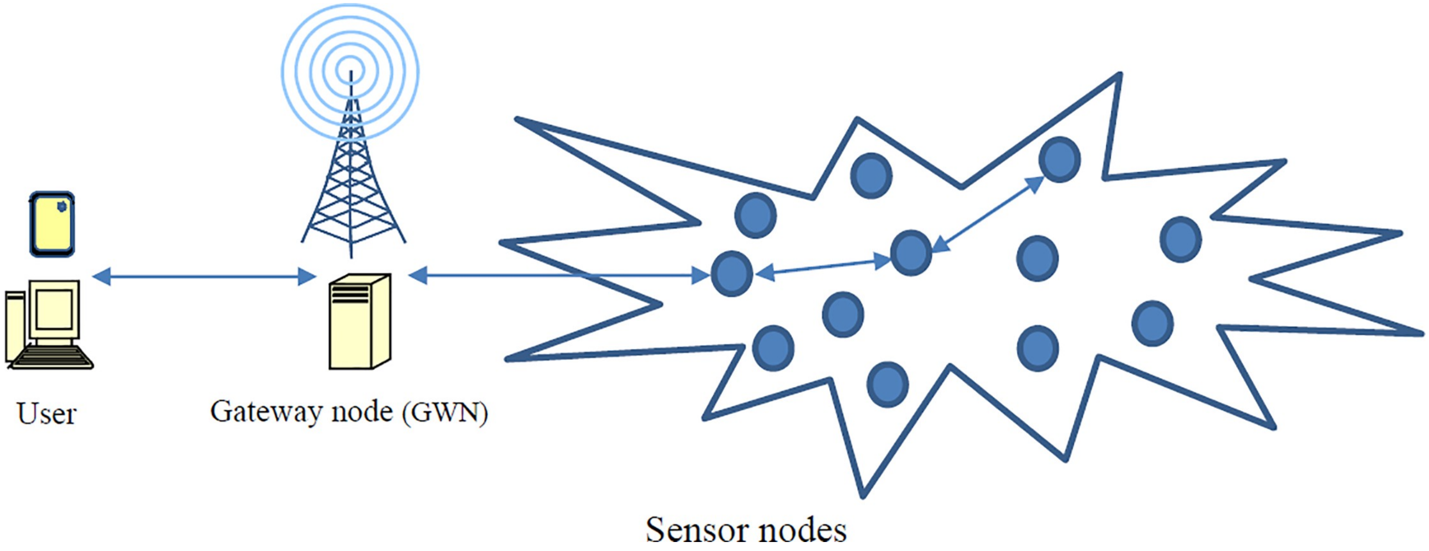

Fig 1. Wireless sensor networks.

https://doi.org/10.1371/journal.pone.0232277.g001

Wireless sensor networks (WSNs) are crucial in these IoT applications [2, 3]. WSNs have become increasingly used in providing services for monitoring environments and activities because of their low cost, flexibility, ease of deployment, and wide range of applications (Fig 1) [4]. As illustrated in Fig 1, WSNs comprise numerous sensor nodes scattered arbitrarily over a certain region. Sensor nodes can sense, process, and transmit information (e.g., temperature and traffic information). Remote users are required to reach a specific sensor node via the gateway node (GWN) [5, 6]. Each scattered sensor node can collect data and route data back to the GWN. Remote users may communicate with a GWN through the Internet. When data from WSNs are made available to users, the legitimacy of each user must be verified before the system can grant access to the data, and the sensor nodes reserved for access must be confirmed to be legitimate. Hence, remote user authentication is necessary and critical for secure information transmission in WSNs [2, 5, 7, 8]. The following basic design criteria must be considered when designing a remote user authentication scheme for WSNs [2, 5, 8]:

1. *Mutual authentication*. Users and sensor nodes must mutually authenticate each other. After they have authenticated each other, they must arrange a session key for information transmission.
2. *Masquerade attack resistance*. An adversary cannot impersonate a legal user to log in to WSNs. In addition, the adversary cannot masquerade as a sensor node to spoof the user.
3. *Replay attack resistance*. The adversary cannot attempt to replay previously intercepted messages to spoof the GWN.
4. *Guessing attack resistance*. The adversary cannot obtain useful information to devise an offline check of the correctness of guessed passwords.

1.1. Preliminaries and technical background

In this subsection, we introduce some preliminaries and the principal technologies that our scheme is based on, such as temporal credential [7] and dynamic ID [9, 10].

A temporal credential is an impermanent attestation of authority issued by a third party. The GWN can issue a temporal credential to each user and sensor node [7]. The expiration time of a user's temporal credential is regulated by the GWN. A user's temporal credential is related to the identity of user and can be securely stored in a smart card. The temporal credential of a sensor node is also related to its identity and confidentially written in its storage. Based on the issuing and signing of temporal credential, the mutual authentication between the user and the GWN is achieved through the verification of temporal credential for the user. The mutual authentication between the sensor node and the GWN is achieved by the verification of the temporal credential for the sensor node.

Each dynamic ID is temporarily assigned by the system and mapped to a specific user [9]. A dynamic ID is a combination of its user's information and a random nonce. The random nonce is an arbitrary number; it is used only once during the communication. In the authentication process, the login message of the user i contains a dynamic ID, called DID_i. The login message is dynamic for each login. For all i, the parameter DID_i is associated with nonce N_i and changed dynamically for each login. The use of a dynamic ID in each login message can avoid the risk of ID-theft [10]. Our scheme introduces dynamic ID to anonymize users.

1.2. Motivation and contribution

Typical IoT installations allow remote users to access data from sensor nodes in WSNs through the Internet. Researchers have been developing effective approaches for merging WSNs into IoT environments [2, 11–16]. Because of the resource constraints of sensor nodes, to design an efficient and secure authentication scheme for WSNs in IoT environments constitutes a nontrivial challenge. In IoT environments, the security effectiveness of remote user authentication is crucial for trustworthy information transmission [2, 3]. Computational efficiency and energy consumption are crucial because of the limited energy resources of WSNs [2, 3]. Moreover, time synchronization is a critical and challenging problem for WSNs; the system must provide a synchronized logical time clock for all devices and objects in IoT environments [3, 17–19]. Any adversary and any malicious node in IoT environments can attack clock synchronization [3, 17]. The communication errors, frequent topological changes, low-cost clocks, and limited energy levels of IoT nodes are other factors that can affect time synchronization [18, 19]. A timestamp-based authentication scheme requires trustworthy timestamps and synchronized time clocks to verify any device's legitimacy. When a system has a serious time synchronization problem, no device can be synchronized with any another device, and thus the system cannot verify any device's legitimacy. Therefore, any serious time synchronization failure causes mutual authentication failure. The time synchronization problem should be contemplated as designing a remote user authentication scheme for WSNs in IoT environments [3, 17–19]. Moreover, when a given user's ID is revealed, an adversary can determine any information concerning the user's identity and monitor the user's activities. An exposed user ID is also useful to the adversary because it provides login information [10]. Therefore, anonymous access for each login should be required. Although several previously published studies have proposed diverse remote user authentication schemes, they have been neither highly secure nor efficient sufficiently to satisfy the requirements of WSNs in IoT environments (Related work in Section 2). This paper proposes a more efficient and secure authentication scheme for WSNs in IoT environments to ameliorate these security weaknesses.

The major contributions of our work are as follows:

1. We propose a new three-party scheme on the basis of temporal credential [7] and dynamic ID [9, 10] for WSNs in IoT environments to achieve security, mutual authentication, and session key agreement. Cryptanalysis revealed that the security functionalities of the proposed scheme qualitatively and quantitatively superior to those of previous schemes; the

proposed scheme can advance the field of authentication schemes. The Burrows–Abadi–Needham (BAN) logic method [3, 20–24] was used to validate our scheme.

2. The proposed scheme performs efficiently in IoT environments, with low computational cost, frugal energy consumption, and little communication cost.

3. Our scheme uses temporal credentials and random nonce instead of the timestamps to verify mutual authentication among U_i, the *GWN*, and S_j. Therefore, our scheme can avoid the time synchronization problem for WSNs in IoT environments [3, 9, 17, 25]. Moreover, dynamic ID technology [9, 10] is applied in our scheme. User identities are consequently anonymous and can be confirmed only by the service provider.

1.3. Organization of the paper

The remainder of this paper is organized as follows: Section 2 introduces a brief review of the related work in WSNs and explains the security weaknesses of the Ostad-Sharif et al. scheme [2] for WSNs in IoT environments; Section 3 details the proposed efficient secure authentication scheme for WSNs in IoT environments; Section 4 presents the security analysis of the proposed scheme; Section 5 discusses the effectiveness and efficiency of the proposed scheme; and finally, Section 6 presents the study's conclusion.

2. Related work in WSNs

To satisfy the security requirements of WSNs, many remote user authentication schemes have been proposed. In 2004, Benenson et al. [26] described the security issues of user authentication in WSNs and proposed a protocol for them, in which the user can achieve successful authentication with any subset of sensors from a set of *n* sensors (*n* being the average number of sensors within a broadcast distance of the user). Watro et al. [27] proposed a TinyPK authentication protocol with the Rivest-Shamir-Adleman (RSA) public key cryptosystem [28] and Diffie-Hellman key agreement algorithm [29]. However, this authentication protocol has the disadvantage of the masquerade attack, in which an adversary can masquerade as a sensor node to spoof the user [5]. Wong et al. [30] proposed a less complex lightweight user authentication protocol for WSNs by using hash function operations. However, the scheme cannot protect against stolen-verifier, replay, and forgery attacks [5, 31]. Moreover, the passwords in the scheme can be revealed easily by any of the sensor nodes, and users cannot change their passwords freely. In 2009, to eliminate the weaknesses of the Wong et al. scheme, Das [5] proposed a two-factor user authentication scheme for WSNs. The scheme implements password-based authentication with the assistance of a GWN to access resource-constrained sensor nodes. However, this scheme is vulnerable to insider, masquerade, offline password-guessing, stolen smart card, and GWN bypassing attacks [7, 8, 32]. The scheme does not provide mutual authentication, a key agreement, and a password change phase for users to change or update their password [7, 8, 32]. Khan et al. [32], Chen et al. [33], and Yeh et al. [8] have subsequently proposed new schemes for improving the inherent security weaknesses of the Das scheme. Khan et al. [32] proposed a user authentication scheme for rectifying the susceptibilities of the Das scheme and achieving a more secure user authentication in WSNs. Afterward, Chen et al. [33] provided a secrecy-improved mutual user authentication scheme for WSNs by applying hash functions. Yeh et al. [8] proposed a new mutual user authentication protocol by using elliptic curves cryptography (ECC) and smart cards for WSNs. Xue et al. [7] showed that the Khan et al. scheme is vulnerable to stolen smart card and GWN bypassing attacks. In addition, the Chen et al. scheme is vulnerable to insider, masquerade, stolen smart card, and GWN bypassing attacks [7]. By contrast, the Yeh et al. scheme is vulnerable to stolen smart card and replay attacks [7]. Xue et al. [7] proposed a temporal-credential-based mutual authentication scheme for users, GWNs, and sensor nodes. With the assistance of password-based authentication, the GWN in the Xue et al. scheme can issue a temporal credential to each user and sensor node. However, the Xue et al. scheme is vulnerable to insider attacks and stolen smart card attacks [34]; the scheme does not offer password protection [34]. In 2016, Chang et al. [35] proposed a flexible authentication scheme for WSNs which operates in two modes. The first mode provides a lightweight authentication scheme, and the second mode is an advanced protocol based on ECC. In 2018, Amin et al. [34] demonstrated that the Chang et al. scheme is insecure against stolen smart card attack and cannot provide password protection. Amin et al. [34] then proposed a robust authentication scheme using smartcards for WSNs. However, the Amin et al. scheme has higher energy consumption, computational costs, and communication costs than those published previously (Section 5) [34]. In healthcare applications, Challa et al. [36] proposed a secure user authentication scheme for wireless healthcare sensor networks. The three factor authentication scheme is designed with ECC. The proposed scheme has several functionality features including dynamic sensor node addition, password updates, biometrics updates, and smart card revocation for WSNs. On the basis of ECC, Li et al. [3] also proposed an anonymous authentication scheme for WSNs in IoT environments. In the scheme, they used fuzzy commitment scheme [3] to handle user biometric information. In 2019, Harbi et al. [37] proposed an ECC-based mutual authentication scheme to secure communication in IoT-enabled WSNs. The sensor network in the system is arranged into clusters to diminish the energy consumption of sensors. Each cluster has a cluster head, which is a leader sensor node. However, Challa et al. scheme, Li et al. scheme, and Harbi et al. scheme are all based on an ECC for WSNs. The ECC approach is a public key cryptography approach based on elliptic curves. According to a related study, the time cost of an ECC point multiplication is much larger than that of hash function operations [2, 3, 7, 34, 35], and the energy consumption for executing an asymmetric ECC cryptosystem is much higher than that for executing a hash function [38, 39].

Currently, researchers are designing effective remote user authentication schemes for WSNs in IoT environments. In 2019, Ostad-Sharif et al. [2] proposed an efficient user authentication scheme and claimed that their scheme is appropriate for WSNs in IoT environments. However, in this section, we argue that the login and authentication phase of the Ostad-Sharif et al. scheme has design faults. Moreover, their scheme cannot provide password change and update a password in its password change phase. Their scheme also has the time synchronization problem [3, 17–19]. The details are presented as follows.

