The ISC Int'l Journal of Information Security

July 2024, Volume 16, Number 2 (pp. 1–15)

http://www.isecure-journal.org

Security Enhancement of an Authentication Scheme Based on DAC and Intel SGX in WSNs

Mustafa Isam Ahmed Al-Baghdadi [1] and Maryam Rajabzadeh Asaar [1,*]

[1] *Department of Electrical and Computer Engineering, Sicence and Research Branch, Islamic Azad University, Sattari St., Tehran, 10587, Tehran, Iran*

ARTICLE INFO.

Article history:
Received:
Revised:
Accepted:
Published Online:

Keywords:
Dynamic Authentication, Wireless Sensor Network, Authentication

Type: Research Article

doi:

dor:

ABSTRACT

Due to the nature of the public channel, designing authentication techniques suitable for wireless sensor networks (WSNs) that satisfy the dedicated considerations is critical. In 2022, Liu *et al.* presented an authentication protocol that employs dynamic authentication credentials (DACs) and Intel software guard extensions (SGX) to guarantee security in WSNs. Then, they proved that it is secure by formal and informal security analysis. This paper shows that it is not secure against desynchronization and offline guessing attacks for long-term random numbers of users. In addition, it suffers from the known session-specific temporary information attack. Then, an improved authentication scheme using DAC and Intel SGX will be presented to address these vulnerabilities. We show that it is secure against the aforementioned attacks by employing formal and informal analysis and has a reasonable communication and computation overhead. It should be highlighted that our proposal's communication and computation overheads are increased negligibly, but it provides more security features compared to the baseline protocol.

1 Introduction

Wireless sensor networks (WSNs) have progressed with advances in the Internet of Things (IoT) [1, 2]. Recently, WSNs, because of their advantages such as easy development, low computation, and high flexibility, have various applications in intelligent transportation, medical systems, and so on [3]. The gateways (GWNs), users, and sensors are participants in WSNs [4]. The sensors, distributed in a zone by design, collect and transfer information, and GWNs are responsible for managing sensors to transfer information correctly to eligible users. Since the information transformation is done on the public channel [5], these networks suffer from attacks [6]. Furthermore, sensors have limited memory computation and storage capabilities, and they are also placed in unprotected environments. Consequently, designing a secure and efficient authentication scheme in WSNs is vital to prevent eavesdropping and altering messages on the public channel [7–9]. Nonetheless, various authentication and key agreement protocols have been proposed so far; a few consider updating authentication credentials at users, GWNs, and sensors. Hence, they suffer from serious attacks without updating on time [10]. Authentication schemes based on dynamic authentication credentials (DACs) suffer from desynchronization attacks, which causes login failure for the next communication. Hence, it is required to save authentication tables of users and sensors on GWN's

* Corresponding author.
Email addresses: mustafaessam9090@gmail.com, asaar@srbiau.ac.ir

ISSN: 2008-2045

memory in authentication schemes based on DAC, where this issue causes these schemes to not secure against the privileged user attack and table lost attack in a way that an adversary with this information can do impersonation attacks. As a consequence, there is a need for a trusted execution environment (TEE) to store secret keys and authentication table in a secure way. Since a trusted platform module (TPM) is not suitable to protect the sensitive information in the GWN, Liu *et al.* [11] in 2022 adopted the Intel software guard extensions (SGX) and DAC together to present a secure authentication scheme. In fact, they employ SGX to keep the master key of GWN, and also authentication tables and credentials are encrypted by the master secret key of GWN to be protected. Consequently, their scheme is resistant against privileged and authentication leakage attacks.

The contributions of this paper are given below.