2.1. Authentication design faults of the Ostad-Sharif et al. scheme in IoT environments

Design faults exist in the login and authentication phase of the Ostad-Sharif et al. scheme [2]. We illustrate this security weakness in the subsequent passages. When a registered user U_i wants to access the information of sensor node S_j, the login and authentication phase of the Ostad-Sharif et al. scheme must be executed in advance. At first, a registered user U_i inserts a smart card into the smart card reader and imprints his/her fingerprint B_i on the sensor device. The smart card contains the secret parameters $\{D_i, C_i, E_i, SCN_i, BK()\}$, in which SCN_i denotes unique smart card number and $BK()$ denotes biometric key generation/extraction function. The smart card reader first extracts masked biometric C_i from the smart card and computes $RN'_i = BK(\mathrm{h}(B_i)) \oplus C'_i$. After finding C'_i, the smart card reader must validate whether C'_i and C_i are equal. If $C'_i \neq C_i$, then the smart card reader terminates the request. However, in the equation above, the smart card reader does not know random number RN'_i and masked biometric C'_i. Therefore, it cannot obtain RN'_i and C'_i from the equation. Finally, a legitimately registered user U_i cannot pass the verification to access the system. This problem will happen to all legitimately registered users. The Ostad-Sharif et al. scheme has design faults in the login and authentication phase.

2.2. Failure to provide password change capability in the Ostad-Sharif et al. scheme

The Ostad-Sharif et al. scheme [2] cannot provide password change capability. We demonstrate this weakness in the following passages. When a registered user U_i wants to update the password PW_i, the password change phase in the scheme must be executed. U_i first inserts a smart card into the smart card reader. He or she then inputs identity ID_i and password PW_i. The smart card contains the secret parameters $\{D_i, C_i, E_i, SCN_i, BK()\}$. After the legitimacy of U_i is verified, U_i enters a new password PW_i^{new}. The smart card computes the following equations:

1. $RPW_i^{new} = \mathrm{h}(\mathrm{ID_i}\|\ PW_i^{new}\|\mathrm{RN_i})$, in which $\mathrm{RN_i}$ denotes random number.
2. $A'_i = D_i \oplus RPW_i$
3. $D_i^{new} = A_i^{new} \oplus RPW_i$
4. $L'_i = E_i \oplus RPW_i$
5. $E_i^{new} = \mathrm{L'_i} \oplus RPW_i$

After D_i^{new} and E_i^{new} have been found, the smart card replaces the secret parameters $\{D_i, E_i\}$ in the smart card with the new parameters $\{D_i^{new}, E_i^{new}\}$. The smart card finally contains the parameters $\{D_i^{new}, C_i, E_i^{new}, SCN_i, BK()\}$. However, in (3), the smart card does not know A_i^{new}; hence, it cannot obtain the new parameter D_i^{new} from (3). Moreover, from (4) and (5), we obtain the following results:

$$E_i^{new} = L'_i \oplus RPW_i = E_i \oplus RPW_i \oplus RPW_i = E_i$$

Finally, the value of the new parameter E_i^{new} is the same as the value of the parameter E_i, and the new parameter D_i^{new} cannot be acquired from the equations. Therefore, a registered user U_i cannot update his/her password. The Ostad-Sharif et al. scheme fails to provide password change capability.

2.3. Time synchronization and authentication problem of the Ostad-Sharif et al. scheme in IoT environments

The Ostad-Sharif et al. scheme uses a timestamp T_i to verify mutual authentication among U_i, the GWN, and S_j for WSNs in IoT environments. Therefore, the Ostad-Sharif et al. scheme must provide synchronized time clocks to all devices in IoT environments for timestamp comparison [3, 17, 18]. However, as mentioned, both adversaries and malicious nodes can attack time synchronization [17]. Frequent topological changes, low-cost clocks, and limited energy of the sensor nodes in IoT environments can also affect time synchronization [18, 19]. The time synchronization of all WSN devices in IoT environments is a nontrivial challenge in itself [3, 17–19]. When a serious time synchronization problem arises in Ostad-Sharif et al. scheme, the GWN, U_i, and S_j cannot be synchronized with each other and then the legitimacy values of the GWN, U_i, and S_j cannot be verified. Hence, the Ostad-Sharif et al. scheme may enter a state such that mutual authentication among the GWN, U_i, and S_j cannot be achieved [3, 17, 18].

3. Proposed scheme

In this section, we propose an efficient and secure authentication scheme for WSNs in IoT environments. The WSN environment contains three participants: the user (U_i), sensor node (S_j), and gateway node (GWN). The scheme applies dynamic ID to achieve security and user anonymity (identity protection) [9, 10]. The scheme applies temporal credential to achieve mutual authentication and session key agreement [7]. Temporal credentials are securely protected and stored in smart cards. The scheme can withstand stolen smart card attacks (Section 4.2). The system protects passwords against off-line password guessing attacks (Section 4.2). The system need not maintain any password or verification table; therefore it can resist the stolen verifier attacks and insider attacks [9, 40, 41]. The scheme can withstand masquerade attacks, replay attacks, GWN bypassing attacks, and GWN spoofing attacks (Section 4.4 and 4.8). Before the registration, users are not obliged to share their IDs and passwords with the GWN; hence, the scheme provides a convenient functionality of adding new users (Section 4.6). To solve the password-changing problem in previous schemes, we also introduce a new password change phase to update the password. In the new password change phase, U_i can freely select and update the password without requiring the communication with any other participants (the GWN and S_j), such that it can avoid additional communication message overhead (Fig 5) [42]. Hash function is operated in our scheme for providing security and computational efficiency. Table 1 lists the definition of the notations in our scheme. The GWN chooses the private keys $K_{GWN\text{-}U}$ and $K_{GWN\text{-}S}$, and only the GWN knows them. The proposed scheme consists of four phases: (1) registration phase, (2) login phase, (3) authentication and key agreement phase, and (4) password change phase. They are described as follows:

3.1. Registration phase

The registration phase comprises two parts, one for users and the other for sensor nodes. We first describe the registration phase for users. In this phase, when a new user U_i undertakes to register, he or she selects the identification ID_i and password PW_i. Subsequently, U_i generates a random number r_i and sends ID_i and $h(r_i \oplus PW_i)$ to the GWN for registration through a secure channel. After receiving the messages from U_i, the GWN selects the expiration time TE_i of the temporal credential of U_i. The GWN computes the temporal credential TC_i and verification information R_i for U_i. The GWN then issues a smart card with the temporal credential TC_i, expiration time TE_i, and verification information R_i to U_i through a secure channel. The steps are detailed as follows (Fig 2):

Step U1. U_i freely chooses identification ID_i and password PW_i.

Step U2. U_i generates a random number r_i and calculates $h(r_i \oplus PW_i)$.

Step U3. $U_i \Rightarrow GWN$: $\{h(r_i \oplus PW_i), ID_i\}$.

U_i transmits $h(r_i \oplus PW_i)$ and ID_i to the GWN through a secure channel.

Step U4. $GWN \Rightarrow U_i$: $\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, h(.)\}$. After receiving the message from U_i, the GWN selects the expiration time TE_i of the temporal credential of U_i and computes the following equations to issue the temporal credential TC_i for U_i.

$P_i = h(ID_i || ID_{GWN} || TE_i)$, $TC_i = h(P_i || K_{GWN\text{-}U} || TE_i)$, $PTC_i = TC_i \oplus h(r_i \oplus PW_i)$,
$Q_i = h(ID_i || K_{GWN\text{-}U})$, $B_i = Q_i \oplus h(ID_i || h(r_i \oplus PW_i))$, and $R_i = h(Q_i)$.

The GWN then issues a smart card with the secret parameters $\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, h(.)\}$ to U_i through a secure channel.

Step U5. U_i stores r_i in the smart card, after which the smart card holds the parameters $\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, r_i, h(.)\}$.

Table 1. Notation definitions.

Notation	Definition
U_i	The *i*th user
S_j	The *j*th sensor node
GWN	The gateway node
ID_i	The identification of U_i
ID_{GWN}	The identification of the *GWN*
SID_j	The identification of S_j
DID_i	The dynamic ID of U_i
DID_{GWN}	The dynamic ID of the *GWN*
PW_i	The password of U_i
PW_j	The password of S_j
BK	Biometric key generation/extraction function
B_i	Biometric of U_i
SCN_i	Unique smart card number
$K_{GWN\text{-}U}$	Private key only known to the GWN
$K_{GWN\text{-}S}$	Private key only known to the GWN
KEY_{ij}	Shard session key between U_i and S_j
TC_i	Temporal credential issued by the GWN to U_i
TC_j	Temporal credential issued by the GWN to S_j
TE_i	Expiration time of a user's temporal credential
TS	Timestamp value
||	String concatenation manipulation
$\rightarrow$	Common channel[a]
$\oplus$	Exclusive-or manipulation
$\Rightarrow$	Secure channel[b]
h(•)	One-way hash function[c]

[a] A common channel is a channel allocated in common to participants.

[b] A secure channel is a channel of delivering messages that can withstand tampering and overhearing.

[c] A hash function has a one-way property that it is computationally infeasible to find a data object to map to a hash result [43].

https://doi.org/10.1371/journal.pone.0232277.t001

We now describe the registration phase for sensor nodes. In this phase, each sensor node S_j is pre-configured with SID_j. After deployment, the sensor node S_j generates a random number r_j and then sends SID_j and $h(r_j \oplus SID_j)$ to the *GWN* for registration through a secure channel. After receiving the messages from S_j, the *GWN* issues a temporal credential TC_j to S_j through a secure channel. The steps are detailed as follows (Fig 3):

Step S1. S_j is pre-configured with SID_j.

Step S2. S_j generates a random number r_j and computes $h(r_j \oplus SID_j)$.

Step S3. $S_j \Rightarrow GWN$: $\{SID_j, h(r_j \oplus SID_j)\}$.

S_j sends SID_j and $h(r_j \oplus SID_j)$ to the *GWN* through a secure channel.

Step S4. $GWN \Rightarrow S_j$: $\{RTC_j\}$. After receiving the message from S_j, the *GWN* computes $TC_j = h(K_{GWN\text{-}S}||SID_j)$ to issue the temporal credential TC_j for S_j and then calculates $RTC_j = TC_j \oplus h(h(r_j \oplus SID_j)||SID_j)$. The *GWN* sends RTC_j to S_j through a secure channel.

Step S5. After receiving the message from the *GWN*, S_j computes $TC_j = RTC_j \oplus h(h(r_j \oplus SID_j)||SID_j)$ to find its temporal credential TC_j and then stores it.

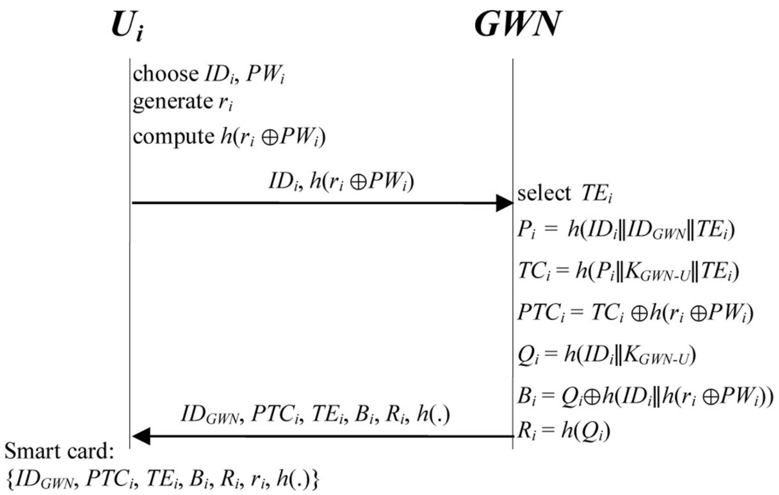
U_i
GWN
choose ID_i, PW_i
generate r_i
compute $h(r_i \oplus PW_i)$
$ID_i, h(r_i \oplus PW_i)$
select TE_i
$P_i = h(ID_i \| ID_{GWN} \| TE_i)$
$TC_i = h(P_i \| K_{GWN\text{-}U} \| TE_i)$
$PTC_i = TC_i \oplus h(r_i \oplus PW_i)$
$Q_i = h(ID_i \| K_{GWN\text{-}U})$
$B_i = Q_i \oplus h(ID_i \| h(r_i \oplus PW_i))$
$R_i = h(Q_i)$
$ID_{GWN}, PTC_i, TE_i, B_i, R_i, h(.)$
Smart card:
$\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, r_i, h(.)\}$

Fig 2. Registration phase for users in the proposed scheme.

https://doi.org/10.1371/journal.pone.0232277.g002

3.2. Login phase

U_i first inserts a smart card into the smart card reader to log in to the system. U_i then gives (ID_i, PW_i) that correspond to the smart card. The smart card of U_i computes verification information R_i^* and then verifies it with the stored R_i in the smart card. After passing verification, the legitimacy of U_i is ensured. Afterward, U_i can read the information stored in the smart card and find its temporal credential TC_i. The steps are detailed as follows (Fig 4):

Step L1. User U_i inserts a smart card into the smart card reader and provides keys (ID_i, PW_i). The smart card of user U_i then computes $Q_i = B_i \oplus h(ID_i \| h(r_i \oplus PW_i))$ and $R_i^* = h(Q_i)$. The smart card validates whether R_i^* and the stored R_i in the smart card are equal. If the values are unequal, the smart card rejects the login request. Otherwise, the legitimacy of U_i is ensured, and U_i can read the information stored in the smart card.