- We analyze the authentication scheme based on DAC and Intel SGX presented by Liu *et al.* [11] and prove that it does not support security against desynchronization attacks. However, it is also not secure against offline guessing attacks for long-term random numbers of users. Furthermore, it is not resistant to known session-specific temporary information attacks. Then, a modified authentication scheme is proposed, which tackles the aforementioned weaknesses.
- In the formal security analyses, it is shown that our proposal accomplishes session key security by using Burrow-Abadi-Needham (BAN) logic and ProVerif software. In addition, the informal security analyses prove that our protocol is secure against various kinds of known attacks, such as desynchronization attacks, and it also provides security against offline users' long-term random number guessing attacks.
- Then, the evaluation of our protocol in terms of security features and communication and computation overheads are given, and we compare the results with other schemes to show that not only our proposal can satisfy the necessary security and usability features of IoT-based applications but also it has an acceptable communication and computation costs.

1.1 Related Work

Various authentication schemes have been presented so far to provide security and privacy for users in WSNs, where related schemes are reviewed hereafter. In 2009, Das [12] gave an efficient two-factor authentication scheme, which is based on users' smart card and their passwords. Nevertheless, in 2010, it was proved by Khan and Alghathbar [13] that Das's scheme suffers from gateway bypassing and privileged user attacks and presented an improved authentication scheme to tackle these weaknesses. Then, Vaidya *et al.* [14] showed that the protocol of Khan *et al.* is not secure against stolen smart card attacks. In 2014, an authentication scheme in which smart card information is encrypted to guarantee stored data security was given by Kim *et al.* [15]. However, in 2017, it was proved by Li *et al.* [16] that the scheme is given by Kim *et al.* is not secure against sensor impersonation and offline guessing attacks, and then an improved scheme was proposed. In 2018, Yu *et al.* [17] presented an authentication scheme for vehicular communications in which session keys are dynamically changed to be secure against man-in-the-middle attacks. Unfortunately, it was proved by Sadri and Asaar [18] that it suffers from impersonation and offline guessing attacks, and they proposed a secure scheme. In 2016, Amin *et al.* [19] gave a privacy-preserving authentication scheme, which is three-factor to be resistant to offline guessing attacks. In 2019, Ostad-Sharif *et al.* [20] showed that it is not secure against replay attacks and also it is not forward-secure, and then to address these vulnerabilities, a lightweight authentication scheme was given. In 2020, Chen *et al.* [21] showed that the scheme presented by Ostad-Sharif *et al.* also has some security drawbacks. To address these weaknesses, Chang and Le [22] introduced a forward-secure authentication scheme, then in 2018, Amin *et al.*[23] proved that Chang *et al.*'s scheme not only suffers from stolen smart card and offline guessing attacks, but also it does not satisfy user untraceability. Next, they provided a new and efficient authentication scheme, which is a three-factor authentication that also satisfies users' privacy and traceability. Employing the DAC technique improves the security of authentication protocols [11], and in 2016, Chang *et al.* [24] gave an authentication protocol that supports dynamic identity. In 2019, it was shown by Yang *et al.* [25] that their scheme is not forward secure and is not efficient, and then a lightweight authentication scheme using XOR operations and hash functions was proposed, where it supports the DAC in a way that authentication certificates have been updated in each session. In 2019, an authentication scheme was presented by Agrawal *et al.* [26] the scheme not only employs a trusted platform module (TPM) to increase its security but also uses a technique that detects sensor capture attacks. Fu and Peng [27] also proposed an authentication scheme using TPM to make sensors resistant to capturing attacks. Currently, Tan *et al.* [28] proposed a tamper-detection authentication scheme using TPM. In DAC-based authentication schemes, the use of TPMs is not suitable since TPMs are adequate for static protection. As a consequence, SGX is used for authen-

ticated schemes using the DAC technique. In 2016, Balisane and Martin [29] introduced an SGX-based authentication scheme to address security drawbacks caused by DNS poisoning and wrong SSL. In 2018, Condé *et al.* [30] presented an SGX-based authentication scheme that not only supports more security to verify users' credentials at SGX but also has a lower overhead. Currently, Sun and Xiao [31] gave another SGX-based authentication scheme in which the key is updated dynamically, and also users' certificates are protected by the SGX. Furthermore, the use of the TEE-based credential technique is increased in a way that Kostiainen *et al.* [32] designed a TEE-based scheme for mobile users, and also similarly, Kostiainen and Asokan [33] proposed a construction for remote credential provision in TEE in a secure way. Then, Marfario *et al.* [33] addressed the vulnerability caused by user enrollment for TEEs and proposed constructive changes to make them secure. After that, lots of works [34–37] are done to improve the security of protocols using the TEE. In 2022, Liu *et al.* [11] presented an SGX-based authentication protocol that employs dynamic authentication credentials to enhance its security. Gateways are equipped with the SGX to protect the data in use and also provide a trusted zone for computation to avoid stolen authentication table and privileged entity attacks.