Step L2. U_i computes $TC_i = PTC_i \oplus h(r_i \oplus PW_i)$ to find its temporal credential TC_i.

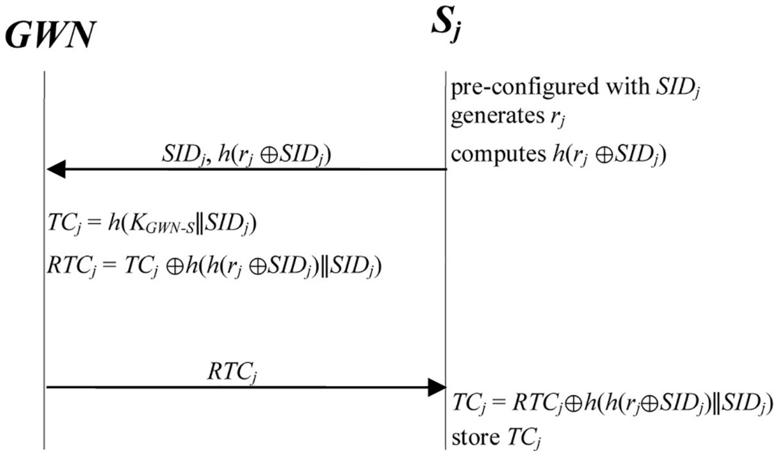
GWN
S_j
pre-configured with SID_j
generates r_j
computes $h(r_j \oplus SID_j)$
$SID_j, h(r_j \oplus SID_j)$
$TC_j = h(K_{GWN\text{-}S} \| SID_j)$
$RTC_j = TC_j \oplus h(h(r_j \oplus SID_j) \| SID_j)$
RTC_j
$TC_j = RTC_j \oplus h(h(r_j \oplus SID_j) \| SID_j)$
store TC_j

Fig 3. Registration phase for sensor nodes in the proposed scheme.

https://doi.org/10.1371/journal.pone.0232277.g003

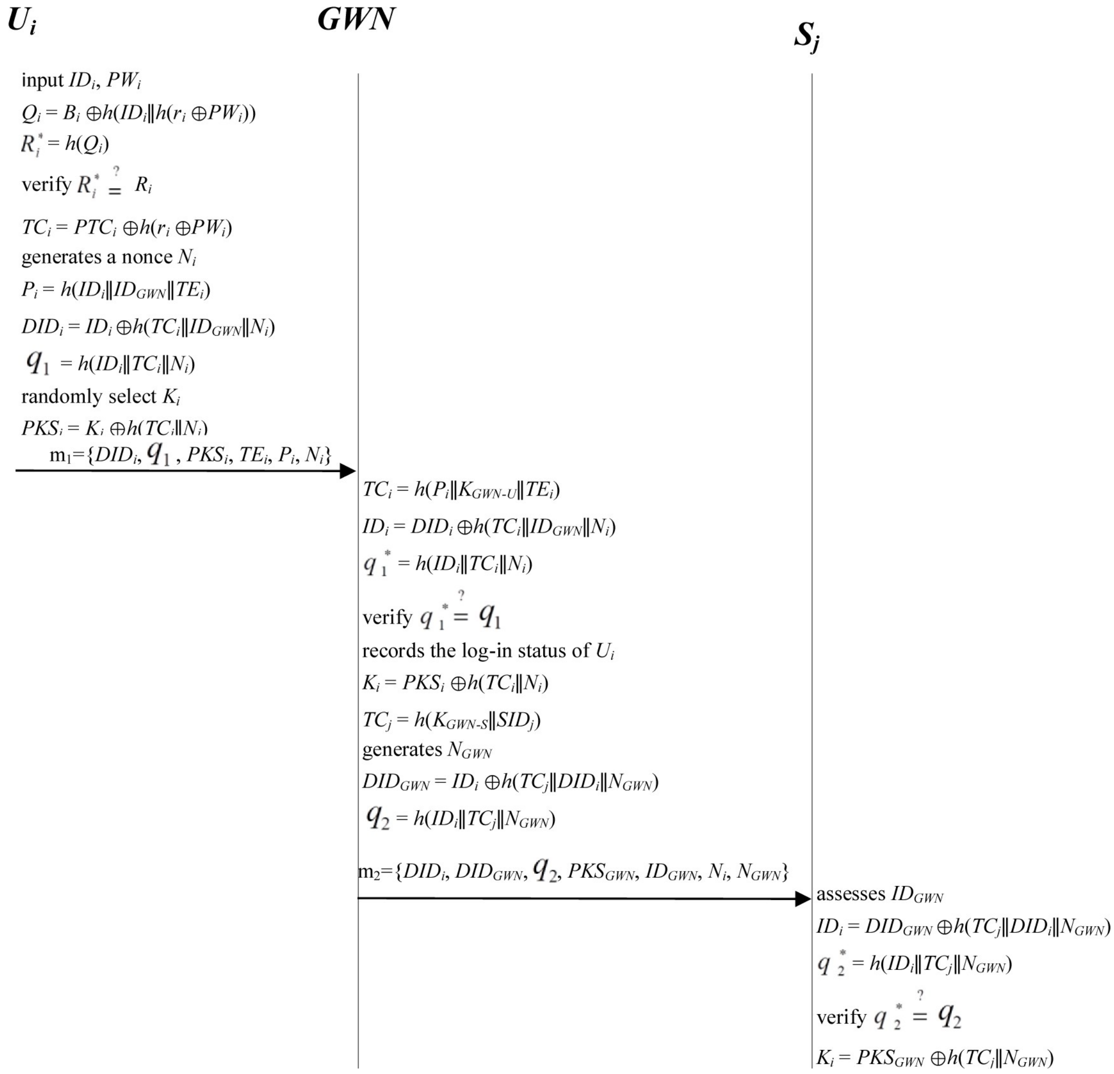
U_i
GWN
S_j
input ID_i, PW_i
$Q_i = B_i \oplus h(ID_i\|h(r_i \oplus PW_i))$
$R_i^* = h(Q_i)$
verify $R_i^* \overset{?}{=} R_i$
$TC_i = PTC_i \oplus h(r_i \oplus PW_i)$
generates a nonce N_i
$P_i = h(ID_i\|ID_{GWN}\|TE_i)$
$DID_i = ID_i \oplus h(TC_i\|ID_{GWN}\|N_i)$
$q_1 = h(ID_i\|TC_i\|N_i)$
randomly select K_i
$PKS_i = K_i \oplus h(TC_i\|N_i)$
$m_1=\{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$
$TC_i = h(P_i\|K_{GWN\text{-}U}\|TE_i)$
$ID_i = DID_i \oplus h(TC_i\|ID_{GWN}\|N_i)$
$q_1^* = h(ID_i\|TC_i\|N_i)$
verify $q_1^* \overset{?}{=} q_1$
records the log-in status of U_i
$K_i = PKS_i \oplus h(TC_i\|N_i)$
$TC_j = h(K_{GWN\text{-}S}\|SID_j)$
generates N_{GWN}
$DID_{GWN} = ID_i \oplus h(TC_j\|DID_i\|N_{GWN})$
$q_2 = h(ID_i\|TC_j\|N_{GWN})$
$m_2=\{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$
assesses ID_{GWN}
$ID_i = DID_{GWN} \oplus h(TC_j\|DID_i\|N_{GWN})$
$q_2^* = h(ID_i\|TC_j\|N_{GWN})$
verify $q_2^* \overset{?}{=} q_2$
$K_i = PKS_{GWN} \oplus h(TC_j\|N_{GWN})$

Fig 4. Login phase; authentication and key agreement phase.

https://doi.org/10.1371/journal.pone.0232277.g004

3.3. Authentication and key agreement phase

After ensuring the legitimacy of U_i and finding the temporal credential TC_i, the system must complete mutual authentication among U_i, the GWN, and S_j. The first step of the mutual authentication phase involves identity verification for U_i, which is conducted by the GWN. Afterward, the second step entails identity verification of the GWN, which is conducted by S_j. The third step involves identity verification for S_j, which is conducted by U_i as well as the GWN. Finally, a session key KEY_{ij} is negotiated between U_i and S_j to conduct encryption during data transmission later on. The steps are detailed as follows (Fig 4):

Step V1. $U_i \rightarrow GWN$: $\{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$. U_i generates a nonce N_i and computes $P_i = h(ID_i||ID_{GWN}||TE_i)$, $DID_i = ID_i \oplus h(TC_i||ID_{GWN}||N_i)$, and $q_1 = h(ID_i||TC_i||N_i)$. Afterward, U_i randomly chooses a secret sharing key K_i and computes $PKS_i = K_i \oplus h(TC_i||N_i)$. After computation, U_i sends the login request message $m_1 = \{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$ to the GWN.

Step V2. $GWN \rightarrow S_j$: $\{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$. After obtaining message m_1, the GWN computes $TC_i = h(P_i||K_{GWN\text{-}U}||TE_i)$, $ID_i = DID_i \oplus h(TC_i||ID_{GWN}||N_i)$, and $q_1^* =$

$h(ID_i||TC_i||N_i)$. The *GWN* then verifies whether q_1^* and q_1 are equal. If $q_1^* \neq q_1$, then the *GWN* terminates the request and sends a reject message to U_i. Otherwise, the legitimacy of U_i is ensured, and the *GWN* accepts the login request. The *GWN* then records the login status of U_i to indicate that *Ui* is logging in to the system. The *GWN* computes $K_i = PKS_i \oplus h(TC_i||N_i)$. At this point, the *GWN* selects a proper sensor node S_j with identification SID_j and calculates its temporal credential $TC_j = h(K_{GWN\text{-}S}||SID_j)$. The *GWN* then generates a nonce N_{GWN} and computes $DID_{GWN} = ID_i \oplus h(TC_j||DID_i||N_{GWN})$, $q_2 = h(ID_i||TC_j||N_{GWN})$, and $PKS_{GWN} = K_i \oplus h(TC_j||N_{GWN})$. After computation, the *GWN* sends the message $m_2 = \{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$to S_j.

Step V3. $S_j \rightarrow U_i$, *GWN*: $\{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$. After receiving message m_2, S_j assesses ID_{GWN} to verify whether the *GWN* is a participant. If verification is true, S_j computes $ID_i = DID_{GWN} \oplus h(TC_j||DID_i||N_{GWN})$ and $q_2^* = h(ID_i||TC_j||N_{GWN})$. S_j then verifies whether q_2^* and q_2 are equal. If $q_2^* \neq q_2$, then S_j terminates the request and returns a reject message. Otherwise, the legitimacy of the *GWN* is ensured, and S_j accepts the request. S_j computes $K_i = PKS_{GWN} \oplus h(TC_j||N_{GWN})$. Afterward, S_j randomly selects a secret sharing key K_j. S_j computes $q_3 = h(ID_i||SID_j||K_i||N_i||N_{GWN})$ and $PKS_j = K_j \oplus h(K_i||N_i||N_{GWN})$. After computation, S_j sends the message $m_3 = \{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$to U_i and the *GWN*.

Step V4. After receiving the message m_3, U_i and the *GWN* separately compute $q_3^* = h(ID_i||SID_j||K_i||N_i||N_{GWN})$. After computation, the *GWN* verifies whether q_3^* and q_3 are equal. If $q_3^* = q_3$, then the *GWN* can verify the legitimacy of S_j. User U_i also verifies whether q_3^* and q_3 are equal. If $q_3^* = q_3$, then U_i can verify the legitimacy of S_j and the *GWN*. Afterward, U_i and the *GWN* separately compute $K_j = PKS_j \oplus h(K_i||N_i||N_{GWN})$. Finally, after ending the mutual authentication phase, U_i, the *GWN*, and S_j separately generate the shared session key KEY_{ij} by computing $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$.

3.4. Password change phase

To update or change the password, a user U_i must insert his/her smart card into the smart card reader. Afterward, U_i gives ID_i and PW_i, which correspond to the smart card. In the first step of the password change phase, the smart card of U_i computes verification information R_i^* and then verifies it with the stored R_i in the smart card. After passing verification, the legitimacy of U_i is ensured. U_i can then read the information stored in the smart card. The second step involves finding the updated value of the parameters $\{PTC_i^{new}, B_i^{new}, r_i^{new}\}$. Finally, the smart card replaces the old value of the parameters $\{PTC_i, B_i, r_i\}$ in the smart card with the updated value of the parameters $\{PTC_i^{new}, B_i^{new}, r_i^{new}\}$. The steps are detailed as follows (Fig 5):

Step P1. A user U_i inserts a smart card into the smart card reader and gives (ID_i, PW_i). The smart card of U_i calculates $Q_i = B_i \oplus h(ID_i||h(r_i \oplus PW_i))$ and $R_i^* = h(Q_i)$ and then verifies whether R_i^* and the stored R_i in the smart card are equal. If the values are unequal, the smart card rejects the login request. Otherwise, the legitimacy of U_i is ensured, and U_i can read the information stored in the smart card.

Step P2. The user U_i selects a new password PW_i^{new}, and then U_i generates a random number r_i^{new}. Then, the smart card calculates $B_i^{new} = Q_i \oplus h(ID_i||h\ (r_i^{new} \oplus PW_i^{new}))$, $PTC_i^{new} = PTC_i \oplus h(r_i \oplus PW_i) \oplus h(r_i^{new} \oplus PW_i^{new})$.

Step P3. The parameters $\{PTC_i, B_i, r_i\}$ in the smart card are replaced with new parameters $\{PTC_i^{new}, B_i^{new}, r_i^{new}\}$. Finally, the smart card contains $\{ID_{GWN}, PTC_i^{new}, TE_i, B_i^{new}, R_i, r_i^{new}, h(.)\}$.

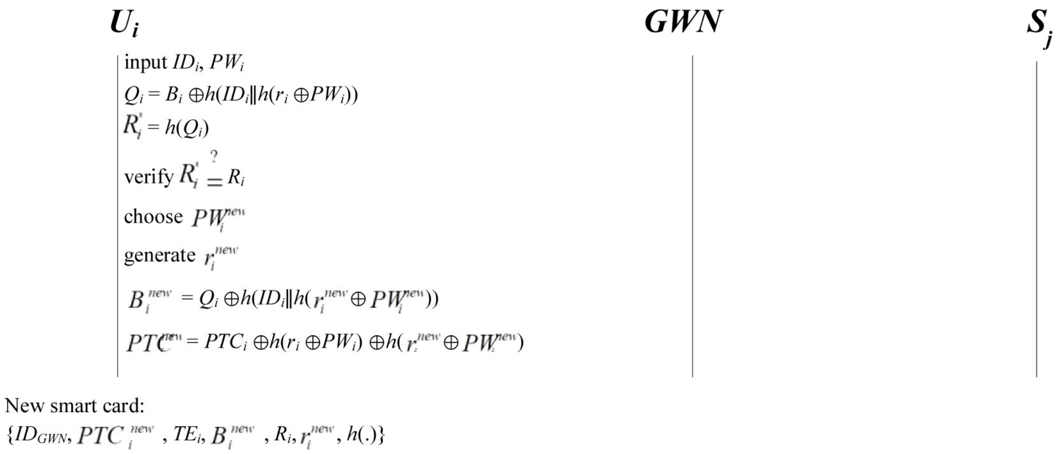
U_i
GWN
S_j
input ID_i, PW_i
$Q_i = B_i \oplus h(ID_i \| h(r_i \oplus PW_i))$
$R_i^* = h(Q_i)$
verify $R_i^* \overset{?}{=} R_i$
choose PW_i^{new}
generate r_i^{new}
$B_i^{new} = Q_i \oplus h(ID_i \| h(r_i^{new} \oplus PW_i^{new}))$
$PTC_i^{new} = PTC_i \oplus h(r_i \oplus PW_i) \oplus h(r_i^{new} \oplus PW_i^{new})$
New smart card:
$\{ID_{GWN}, PTC_i^{new}, TE_i, B_i^{new}, R_i, r_i^{new}, h(.)\}$

Fig 5. Password change phase in the proposed scheme (U_i can update the password without requiring the communication with the *GWN* and S_j).

https://doi.org/10.1371/journal.pone.0232277.g005

4. Security analysis

This section presents the security analysis of the proposed scheme and proves its security strength. Our scheme can overcome the weaknesses of previous schemes. Our proposed scheme has the following main security features.

4.1. Mutual authentication and session key agreement

Mutual authentication is a critical feature for verifying mutual validity among the *GWN*, U_i, and S_j in WSNs. Because encryption and a message authentication code (MAC) are required to protect data transmission between U_i and S_j, a session key must be negotiated in advance between these two participants [7]. In this section, we first illustrate the mutual authentication analysis of the proposed scheme, then we present the formal proofs. In the authentication and key agreement phase of the proposed scheme, mutual authentication between the *GWN* and S_j is accomplished by calculating verification information q_2 and q_3. In Step V3, S_j can verify the legitimacy of the *GWN* after determining whether q_2 and q_2^* are equal, where $q_2 = h(ID_i||TC_j||N_{GWN})$. Temporal credential TC_j is included in verification information q_2. This shows that the sensor node S_j can authenticate the validity of the *GWN*. In Step V4, the *GWN* can verify the legitimacy of S_j after confirming whether q_3 and q_3^* are equal, where $q_3 = h(ID_i||SID_j||K_i||N_i||N_{GWN})$. A secret sharing key K_i is included in verification information q_3. This shows that the *GWN* can authenticate S_j. By contrast, mutual authentication between U_i and the *GWN* is accomplished by calculating verification information q_1 and q_3. In Step V2, the *GWN* can verify the legitimacy of U_i after determining whether q_1^* and q_1 are equal, where $q_1 = h(ID_i||TC_i||N_i)$. Temporal credential TC_i is included in verification information q_1. This shows that the *GWN* can authenticate the user U_i. In Step V4, U_i can verify the legitimacy of S_j after confirming whether q_3 and q_3^* are equal, where $q_3 = h(ID_i||SID_j||K_i||N_i||N_{GWN})$. A secret sharing key K_i is included in verification information q_3. This shows that the user U_i can authenticate the sensor node S_j. In addition, because S_j has authenticated the validity of the *GWN*, the user U_i further authenticates the validity of the *GWN* as well. Therefore, on the basis of temporal credential signing and the secret sharing key, U_i, S_j, and the *GWN* can mutually authenticate each other in the proposed protocol. In Step V4, after completing the mutual authentication phase, U_i, the *GWN*, and S_j can separately generate the shared session key KEY_{ij} by computing $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$, where secret sharing key K_i and K_j are

Table 2. Notations of BAN logic.

Notation	Definition
$P \triangleleft X$	P ***sees*** X : P can receive and read X (possibly after doing some decryption).
$P \mid\sim X$	P ***said*** X : P once said X. P once sent a message including the statement X.
$P \mid\Rightarrow X$	P ***controls*** X : P has jurisdiction over X.
$P \mid\equiv X$	P ***believes*** X : P is entitled to believe X.
$\#(X)$	***fresh***(X) : X is regarded as a fresh statement.
$\langle X\rangle_Y$	X is combined with Y; Y is a secret.
(X,Y)	X and Y are said simultaneously.
$P \leftrightarrow^K Q$	P and Q share a common key K.
$P \leftarrow Y_{\rightarrow} \leftarrow Q$	Statement Y is identified only to P and Q.

https://doi.org/10.1371/journal.pone.0232277.t002

selected randomly. This shows that U_i, S_j, and the GWN can share a common session key after finishing the mutual authentication phase. The common session key is validated by U_i, the GWN, and S_j. This illustration indicates that our scheme provides session key agreement and mutual authentication. The formal proofs are given in the following lemmas and Proposition 1. We use the BAN logic method [3, 21–24] to formally validate the mutual authentication and session key agreement of our scheme. The BAN logic method is widely used to validate authentication and key establishment protocols [3, 21–24]. The BAN logic method accomplishes to introduce the logic of authentication and explain the protocols step-by-step. The notations of BAN logic are presented in Table 2. In Table 2, the symbols X and Y range over statements; Q and P are principals [20–22, 42].

The essential logical postulates for the BAN logic are listed as follows [20–22, 42]:

1. *Freshness-propagation* rule: $\frac{P\mid\equiv(X)}{P\mid\equiv(X,Y)}$. That is, if P is entitled to believe that one part of a formula (X,Y) is fresh, then he also is entitled to believe that the entire formula (X,Y) must also be fresh.

2. *Receiving* rule: $\frac{P\triangleleft(X,Y)}{P\triangleleft X}$ and $\frac{P\triangleleft\langle X\rangle_Y}{P\triangleleft X}$. That is, if a principal P can receive and read a formula (X,Y) or formula $\langle X\rangle_Y$, then he also can receive and read its components X.

3. *Nonce-verification* rule: $\frac{P\mid\equiv(X),\ P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$. That is, if P is entitled to believes that X is a fresh statement and that Q once said X, then P believes that Q believes X.

4. *Jurisdiction* rule: $\frac{P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$. That is, if P believes that Q has jurisdiction over X and P believes that Q believes X, then P believes X.

5. *Message-meaning* rule: $P\mid\equiv Q\frac{\longleftrightarrow Y\ P, P\triangleleft\langle X\rangle_Y}{P\mid\equiv Q\mid\sim X}$. That is, if P is entitled to believe that the key Y is shared with Q, and P sees X encrypted under Y, then P is entitled to believe that Q once said X.

6. *Session-key* rule: $\frac{P\mid\equiv(K),\ P\mid\equiv Q\mid\equiv X}{P\mid\equiv P\leftrightarrow^K Q}$, where statement X is an element of the combination session key K [21, 44]. That is, if P is entitled to believe that K is a fresh statement and that Q believes X, then P believes that P and Q share a common key K.

To validate the proposed protocol, we first summarize our scheme in the generic form [20, 21, 42]:

Message m_1. $U_i \rightarrow GWN$: $\{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$
$= \{ID_i \oplus h(TC_i \| ID_{GWN} \| N_i), h(ID_i \| TC_i \| N_i), K_i \oplus h(TC_i \| N_i), TE_i,$

$h(ID_i||ID_{GWN}||TE_i)$, $N_i\}$.
Message m_2. $GWN\rightarrow S_j$: $\{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$
$= \{ID_i \oplus h(TC_i||ID_{GWN}||N_i), ID_i \oplus h(TC_j||DID_i||N_{GWN}),$
$h(ID_i||TC_j||N_{GWN}), K_i\oplus h(TC_j||N_{GWN}), ID_{GWN}, N_i, N_{GWN}\}$.
Message m_3. $S_j \rightarrow GWN$: $\{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$
$= \{SID_j, h(ID_i||SID_j||K_i||N_i||N_{GWN}), K_j\oplus h(K_i||N_i||N_{GWN}),N_i, N_{GWN}\}$.
Message m_3. $S_j \rightarrow U_i$: $\{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$
$= \{SID_j, h(ID_i||SID_j||K_i||N_i||N_{GWN}), K_j\oplus h(K_i||N_i||N_{GWN}), N_i, N_{GWN}\}$.
Subsequently, we transform the generic form into the idealized form:
I$_1$. $U_i \rightarrow GWN$: $\langle N_i\rangle_{TC_i}, \langle N_i\rangle_{TC_i}, \langle\langle N_i\rangle_{TC_i}\rangle_{K_i}$
I$_2$. $GWN\rightarrow S_j$: $\langle N_i\rangle_{TC_i}, \langle N_{GWN}\rangle_{TC_j}, \langle N_{GWN}\rangle_{TC_j}, \langle\langle N_{GWN}\rangle_{TC_j}\rangle_{K_i}$
I$_3$. $S_j \rightarrow GWN$: $\langle N_i, N_{GWN}\rangle_{K_i}, \langle\langle N_i, N_{GWN}\rangle_{K_i}\rangle_{K_j}$
I$_4$. $S_j \rightarrow U_i$:$\langle N_i, N_{GWN}\rangle_{K_i}, \langle\langle N_i, N_{GWN}\rangle_{K_i}\rangle_{K_j}$
To analyze our scheme, we use the following assumptions:
A$_1$. $GWN| \equiv U_i \leftarrow \rightarrow TC_i\, GWN$ A$_2$. $GWN| \equiv S_j \leftarrow \rightarrow K_i\, GWN$ A$_3$.
$S_j| \equiv GWN \leftarrow \rightarrow TC_j\, S_j$ A$_4$. $U_i| \equiv S_j \leftarrow \rightarrow K_i\, U_i$ A$_5$. $GWN|\equiv\#(N_i)$ A$_6$.
$S_j|\equiv\#(N_{GWN})$
A$_7$. $U_i|\equiv\#(N_i,N_{GWN})$ A$_8$. $GWN|\equiv\#(N_i,N_{GWN})$
A$_9$. $GWN|\equiv U_i|\Rightarrow N_i$ A$_{10}$. $S_j|\equiv GWN|\Rightarrow N_{GWN}$
A$_{11}$.$GWN|\equiv S_j|\Rightarrow(N_i,N_{GWN})$ A$_{12}$. $U_i|\equiv S_j|\Rightarrow(N_i,N_{GWN})$

Lemma 1. *The GWN in our scheme can authenticate U_i; S_j can authenticate the GWN.*

Proof: In our scheme, U_i produces a nonce N_i. Then, U_i transmits N_i to the *GWN*. After obtaining N_i, the *GWN* generates a nonce N_{GWN} and then sends nonces (N_i, N_{GWN}) to S_j.