1.2 Organization of the Paper

The rest of this paper is organized as follows. Section 2 presents background information, including the system, security model, and notations used in the paper. Section 3 and Section 4 present the Liu *et al.* scheme [11], and its security analysis, respectively. Then, our proposed protocol and its security analysis are presented in Section 5. Section 8 and Section 9 give the performance analysis and conclusion, respectively.

2 Background

2.1 System Model

The user, a GWN, and a sensor are participants of an authentication protocol in WSNs such that data from sensors are transferred through gateways to users. When users are authenticated by the GWN, it is possible to generate a shared key between a user and a sensor to transmit data from the sensor to the user securely. The GWNs and users do not have limitations in computations, storage and energy consumption, while sensors have limitations in computation, storage and communication in WSNs. The registration phase is done by a secure channel, and messages are exchanged over a secure channel in this phase, while exchanged messages in the login and authentication phase are transferred on the public channel.

2.2 Security Model

The security model to analyze is based on the widely accepted Dolev–Yao (DY) threat model [38]. An adversary in the DY model can resend, alter, omit, and intercept messages on the public channel in transmission. Furthermore, sensors can be captured by adversaries, and their information can be extracted. Moreover, the adversary by side-channel attacks can derive information stored on smart cards [39]. The SGX is considered a TEE in the authentication scheme, and also it is assumed that attacks such as software-based fault injection attacks [40] and foreshadow attacks [41] cannot be applied to the SGX.

2.3 Notations

In this subsection, notations used throughout the manuscript are introduced in Table 1.

Table 1. Notations

Notation	Description
ID_i	Identity of user U_i
ID_G	Identity of gateway GWN
ID_j	Identity of sensor S_j
ID_{SC}	Identity of smart card SC
PW_i	Password of U_i
RTS_i	a random temporary string
K_u, K_s	Secret key of GWN for users and sensors, respectively
PID_i, PID_j	Pseudo identity of U_i and ID_j, respectively
r_j, y_i, y_j	Random numbers selected by GWN
r_i, w_i, x_1, N_i	Random numbers selected by U_i
x_2, x_3, x_4	Random numbers selected by GWN
K_j	Random number selected by S_j
RID_i	Random identity of user U_i
TC_i	Temporary credential of user U_i
PTC_i	Pseudo temporary credential of user U_i
TC_j	Temporary credential of sensor S_j
PTC_j	Pseudo temporary credential of sensor S_j
$h(.)$	One-way hash function
$\|Z\|$	The size of Z
$\oplus$	XOR operation

3 Review of Liu *et al.*'s Scheme

In this section, the details of the protocol are reviewed. Then, its security analysis is given.

3.1 Registration Phase

In this phase, users and sensors are registered by GWN through a secure channel, and since this phase

is not used in security analysis, the details of this phase are not given here [11].

3.2 The Login and Authentication Phase

In this phase, the user U_i, GWN, and sensor S_j authenticates each other, then a session key between U_i and S_j named as SK will be generated for further communication. In what follows, these phases are given.