To prove that the *GWN* can authenticate U_i, the following belief must be demonstrated:
B$_1$. $GWN|\equiv N_i$
To prove that S_j can authenticate the *GWN*, the following belief must be demonstrated:
B$_2$. $S_j|\equiv N_{GWN}$
The steps for proving B$_1$:

S$_1$. *GWN* ***sees*** $\langle N_i\rangle_{TC_i}$ (Apply the *Receiving* rule and I$_1$)

S$_2$. *GWN* **believes** U_i ***said*** N_i. (Apply the *Message-meaning* rule, A$_1$, and S$_1$)

S$_3$. *GWN* **believes** U_i ***believes*** N_i. (Apply the *Nonce-verification* rule, A$_5$, and S$_2$)

S$_4$. *GWN* **believes** N_i. That is, $GWN|\equiv N_i$ (Apply the *Jurisdiction* rule, A$_9$, and S$_3$)

Consequently, the *GWN* authenticates U_i.
Similarly, the steps of the proof for B$_2$:

S$_5$. S_j ***sees*** $\langle N_{GWN}\rangle_{TC_j}$ (Apply I$_2$ and *Receiving* rule)

S$_6$. S_j ***believes*** *GWN* ***said*** N_{GWN}. (Apply the *Message-meaning* rule, A$_3$, and S$_5$)

S$_7$. S_j ***believes*** *GWN* ***believes*** N_{GWN}. (Apply the *Nonce-verification* rule, A$_6$, and S$_6$)

S$_8$. S_j ***believes*** N_{GWN}. That is, $S_j|\equiv N_{GWN}$ (Apply the *Jurisdiction* rule, A$_{10}$, and S$_7$).

Lemma 2. *The GWN in our scheme can authenticate S_j ; U_i can also authenticate S_j.*

Proof: In our scheme, after receiving nonces (N_i, N_{GWN}), the S_j returns (N_i, N_{GWN}) to the *GWN* and U_i.

To prove that the *GWN* can authenticate S_j, the following belief must be demonstrated:

B$_3$. $GWN|\equiv(N_i,N_{GWN})$

To prove that the U_i can authenticate S_j, the following belief must be demonstrated:

B$_4$. $U_i|\equiv(N_i,N_{GWN})$

The steps of the proof for B$_3$:

S$_9$. GWN ***sees*** $\langle N_i, N_{GWN}\rangle_{K_i}$ (Apply the *Receiving* rule and I$_3$)

S$_{10}$. GWN ***believes*** S_j ***said*** (N_i,N_{GWN}). (Apply the *Message-meaning* rule, A$_2$, and S$_9$)

S$_{11}$. GWN ***believes*** S_j ***believes*** (N_i,N_{GWN}).(Apply the *Nonce-verification* rule, A$_8$, and S$_{10}$)

S$_{12}$. GWN ***believes*** (N_i,N_{GWN}). That is, $GWN|\equiv(N_i,N_{GWN})$ (Apply the *Jurisdiction* rule, A$_{11}$, and S$_{11}$).

Consequently, the GWN can authenticate S_j.

Similarly, the steps of the proof for B$_4$:

S$_{13}$. U_i ***sees*** $\langle N_i, N_{GWN}\rangle_{K_i}$ (Apply the *Receiving* rule and I$_4$)

S$_{14}$. U_i ***believes*** S_j ***said*** (N_i,N_{GWN}). (Apply the *Message-meaning* rule, A$_4$, and S$_{13}$)

S$_{15}$. U_i ***believes*** S_j ***believes*** (N_i,N_{GWN}). (Apply the *Nonce-verification* rule, A$_7$, and S$_{14}$)

S$_{16}$. U_i ***believes*** (N_i,N_{GWN}). That is, $U_i|\equiv(N_i,N_{GWN})$. (Apply the *Jurisdiction* rule, A$_{12}$, and S$_{15}$)

Lemma 3. *In our scheme, the GWN, U_i, and S_j can coordinate the common session key KEY_{ij}.*

Proof: To prove that U_i, the GWN, and S_j in our scheme can share a session key $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$, the following beliefs must be demonstrated:

B$_5$. $U_i| \equiv U_i \overset{KEY_{ij}}{\longleftrightarrow} GWN$

B$_6$. $GWN| \equiv GWN \overset{KEY_{ij}}{\longleftrightarrow} U_i$

B$_7$. $S_j| \equiv S_j \overset{KEY_{ij}}{\longleftrightarrow} GWN$

B$_8$. $GWN| \equiv GWN \overset{KEY_{ij}}{\longleftrightarrow} S_j$ The steps for proving B$_5$ are:

S$_{17}$. U_i ***believes*** S_j ***believes*** (N_i,N_{GWN}). (Apply S$_{15}$)

S$_{18}$. S_j ***believes*** GWN ***believes*** N_{GWN}. (Apply S$_7$)

S$_{19}$. U_i ***believes*** GWN ***believes*** N_{GWN}. (Apply the Lemma 1, the Lemma 2, S$_{17}$, and S$_{18}$)

S$_{20}$. U_i ***believes fresh*** (N_i, N_{GWN}). (Apply A$_7$)

S$_{21}$. U_i ***believes fresh*** (KEY_{ij}). (Apply S$_{20}$ and *Freshness-propagation* rule)

S$_{22}$. *Ui* ***believes*** $U_i \overset{KEY_{ij}}{\longleftrightarrow} GWN$. That is, $U_i| \equiv U_i \overset{KEY_{ij}}{\longleftrightarrow} GWN$. (Apply the *Session-key* rule, S$_{19}$, and S$_{21}$)

Consequently, U_i believes that U_i shares the session key KEY_{ij} with the GWN.

Similarly, the steps of the proof for B$_6$:

S$_{23}$. GWN ***believes*** U_i ***believes*** N_i. (Apply S$_3$)

S$_{24}$. GWN ***believes fresh*** (N_i). (Apply A$_5$)

S$_{25}$. GWN ***believes fresh*** (KEY_{ij}). (Apply S$_{24}$ and *Freshness-propagation* rule)

S$_{26}$. GWN ***believes*** $GWN \overset{KEY_{ij}}{\longleftrightarrow} U_i$. That is, $GWN| \equiv GWN \overset{KEY_{ij}}{\longleftrightarrow} U_i$. (Apply S$_{23}$, S$_{25}$, and *Session-key* rule)

Consequently, the GWN believes that GWN shares the session key KEY_{ij} with U_i.

The steps of the proof for B$_7$ are:

S_{27}. S_j ***believes*** GWN ***believes*** N_{GWN}. (Apply S_7)

S_{28}. S_j ***believes fresh*** (N_{GWN}). (Apply A_6)

S_{29}. S_j ***believes fresh*** (KEY_{ij}). (Apply the *Freshness-propagation* rule and S_{28})

S_{30}. S_j ***believes*** $S_j \overset{KEY_{ij}}{\longleftrightarrow} GWN$. That is, $S_j| \equiv S_j \overset{KEY_{ij}}{\longleftrightarrow} GWN$. (Apply the *Session-key* rule, S_{27}, S_{29})

Consequently, S_j believes that S_j shares the session key KEY_{ij} with the GWN.
Similarly, the steps of the proof for B_8 are:

S_{31}. GWN ***believes*** S_j ***believes*** (N_i, N_{GWN}). (Apply S_{11})

S_{32}. GWN ***believes fresh*** (N_i). (Apply A_5)

S_{33}. GWN ***believes fresh*** (KEY_{ij}). (Apply S_{32} and *Freshness-propagation* rule)

S_{34}. GWN ***believes*** $GWN \overset{KEY_{ij}}{\longleftrightarrow} S_j$. That is, $GWN| \equiv GWN \overset{KEY_{ij}}{\longleftrightarrow} S_j$. (Apply S_{31}, S_{33}, and *Session-key* rule)

Consequently, the GWN believes that GWN shares the session key KEY_{ij} with S_j.

Proposition 1. *U_i, the GWN, and S_j in our scheme can mutually authenticate each other; they can share a common session key.*

Proof: From Lemma 2, U_i in our scheme can authenticate S_j. In addition, S_j can authenticate the GWN (Lemma 1). Thus, U_i can further authenticate the GWN as well. Conversely, the GWN can authenticate U_i (Lemma 1). Consequently, the GWN and U_i in our scheme can mutually authenticate each other. The GWN can authenticate S_j (Lemma 2). Conversely, S_j can authenticate the GWN (Lemma 1). Consequently, the GWN and S_j in our scheme can mutually authenticate each other. Mutual authentication can be provided in our scheme. After finishing the mutual authentication, U_i, the GWN, *and* S_j can share a session key $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$ (Lemma 3). Session key agreement can also be provided in our scheme.

4.2. Password protection, guessing attack resistance, and stolen smart card attack resistance

When a user's smart card is stolen or lost in a stolen smart card attack, an adversary can acquire information from the smart card. Then, the adversary masquerades as an authorized user to access to the GWN. However, password protection functionality can prevent the leakage of password information, such that the adversary cannot obtain useful information to perform an off-line password guessing attack.

Proposition 2. *The proposed scheme can provide password protection, guessing attack resistance, and stolen smart card attack resistance.*

Proof: In our scheme, the password presents with the $h(r_i \oplus PW_i)$ form, in which PW_i and r_i are hidden. $h(r_i \oplus PW_i)$ is not stored in the smart card, the GWN, or any other device. Thus, the adversary cannot directly obtain PW_i by performing an off-line password guessing attack on $h(r_i \oplus PW_i)$ [45]. Therefore, the proposed scheme can provide password protection and guessing attack resistance. Moreover, smart card secrets can be breached by monitoring power consumption or by analyzing leaked information [25, 42, 46]. When the adversary has a smart card that has been lost by its legitimate owner, the adversary can acquire the secret parameters from that smart card by applying the previously discussed method. We can prove that the proposed scheme can also provide stolen smart card attack resistance. That is, in the proposed scheme, the adversary cannot masquerade as a legitimate user to log in to the GWN when the adversary has obtained a legitimate user's smart card. Suppose that when the smart card of user U_i is stolen or lost, the adversary obtains that the smart card. The adversary can obtain the secret parameters $\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, r_i, h(.)\}$ from the smart card. To impersonate a legitimate user, the adversary must produce a new N_i'', randomly choose an imitative secret sharing key K_i'', and create an imitative login request message $\{DID_i'', q_1'', PKS_i'', TE_i, P_i, N_i''\}$ for the GWN. The imitative parameters $\{DID_i'', q_1'', PKS_i'', P_i\}$ are obtained using the following equations:

$DID_i'' = \mathrm{ID_i} \oplus \mathrm{h(TC_i \| ID_{GWN} \|} N_i'')$,

$q_1'' = \mathrm{h(ID_i \| TC_i \|} N_i'')$,

$PKS_i'' = K_i'' \oplus \mathrm{h(TC_i \|} N_i'')$,

$\mathrm{P_i = h(ID_i \| ID_{GWN} \| TE_i)}$.

Therefore, to obtain the imitative parameters $\{DID_i'', q_1'', PKS_i'', P_i\}$, the adversary must first obtain TC_i and ID_i by using the following equations:

$\mathrm{TC_i = h(P_i \| K_{GWN\text{-}U} \| TE_i)}$,

$TC_i = PTC_i \oplus h(r_i \oplus PW_i)$,

$ID_i = DID_i \oplus h(TC_i \| ID_{GWN} \| N_i)$.

Nevertheless, the adversary cannot acquire TC_i and ID_i because he/she does not possess $K_{GWN\text{-}U}$ and PW_i. Only the GWN knows the private key $K_{GWN\text{-}U}$ in our scheme. As previously discussed, the proposed scheme can provide password protection, and that the adversary cannot acquire PW_i by executing an off-line password guessing attack. Therefore, the imitative parameter set $\{DID_i'', q_1'', PKS_i'', P_i\}$of a login request message is not acquired. The adversary cannot masquerade as an authorized user by only using a smart card.

4.3. Two-factor security

By involving a smart card and a password in the login phase, two-factor security in our scheme can be achieved [9, 37, 47, 48].

Proposition 3. Two-factor security can be provided in our scheme.

Proof: First, assume that the adversary only has the smart card of $\mathrm{U_i}$. Let us even assume that the adversary can intercept login request message $m_1 = \{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$. As mentioned in Proposition 2, the adversary can obtain the secret parameters $\{ID_{GWN}, PTC_i, TE_i, B_i, R_i, r_i, h(.)\}$ from the smart card. To impersonate a legitimate user, the adversary must produce a new N_i'', randomly choose a new sharing key K_i'', and create an imitative login request message $\{DID_i'', q_1'', PKS_i'', \mathrm{TE_i}, \mathrm{P_i}, N_i''\}$ for the GWN, where $DID_i'' = \mathrm{ID_i} \oplus \mathrm{h(TC_i \| ID_{GWN} \|} N_i'')$, $q_1'' = \mathrm{h(ID_i \| TC_i \|} N_i'')$, and $PKS_i'' = K_i'' \oplus \mathrm{h(TC_i \|} N_i'')$. Consequently, to gain the parameter set $\{DID_i'', q_1'', PKS_i''\}$, the adversary must acquire $\mathrm{TC_i}$ and $\mathrm{ID_i}$ by applying the following equations: $\mathrm{TC_i = h(P_i \| K_{GWN\text{-}U} \| TE_i)}$, $\mathrm{TC_i = PTC_i \oplus h(r_i \oplus PW_i)}$, and $\mathrm{ID_i = DID_i \oplus h(TC_i \| ID_{GWN} \| N_i)}$. Nevertheless, the adversary cannot acquire $\mathrm{TC_i}$ and $\mathrm{ID_i}$ because he/she does not possess $\mathrm{K_{GWN\text{-}U}}$ and $\mathrm{PW_i}$. Only the GWN knows the private key $\mathrm{K_{GWN\text{-}U}}$ in our scheme, and we have proven that the proposed scheme can provide password protection to prevent the leakage of $\mathrm{PW_i}$ information (Section 4.2). Therefore, the parameter set $\{DID_i'', q_1'', PKS_i''\}$ of the login request message is not acquired, and the adversary cannot disguise as an authorized user by only using the smart card. Secondly, assume that the adversary only has the password $\mathrm{PW_i}$ and identification $\mathrm{ID_i}$ of $\mathrm{U_i}$. Under this condition, the adversary also cannot acquire $\mathrm{TC_i}$ to calculate the parameters $\{DID_i'', q_1'', PKS_i''\}$ because he/she does not know $\mathrm{K_{GWN\text{-}U}}$ and $\mathrm{PTC_i}$ (which are not stored in the smart card). Therefore, the adversary cannot impersonate an authorized user when he/she either acquires information from the smart card or knows $\{\mathrm{ID_i}, \mathrm{PW_i}\}$. Our scheme can withstand this type of masquerade attack and provide two-factor security.

4.4. Masquerade attack resistance and replay attack resistance

Protection against masquerade attacks is a principal security feature for any remote user authentication scheme. Replay attack resistance means that the adversary cannot attempt to replay any previously intercepted message to spoof the *GWN*.

Proposition 4. *Our scheme can provide masquerade attack resistance and replay attack resistance.*

Proof: Proposition 3 has demonstrated that our scheme can protect against masquerade attacks caused by either the loss of a smart card or the revelation of sensitive identification and password details $\{ID_i, PW_i\}$. The reliability of our scheme against other masquerade attacks must be demonstrated. We can even assume that the adversary is a legitimate user L and undertakes to impersonate a user U_i. Adversary L may intercept the login request message $m_1 = \{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$. Adversary L can have $\{ID_l, PW_l\}$ and acquire $\{ID_{GWN}, PTC_l, TE_l, B_l, R_l, r_l, h(.)\}$ from his/her smart card because he/she is an admissible user. Adversary L generates a new nonce $N_i^{''}$, randomly chooses an imitative secret sharing key $K_i^{''}$, and creates an imitative login request message $\{DID_i^{''}, q_1^{''}, PKS_i^{''}, TE_i, P_i, N_i^{''}\}$ for the *GWN*, where $DID_i^{''} = ID_i \oplus h(TC_i||ID_{GWN}||N_i^{''})$, $q_1^{''} = h(ID_i||TC_i||N_i^{''})$, and $PKS_i^{''} = K_i^{''} \oplus h(TC_i||N_i^{''})$. Nevertheless, adversary L still cannot acquire TC_i and ID_i to calculate the parameters $\{DID_i^{''}, q_1^{''}, PKS_i^{''}\}$ because he/she does not possess $K_{GWN\text{-}U}$ and PW_i (Proposition 2). In addition, adversary L cannot compute the shared session key $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$ because he or she does not know K_i and K_j in KEY_{ij}. Thus, adversary L cannot impersonate any other legitimate user. Consequently, our scheme can protect against masquerade attacks when an adversary impersonates any other legitimate user. Adversary L can undertake to replay the intercepted message $\{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$ to the *GWN*. However, after receiving message $m_3 = \{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$, adversary L cannot compute the shared session key $KEY_{ij} = h(K_i||K_j||N_i||N_{GWN}||SID_j)$ because he or she cannot obtain K_i and K_j in KEY_{ij}. Consequently, resistance to replay attacks is guaranteed as well. Next, we prove that an adversary cannot masquerade as a sensor node to spoof the user. Suppose adversary L has intercepted message m_2 when the *GWN* attempts to send it to S_j; that is, the message $\{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$. To masquerade as a sensor node to spoof the user, the adversary must randomly choose an imitative secret sharing key $K_j^{''}$ and send an imitative response message $\{SID_j, q_3, PKS_j^{''}, N_i, N_{GWN}\}$ to the *GWN*, where $q_3 = h(ID_i||SID_j||K_i||N_i||N_{GWN})$ and $PKS_j^{''} = K_j^{''} \oplus h(K_i||N_i||N_{GWN})$. To obtain the parameters $\{q_3, PKS_j^{''}\}$, the adversary must first know K_i. Moreover, K_i can be obtained by using the equation $K_i = PKS_{GWN} \oplus h(TC_j||N_{GWN})$. Nevertheless, the adversary cannot acquire K_i because he/she does not possess the temporal credential TC_j. Therefore, the parameters $\{q_3, PKS_j^{''}\}$ cannot be acquired, and the adversary cannot send an imitative response message $\{SID_j, q_3, PKS_j^{''}, N_i, N_{GWN}\}$ to the *GWN*. Consequently, our scheme can protect against masquerade attacks when an adversary masquerades as a sensor node to spoof the user.

4.5. Stolen verifier attack resistance and insider attack resistance

The stolen verifier attack means that the adversary steals the verification table from the *GWN* or S_j. By contrast, an insider attack involves any privileged insider of the *GWN* purposely obtaining a user password, which leads to security defects in the remote user authentication scheme [41, 49].

Proposition 5. *Our scheme can protect against stolen verifier attacks and insider attacks.*

Proof: The *GWN* and S_j in our scheme do not retain any verification table for verifying the legitimacy of registered users or sensor nodes. Therefore, the adversary cannot find any verifiable information in the *GWN* or S_j to impersonate a legitimate user. Consequently, our scheme can protect against stolen verifier attacks [9, 40]. Moreover, because U_i presents $h(r_i \oplus PW_i)$ to register with the *GWN*. r_i and PW_i are hidden from the *GWN*. In addition, the *GWN* does not store any verifier $h(r_i \oplus PW_i)$. The privileged insider of the *GWN* cannot acquire PW_i by executing any off-line password guessing attack [45]. Consequently, our scheme can resist insider attacks [41].

4.6. Password updating, adding new user functionality, and time synchronization avoidance

In our scheme, users are not obliged to share their IDs and passwords with the *GWN* before the registration. During the registration process, a new user U_i can freely choose some identification string ID_i and password PW_i as favorite strings without requiring assistance from the *GWN*. Any new legitimate user can be freely added to the system after the registration. Therefore, the proposed scheme provides a convenient functionality for adding new users. Moreover, as mentioned, it is strongly recommended that for security policy, users update or change their passwords frequently to protect against compromise [32]. In the password change phase of our scheme, a legitimate user U_i can freely choose his/her new password to update or change the password without requiring extra communication message overhead to exchange messages with the *GWN* (Fig 5). Consequently, our scheme provides the functionalities of freely chosen passwords and efficient password updating. Finally, our scheme does not require any timestamp to verify mutual authentication among U_i, the *GWN*, and S_j because our scheme is a nonce-based scheme. Consequently, our scheme is not obliged to provide synchronized time clocks for all devices [3, 17, 18], and it can avoid the time-synchronization problem for WSNs in IoT environments [3, 17, 25].

4.7. User anonymity (identity protection)

The user anonymity (identity protection) means that the identity of any user is disclosed only to service providers [9].

Proposition 6. *Our scheme can provide user anonymity to protect user identity.*

Proof: The adversary can intercept message $m_1 = \{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$ to acquire the identification string of U_i. The parameters DID_i, q_1, PKS_i, and P_i are obtained using the following equations:

$DID_i = ID_i \oplus h(TC_i||ID_{GWN}||N_i)$,

$q_1 = h(ID_i||TC_i||N_i)$,

$PKS_i = K_i \oplus h(TC_i||N_i)$,

$P_i = h(ID_i||ID_{GWN}||TE_i)$.

However, in Proposition 2, we show that the adversary cannot obtain ID_i and TC_i because he or she does not know $K_{GWN\text{-}U}$ and PW_i. The identification string ID_i also cannot be derived from the equations above. Therefore, an adversary cannot acquire ID_i to identify the user U_i, and our scheme can provide user anonymity to protect user identity.

4.8. GWN bypassing attack resistance and GWN spoofing attack resistance

A *GWN* bypassing attack occurs when an adversary can bypass the *GWN* to forge a verification message straight to the sensor node S_j without passing the *GWN* login [7]. By contrast, a *GWN* spoofing attack occurs when an adversary may impersonate the *GWN* to obtain private login information of U_i.

Proposition 7. *Our scheme can protect against GWN bypassing attacks and GWN spoofing attacks.*

Proof: To bypass the *GWN*, an adversary must send an imitative verification message $m_2 = \{DID_i, DID_{GWN}, q_2, PKS_{GWN}, ID_{GWN}, N_i, N_{GWN}\}$ straight to S_j, where $q_2 = h(ID_i||TC_j||N_{GWN})$. However, the adversary cannot obtain q_2 to create an imitative message m_2 because he or she does not know the temporal credential TC_j; thus, the adversary cannot bypass the *GWN* to forge m_2 to S_j. Without m_2, S_j cannot respond with any other messages. Consequently, our scheme can prevent *GWN* bypassing attacks. By contrast, the adversary may attempt to impersonate the *GWN* to acquire the secret login information of U_i. To pose as the *GWN*, the adversary can intercept some login request message $m_1 = \{DID_i, q_1, PKS_i, TE_i, P_i, N_i\}$ and respond with an imitative message $m_3 = \{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$ to U_i, where $q_3 = h(ID_i||SID_j||K_i||N_i||N_{GWN})$. Verification information q_3 includes a secret sharing key K_i. However, as mentioned in Proposition 4, the adversary cannot acquire K_i because he/she does not know temporal credential TC_j. Therefore, the adversary cannot obtain q_3; thus, the adversary cannot send an imitative message $m_3 = \{SID_j, q_3, PKS_j, N_i, N_{GWN}\}$ to respond to U_i. The adversary cannot convince U_i that he/she is a legitimate *GWN*. Consequently, our scheme can protect against *GWN* spoofing attacks.

5. Performance evaluation and functionality comparison

Performance and functionality evaluations are critical to establish validity for practical deployment. In this section, the performance and functionality of our scheme are evaluated. The performance efficiency and functional effectiveness of our authentication scheme are demonstrated.

5.1. Functionality comparison

Table 3 presents a functionality comparison of our scheme versus previous related schemes. In Table 3, *Yes* denotes the scheme has a security feature; *No* denotes the contrary. The weaknesses of the previous related schemes for WSNs are mentioned in Section 2 and summarized in Table 3. We present a practical scenario to show that the proposed scheme can provide secure functionality and effectiveness for WSNs in IoT environments. Suppose that an adversary, *Eve*, undertakes to damage our scheme by executing the following attacks: guessing attack, stolen smart card attack, masquerade attack, replay attack, stolen verifier attack, insider

Table 3. Functionality comparison of our scheme with other related schemes.

	Ours Ostad-Sharif Amin et al. Chang et al. Xue et al. Yeh et al. Khan et al. Chen et al. Das **(2019)[2] (2018)[34] (2016)[35] (2013)[7] (2011)[8] (2010)[32] (2010)[33] (2009)[5]**
Password protection	Yes Yes Yes No No Yes Yes No No
Stolen smart card attack resistance	Yes Yes Yes No No No No No No
Masquerade attack resistance	Yes Yes Yes No Yes Yes Yes No No
Replay attacks resistance	Yes Yes Yes No Yes No Yes Yes Yes
Insider attack resistance	Yes Yes Yes Yes No Yes Yes No No
Password updating/changing	Yes No Yes Yes No No Yes No No
Time synchronization avoidance	Yes No No No No Yes No No No
Mutual authentication	Yes No Yes Yes Yes Yes Yes Yes No
Session key agreement	Yes Yes Yes Yes Yes Yes No No No
User anonymity	Yes Yes Yes No Yes No Yes Yes Yes
GWN bypassing attack resistance	Yes Yes Yes Yes Yes Yes No No No

https://doi.org/10.1371/journal.pone.0232277.t003 attack, user anonymity attack, or GWN bypassing attack. Section 4 has shown that our scheme has the following abilities. *Eve* cannot directly obtain a user's password by executing a password guessing attack. When *Eve* steals a user's smart card, she cannot impersonate an authorized user to access the system. When *Eve* is even a legitimate user who pretends to be a different legitimate user, our scheme can protect against this masquerade attack. Moreover, *Eve* may undertake to replay some intercepted message to the *GWN*. Our scheme can provide resistance to replay attacks. *Eve* cannot breach the system by stealing the verification table. Even as an insider, *Eve* cannot acquire a password by executing any password guessing attack. *Eve* may intercept a login request message from the user to acquire the identification information, but the identification information of a user cannot be derived. Finally, *Eve* cannot forge an imitative message and send it straight to the sensor node to bypass the GWN. Moreover, our scheme has other security functionalities, which include updating passwords, choosing passwords freely, adding new users, and time synchronization avoidance. Our scheme provides a secure common session key and mutual authentication. Our scheme can thus protect against all listed attacks from *Eve*.

5.2. Performance evaluation

The proposed scheme comprises four phases: registration phase, login phase, authentication and key agreement phase, and password change phase. In a WSN environment, the performance of the authentication scheme is affected mainly by the authentication and key agreement phase [2, 7, 34, 35]. This phase is the main part of the authentication scheme and is what chiefly distinguishes it from the various authentication schemes in WSNs [2, 7, 34, 35]. Therefore, we focus our discussion on the performance comparison of the authentication and key agreement phase in the authentication schemes. The performance comparison is usually separated into communication costs and computational costs [2, 7, 34, 35, 42]. The computational costs are defined as the time spent by the user and service provider in the process [2, 7, 34, 35, 42]. By contrast, the communication costs are defined as the number of messages dispatched by the user and service provider in the process [9, 42]. The performance comparison of our scheme and previous related schemes is shown in Table 4. Table 4 presents the computational

Table 4. Performance comparison of our scheme with other related schemes.

	Ours Ostad-Sharif Amin et al. Chang et al. Xue et al. Yeh et al. Khan et al. Chen et al. Das **(2019)[2] (2018)[34] (2016)[35] (2013)[7] (2011)[8] (2010)[32] (2010)[33] (2009)[5]**
【*Computational cost*】	
authentication phase	
User	$4T_h$ $10T_h$ $13T_h$ $3T_h$ $5T_h$ $2T_{ecc}+1T_h$ $3T_h$ $4T_h$ $3T_h$
GWN	$8T_h$ $14T_h$ $14T_h$ $5T_h$ $11T_h$ $4T_{ecc}+3T_h$ $5T_h$ $5T_h$ $4T_h$
Sensor node	$3T_h$ $3T_h$ $2T_h$ $1T_h$ $3T_h$ $2T_{ecc}+2T_h$ $2T_h$ $2T_h$ $1T_h$
key agreement phase	
User	$3T_h$ $2T_h$ $1T_h$ $3T_h$ $3T_h$ $1T_h$ $-^*$ $-^*$ $-^*$
GWN	$3T_h$ $3T_h$ $3T_h$ $3T_h$ $3T_h$ $1T_h$ $-^*$ $-^*$ $-^*$
Sensor node	$3T_h$ $2T_h$ $2T_h$ $4T_h$ $3T_h$ $1T_h$ $-^*$ $-^*$ $-^*$
Total	$24T_h$ $34T_h$ $35T_h$ $19T_h$ $28T_h$ $8T_{ecc}+9T_h$
【*Communication cost*】	
Transmitted message	4 6 6 4 4 3 4 4 3

* Khan et al. scheme, Chen et al. scheme and Das scheme do not provide the key agreement phase for session key agreement.

https://doi.org/10.1371/journal.pone.0232277.t004 costs and communication costs of the authentication and key agreement phase in each authentication scheme run without the consideration of interference and packet loss [2, 7, 21, 34, 35]. The notation T_h is defined as the time complexity of the hash function; T_{ecc} is the time complexity of the encryption/decryption operation in elliptic curve cryptography (ECC) algorithm [7]. The computational costs of the exclusive-or operation are usually neglected because it necessitates minimal computations [2, 7, 34, 35]. We first analyze the computational costs of the authentication and key agreement phase for each scheme as follows:

1. In the authentication phase of the Ostad-Sharif et al. scheme [2], the user requires $10T_h$ to compute the parameters of the login request message and the response message. The GWN must spend $14T_h$ to compute the parameters in a response message for the user and a request message for the sensor node. The sensor node must expend $3T_h$ to confirm whether the verification equations hold. In addition, the user, GWN, and sensor node must expend $2T_h$, $3T_h$, and $2T_h$ separately to negotiate the shared session key in the key agreement phase. Accordingly, the total computational costs for the user, GWN, and sensor node are $12T_h$, $17T_h$, and $5T_h$, respectively [2].

2. In the authentication phase of the Amin et al. scheme [34], the user requires $13T_h$ to compute the parameters of the login request message and the response message. The GWN must spend $14T_h$ to compute the parameters in a request message for the sensor node and a response message for the user. The sensor node must expend $2T_h$ to confirm whether the verification equations hold. In addition, the user, GWN, and sensor node must expend $1T_h$, $3T_h$, and $2T_h$ separately to negotiate the shared session key in the key agreement phase. Accordingly, the total computational costs for the user, GWN, and sensor node are $14T_h$, $17T_h$, and $4T_h$, respectively [34].

3. In the authentication phase of the Chang et al. scheme [35], the user requires $3T_h$ to compute the parameters of the login request message. The sensor node must expend $1T_h$ to compute the parameters in a message for the GWN. The GWN must spend $5T_h$ to verify the login request. In addition, the user, GWN, and sensor node must expend $3T_h$, $3T_h$, and $4T_h$ separately to negotiate the shared session key in the key agreement phase. Accordingly, the total computational costs for the user, GWN, and sensor node are $6T_h$, $8T_h$, and $5T_h$, respectively [35].

4. In the authentication phase of the Xue et al. scheme [7], the user requires $5T_h$ to compute the parameters of the login request message. The GWN must spend $11T_h$ to verify the login request message and compute the parameters of the request message for the sensor node. The sensor node must expend $3T_h$ to confirm whether the verification equations hold. Moreover, the user, GWN, and sensor node must expend $3T_h$, $3T_h$, and $3T_h$ separately to negotiate the shared session key in the key agreement phase. Accordingly, the total computational costs for the user, GWN, and sensor node are $8T_h$, $14T_h$, and $6T_h$, respectively [7].

5. In the Khan et al. scheme [32], the user must expend $3T_h$ to generate a login request message. The GWN must expend $5T_h$ to confirm whether the verification equations hold and to calculate the parameters of the request message for the sensor node. The sensor node requires $2T_h$ to confirm whether the verification equations hold and to generate a response message for the GWN. However, the Khan et al. scheme does not provide the key agreement phase for the session key agreement.

6. In the Chen et al. scheme [33], the user must expend $4T_h$ to produce a login request message and to validate a response message. The GWN requires $5T_h$ to validate a login request message and to respond to a user's request. The sensor node must expend $2T_h$ to verify the request message from the GWN and to generate a response message for the user. However, the Chen et al. scheme also does not provide any key agreement phase.

7. In the Das scheme [5], the user requires $3T_{\mathrm{h}}$ to generate the login request message. The GWN must expend $4T_{\mathrm{h}}$ to confirm whether the verification equations hold and to calculate the parameters of the request message for the sensor node. The sensor node requires $1T_{\mathrm{h}}$ to confirm whether the verification equations hold and to generate a response message for the user. The Das scheme [5] does not provide the key agreement phase as well.

8. The Yeh et al. scheme [8] uses elliptic curve cryptography (ECC) to provide both the authentication phase and session key agreement phase. That scheme requires that the user, GWN, and sensor node expend $2T_{\mathrm{ecc}} + 1T_{\mathrm{h}}$, $4T_{\mathrm{ecc}} + 3T_{\mathrm{h}}$, and $2T_{\mathrm{ecc}} + 2T_{\mathrm{h}}$ separately to complete the authentication phase [7]. Moreover, the user, GWN, and sensor node must expend $1T_{\mathrm{h}}$, $1T_{\mathrm{h}}$, and $1T_{\mathrm{h}}$ separately to compute a shared session key in the key agreement phase [7]. Accordingly, the total computational costs of the user, GWN, and sensor node are $2T_{\mathrm{ecc}} + 2T_{\mathrm{h}}$, $4T_{\mathrm{ecc}} + 4T_{\mathrm{h}}$, and $2T_{\mathrm{ecc}} + 3T_{\mathrm{h}}$, respectively [7].

9. Our proposed scheme provides both the authentication phase and key agreement phase. In the authentication phase of our scheme, the user requires only $4T_{\mathrm{h}}$ to calculate the parameters of a login request message. The GWN expends only $8T_{\mathrm{h}}$ to verify the login request and to calculate the parameters of the request message for the sensor node. The sensor node requires only $3T_{\mathrm{h}}$ to confirm whether the verification equations hold. In the key agreement phase, the user, GWN, and sensor node expend only $3T_{\mathrm{h}}$, $3T_{\mathrm{h}}$, and $3T_{\mathrm{h}}$, respectively, to negotiate the shared session key. Accordingly, the total computational costs for the user, GWN, and sensor node are $7T_{\mathrm{h}}$, $11T_{\mathrm{h}}$, and $6T_{\mathrm{h}}$, respectively.

Our proposed scheme uses only the hash function and XOR operations to design a simple authentication and key agreement scheme. However, the Yeh et al. scheme [8] provides a authentication and key agreement scheme which is established by an asymmetric encryption algorithm (specifically, an ECC). According to an experimental finding obtained in a related study, the one-way hash function is computationally efficient. The time complexity of the hash function is less than that of an asymmetric ECC encryption operation [2, 3, 7, 34, 35]. The following is a practical example for the computational costs: In an environment with a CPU of 3.2 GHz and with 3.0 GB of RAM, completing a one-way hash operation requires 0.02 ms on average when using SHA-1, and completing an asymmetric ECC encryption operation requires 0.45 ms on average when using ECC-160 [7].

For the user in each scheme run, the Yeh et al. scheme requires 0.94 ms for $2T_{\mathrm{ecc}} + 2T_{\mathrm{h}}$. The Amin et al. scheme requires 0.28 ms for $14T_{\mathrm{h}}$. The Ostad-Sharif et al. scheme requires 0.24 ms for $12T_{\mathrm{h}}$. By contrast, our scheme can perform the run in only 0.14 ms for $7T_{\mathrm{h}}$. Therefore, the computational load of the user in the proposed scheme is reduced to 14.89% compared with the Yeh et al. scheme and to 58.33% compared with the Ostad-Sharif et al. scheme.

For the GWN in each scheme run, the Yeh et al. scheme requires 1.88 ms for $4T_{\mathrm{ecc}} + 4T_{\mathrm{h}}$. The Amin et al. scheme requires 0.34 ms for $17\mathrm{T}_{\mathrm{h}}$. The Ostad-Sharif et al. scheme requires 0.34 ms for $17\mathrm{T}_{\mathrm{h}}$. By contrast, our scheme can perform the run in only 0.22 ms for $11T_{\mathrm{h}}$. Therefore, the computational load of the GWN in the proposed scheme is reduced to 11.7% compared with the Yeh et al. scheme and to 64.7% compared with the Ostad-Sharif et al. scheme.

For the sensor node in each scheme run, the Yeh et al. scheme requires 0.96 ms for $2T_{\mathrm{ecc}} + 3T_{\mathrm{h}}$. The Amin et al. scheme requires 0.08 ms for $4\mathrm{T}_{\mathrm{h}}$. The Ostad-Sharif et al. scheme requires 0.1 ms for $5\mathrm{T}_{\mathrm{h}}$. By contrast, our scheme can perform the run in 0.12 ms for $6T_{\mathrm{h}}$. Therefore, the computational load of the sensor node in the proposed scheme is reduced to 12.5% compared with the Yeh et al. scheme.

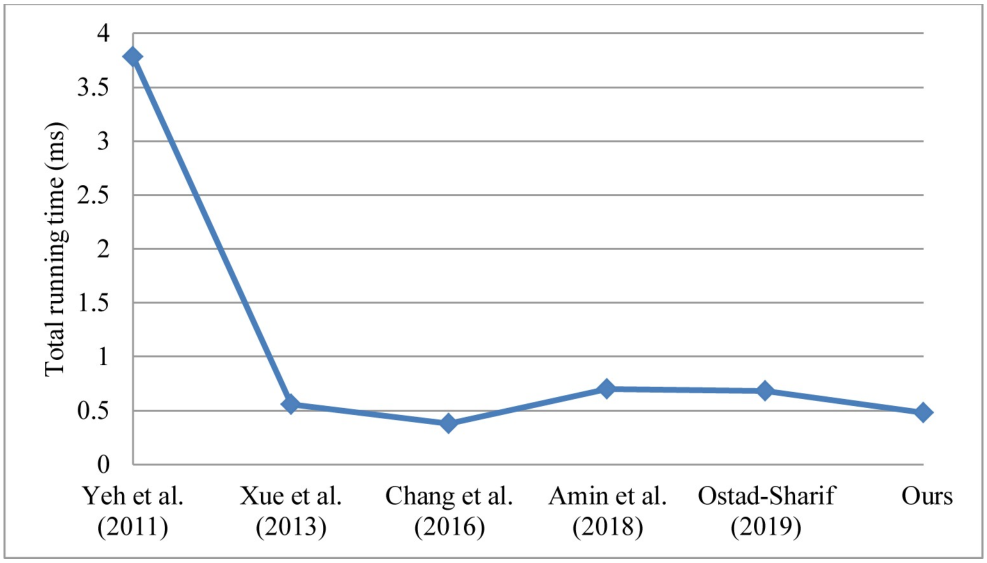

Fig 6. Comparison of running time.

https://doi.org/10.1371/journal.pone.0232277.g006

In Table 4, the total computational costs of the schemes of Yeh et al., Xue et al., Chang et al., Amin et al., Ostad-Sharif et al., and ours are $8T_{\mathrm{ecc}}+9T_{\mathrm{h}}$, $28T_{\mathrm{h}}$, $19T$h, $35T$h, $34T$h, and $24T$h, respectively. Therefore, the total running time of the schemes of Yeh et al., Xue et al., Chang et al., Amin et al., Ostad-Sharif et al., and ours are 3.78, 0.56, 0.38, 0.7, 0.68, and 0.48 ms, respectively (Fig 6). Therefore, the total running time of our scheme is 12.7%, 85.7%, 68.6%, and 70.6% of that of the schemes of Yeh et al., Xue et al., Amin et al., and Ostad-Sharif et al., respectively. Although the total running time of our scheme (0.48 ms) is slightly greater than that of the Chang et al. scheme (0.38 ms), our scheme can overcome the security weaknesses of previous related schemes and provide greater security functionality (Table 3).

The energy consumption of the Yeh et al. scheme [8] is ascribed chiefly to the asymmetric ECC cryptosystem and hash functions. By contrast, the energy consumption of our scheme is principally attributed to the hash functions. As mentioned, the energy consumption for executing the hash function is much lower than that for executing an asymmetric ECC cryptosystem [38, 39]. A practical example follows: While using SHA-1 to compute the hash value, a 1-byte data packet requires 0.76 μJ of energy [43, 38, 39]. Nevertheless, a 163-bit ECC asymmetric cryptosystem requires 134.2 mJ of energy [38, 39]. As previously discussed, the total computational costs of the schemes of Yeh et al., Xue et al., Chang et al., Amin et al., Ostad-Sharif et al., and ours are $8T_{\mathrm{ecc}}+9T_{\mathrm{h}}$, $28T_{\mathrm{h}}$, $19T_{\mathrm{h}}$, $35T_{\mathrm{h}}$, $34T_{\mathrm{h}}$, and $24T_{\mathrm{h}}$, respectively (Table 4). Consequently, the total energy consumption levels of the schemes of Yeh et al., Xue et al., Chang et al., Amin et al., Ostad-Sharif et al., and ours are 1073606.8, 21.3, 14.4, 26.6, 25.8, and 18.2 μJ, respectively. Consequently, in each scheme run, the total energy consumed by our scheme is 0.0017%, 85.4%, 68.4%, and 70.5% of that consumed by the schemes of Yeh et al., Xue et al., Amin et al., and Ostad-Sharif et al., respectively (Fig 7). Because the total energy consumption of the Yeh et al. scheme is excessive relative to other schemes, it cannot be shown in Fig 7. Although the total energy consumption of our scheme (18.2 μJ) is slightly greater than that of the Chang et al. scheme (14.4 μJ), our scheme provides superior security functionality to overcome the weaknesses of previous schemes (Table 3).

As mentioned, the communication cost accounts for the number of messages transmitted. A low number of transmitted messages results in less consumption for the message overhead

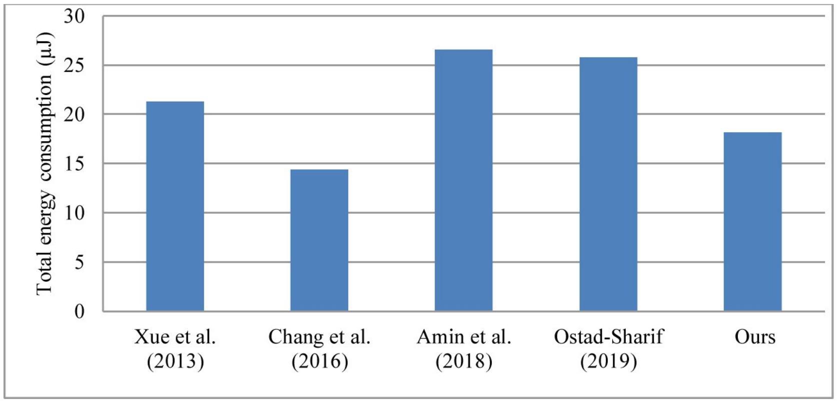

Fig 7. Comparison of energy consumption.

https://doi.org/10.1371/journal.pone.0232277.g007

[9, 42]. In completing the authentication and key agreement phase, the total numbers of transmitted messages of the schemes of Ostad-Sharif et al., Amin et al., Chang et al., Xue et al., Yeh et al., and ours are 6, 6, 4, 4, 3, and 4, respectively (Table 4). Although the communication costs of the proposed scheme (4 transmitted messages) is slightly greater than the Yeh et al. scheme (3 transmitted messages), the Yeh et al. scheme is subject to high computational costs (3.78 ms, Fig 6) and large energy consumption (1073606.8 μJ) due to its use of ECC.

In this subsection, we demonstrate that our scheme is highly efficient because of the superior performance: low computational cost (0.14 ms for the user, 0.12 ms for the sensor node, and 0.22 ms for the GWN), low energy consumption (18.2 μJ for the authentication and key agreement phase), and low communication cost (4 transmitted messages for the authentication and key agreement phase, 0 transmitted messages for the password change phase).

6. Conclusions

This paper analyzes the security weaknesses of related authentication schemes and proposes a more efficient and secure authentication scheme for WSNs in IoT environments. The BAN logic method is used to prove our scheme. Finally, we compare the functional effectiveness and performance efficiency of our scheme with those of previously published schemes. Cryptanalysis revealed that our scheme overcomes the security weaknesses of the previously published schemes. Our scheme satisfies the requirement of basic design criteria for the authentication scheme as well. Consequently, our scheme can enhance security effectiveness in real-world IoT environments and provide additional security functionalities compared with the other discussed schemes. Moreover, performance analysis revealed that our scheme demonstrates high efficiency and superior performance.

Our future work and challenges include attempting to find security risks in heterogeneous IoT environments. Various heterogeneous IoT applications can cause serious challenges in securing networks. Future studies will further evaluate the reliability and scalability of the proposed scheme in heterogeneous IoT environments. Moreover, we also study highly secure machine learning-based authentication schemes for WSNs in intelligent IoT environments. The integration of Big Data with intelligent IoT networks will be challenging due to the limited resources of WSNs.

Acknowledgments

The authors gratefully acknowledge the helpful comments and suggestions of the editor and the reviewers, which have improved the presentation

Author Contributions

Writing – original draft: Chi-Tung Chen, Cheng-Chi Lee, Iuon-Chang Lin.

Writing – review & editing: Chi-Tung Chen, Cheng-Chi Lee, Iuon-Chang Lin.

References

1. Atzori L, Iera A, Morabito G. The Internet of things: A survey. Computer Networks. 2010; 2787–2805.
2. Ostad-Sharif A, Arshad H, Nikooghadam M, Abbasinezhad-Mood D. Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme. Future Generation Computer Systems. 2019; 100: 882–892.
3. Li X, Niu J, Kumari S, Wu F, Sangaiah AK, Choo KK R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. Journal of Network and Computer Applications. 2018; 103: 194–204.
4. Callaway EH. Wireless Sensor Networks: Architectures and Protocols. Auerbach Publications; 2004.
5. Das ML. Two-factor user authentication in wireless sensor networks. IEEE transactions on wireless communications. 2009; 8: 1086–1090.
6. Akyildiz IF, Weilian S, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. IEEE Communication Magazine. 2002; 40: 102–114.
7. Xue k, Ma C, Hong P, Ding R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. Journal of Network and Computer Applications. 2013; 36: 316–323.
8. Yeh HL, Chen TH, Liu PC, Kim TH, Wei HW. A secured authentication protocol for wireless sensor networks using elliptic curves cryptograph. Sensors. 2011; 11: 4767–4779. https://doi.org/10.3390/s110504767 PMID: 22163874
9. Liao YP, Wang SS. A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009; 31: 24–29.
10. Das ML, Saxena A, Gulati V P. A dynamic ID-based remote user authentication scheme. IEEE Transactions on Consumer Electronics. 2004; 50: 629–631.
11. Mishra D, Vijayakumar P, Sureshkumar V, Amin R, Islam SK H. Efficient authentication protocol for secure multimedia communications in IoT-enabled wireless sensor networks. Multimedia Tools and Applications. 2018; 77: 18295–18325.
12. Yang SK, Shiue YM, Su ZY, Liu IH, Liu CG. An authentication information exchange scheme in WSN for IoT applications. IEEE Access. 2020; 8: 9728–9738.
13. Ting PY, Tsai JL, Wu TS. Signcryption Method Suitable for Low-Power IoT Devices in a Wireless Sensor Network. IEEE Systems Journal. 2018: 12(3): 2385–2394.
14. Haseeb K, Islam N, Saba T, Rehman A, Mehmood Z. LSDAR:A light-weight structure based data aggregation routing protocol with secure internet of things integrated next-generation sensor networks. Sustainable Cities and Society. 2020; 54: 101995.
15. Deebak BD, Fadi AT. A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks. Ad Hoc Networks. 2020; 97: 102022.
16. Vijayakumar P, Obaidat MS, Azees M, Hafizul Islam SK, Kumar N. Efficient and Secure Anonymous Authentication With Location Privacy for IoT-Based WBANs. IEEE Transactions on Industrial Informatics. 2020; 16(4): 2603–2611.
17. Fan K, Sun S, Yan Z, Pan Q, Li H, Yang Y. A blockchain-based clock synchronization Scheme in IoT. Future Generation Computer Systems. 2019; 101: 524–533.
18. Yıldırım KS, Gürcan Ö. Efficient Time Synchronization in a Wireless Sensor Network by Adaptive Value Tracking. IEEE Transactions on Wireless Communications. 2014; 13: 3650–3664.
19. Skiadopoulos K, Tsipis A, Giannakis K, Koufoudakis G, Christopoulou E, Oikonomou K, et al. Synchronization of data measurements in wireless sensor networks for IoT applications. Ad Hoc Networks. 2019; 89: 47–57.
20. Chen CT. Improved efficient authentication scheme with anonymity in global mobility networks. International Journal of Innovative Computing, Information, and Control. 2013; 9: 3319–3339.
21. Chang C C, Cheng TF. A robust and efficient smart card based remote login mechanism for multi-server architecture. International Journal of Innovative Computing, Information, and Control. 2011; 7: 4589–4602.

22. Burrows M, Abadi M, Needham R. A logic of authentication. ACM Transactions on Computer Systems. 1990; 8: 18–36.

23. Jiang Q, Ma J, Wei F, Tian Y, Shen J, Yang Y. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. Journal of Network and Computer Applications. 2016; 76: 37–48.

24. Amina R, Islam SK H, Biswas GP, Obaidat MS. A robust mutual authentication protocol for WSN with multiple base-stations. Ad Hoc Networks. 2018; 75–76: 1–18.

25. Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009; 31: 1118–1123.

26. Benenson Z, Gartner F, Kesdogan D. User authentication in sensor networks. Proceedings of informatik. 2004 Sep.

27. Watro R, Kong D, Cuti S, Gardiner C, Lynn C, Kruus P. TinyPK: securing sensor networks with public key technology. Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks. 2004; 59–64.

28. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public key cryptosystems. communications of the ACM. 1978; 21(2): 120–126.

29. Diffie W, Hellman ME. Multiuser cryptographic techniques. AFIPS Conference Proceedings. 1976 June 8; 109–112.

30. Wong K, Zheng Y, Cao J, Wang S. A dynamic user authentication scheme for wireless sensor networks. Proceedings of the IEEE international conference on sensor networks, ubiquitous, and trustworthy computing. 2006 Jun; 244–251.

31. Tseng HR, Jan RH, Yang W. An improved dynamic user authentication scheme for wireless sensor networks. Proceedings of IEEE Globecom. 2007 Nov; 986–990.

32. Khan MK, Alghathbar K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. Sensors. 2010; 10: 2450–2459. https://doi.org/10.3390/s100302450 PMID: 22294935

33. Chen TH, Shih WK. A robust mutual authentication protocol for wireless sensor networks. ETRI Journal. 2010; 32: 704–712.

34. Amin R, Islam SK H, Kumar N, Choo KK R. An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. Journal of network and computer applications. 2018; 104: 133–144.

35. Chang CC, Le HD. A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks. IEEE Transactions on Wireless Communications. 2016; 15: 357–366.

36. Challa S, Das AK, Odelu V, Kumar N, Kumari S, Khan MK, et al. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. Computers and Electrical Engineering. 2018; 69: 534–554.

37. Harbi Y, Aliouat Z, Refoufi A, Harous S, Bentaleb A. Enhanced authentication and key management scheme for securing data transmission in the internet of things. Ad Hoc Networks. 2019; 94: 101948.

38. Potlapally NR, Ravi S, Raghunathan A, Jha NK. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Transactions on Mobile Computing. 2006; 5: 128–143.

39. Chang CC, Lee CY, Chiu YC. Enhanced authentication scheme with anonymity for roaming service in global mobility networks. Computer Communications. 2009; 32: 611–618.

40. Tsai J L. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers & Security. 2008; 27: 115–121.

41. Ku WC, Chen SM. Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics. 2004; 50: 204–207.

42. Chen CT, Lee CC. A two-factor authentication scheme with anonymity for multi-server environments. Security and Communication Networks. 2015; 8: 1608–1625.

43. Stallings W. Cryptography and network security. Fifth ed. Pearson Education; 2011.

44. Yang S, Li X. A limitation of BAN logic analysis on a man-in-the-middle attack. Journal of Information and Computing Science. 2006; 3: 131–138.

45. Lee CC, Chang RX, Ko HJ. Improving two novel three-party encrypted key exchange protocols with perfect forward secrecy. International Journal of Foundations of Computer Science. 2010; 21: 979–991.

46. Messergers TS, Dabbish EA, Sloan RH. Examining smart card security under the threat of power analysis attacks. IEEE Transactions on Computers. 2002; 51: 541–552.

47. Lee CC, Lai YM, Chen CT, Chen SD. Advanced Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks. Wireless Personal Communications. 2017; 94: 1281–1296.

48. Tian X, Zhu RW, Wong DS. Improved efficient remote user authentication schemes. International Journal of Network Security. 2007; 4: 149–154.

49. Juang WS, Chen ST, Liaw HT. Robust and efficient password-authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics. 2008; 55: 2551–2556.