- **Step 1.** User U_i inters ID_i and PW_i when it inserts its smart card. Then, the smart card computes $r_i = Rr_i \oplus h(ID_{sc}, ID_i, PW_i)$, $RPW_i = h(ID_{sc}, r_i, PW_i)$ and $B_i^* = h(ID_{sc}, RPW_i)$. If B_i^* is equal to B_i, then the user U_i inputs ID_j of a sensor S_j, then the smart card extracts a timestamp T_1, and generates a random number N_i, then computes $TC_i = PTC_i \oplus h(r_i, ID_{sc})$, $q_1 = h(TC_i, ID_j, N_i, r_i)$, $PKS_i = N_i \oplus h(TC_i, r_i, T_1)$, $PID_j = ID_j \oplus h(TC_i, T_1, N_i)$. Then, SC sends to GWN the message $m_1 = \{q_1, PKS_i, PID_j, PTC_i, T_1\}$.
- **Step 2.** If T_1 is fresh, GWN according to PTC_i selects PDK_i and BN_i from its table, and sends (PTC_i, BN_i) along with ID_G to the security interface of SGX. The interface SGX chooses a key K_u and calculates $r_i = BN_i \oplus h(PTC_i, ID_G, K_u)$, $DK_i = PDK_i \oplus r_i$, $TC_i = h(DK_i, r_i)$, $N_i = PKS_i \oplus h(TC_i, r_i, T_1)$, $ID_j = PID_j \oplus h(TC_i, T_1, N_i)$, $KID_j = ID_j \oplus ID_G$, and $q_1^* = h(TC_i, ID_j, N_i, r_i)$. Then, it examines if $q_1^* = q_1$ is held. If it is held, it authenticates the user; otherwise, it rejects messages. If GWN accepts user authentication, it according to KID_j selects (PDK_j, CN_j) and sends (KID_j, CN_j) to the security interface of SGX. Then, SGX selects K_s corresponding to KID_j, and computes $r_j = CN_j \oplus h(ID_j, ID_G, K_s)$. Then, it computes $DK_j = PDK_j \oplus r_j$, $TC_j = h(DK_j, r_j)$, $PID_j = h(DK_j, ID_j)$, $q_2 = h(TC_j \oplus r_i, ID_j)$. Then, GWN extracts T_2, calculates $PKS_N = r_i \oplus h(TC_j, ID_j, T_2)$, and sends S_j the message $m_2 = \{q_2, PKS_N, PID_j, T_2\}$.
- **Step 3.** The sensor S_j checks validity of T_2, if it is not valid, S_j rejects the message; otherwise, it calculates $TC_j = PTC_j \oplus PID_j$, $r_j = PKS_N \oplus h(TC_j, ID_j, T_2)$ and $q_2^* = h(TC_j \oplus jr_i, ID_j)$, and checks if $q_2^* = q_2$ is held. If it is not held, it will reject it. Otherwise, S_j selects a random number K_j, computes $q_3 = h(N_i, K_j, TC_j \oplus ID_j)$ and gets a timestamp T_3, and computes $PKS_j = K_j \oplus h(TC_j, N_i, ID_j, T_3)$ and $SK = h(N_i, K_j)$, and sends to GWN the message $m_3 = \{q_3, PKS_j, T_3\}$.
- **Step 4.** Then, GWN checks validity of T_3, and if it is not valid, it terminates; otherwise, it computes $K_j = PKS_j \oplus h(TC_j, r_j, ID_j, T_3)$ and $q_3^* = h(N_i, K_j, TC_j \oplus ID_j)$. If $q_3^* = q_3$, it accepts K_j, then it computes $PKS_k = K_j \oplus H(TC_j, (N_i \oplus ID_j))$ and $q_4 = h(TC_i, (r_i \oplus N_i), T_4)$, and sends $m_4 = \{q_4, PKS_k, T_4\}$ to U_i.
- **Step 5.** The user U_i checks the freshness of T_4, if it is not fresh, it terminates; otherwise, SC calculates $q_4^* = h(TC_i, (r_i \oplus N_i), T_4)$, and checks if $q_4 = q_4^*$ holds. If it holds, SC calculates $K_j = PKS_K \oplus h(TC_i, (N_i \oplus ID_j))$ and $SK = h(N_i, K_j)$.

3.3 The Dynamic Credentials Update Phase

In this phase, credentials are updated, and the details for updating are given in what follows.

- **Step 1.** For updating credentials when GWN receives T_5, it computes $q_5 = h(K_j, TC_j \oplus T_5)$, $DK_j^{new} = K_j \oplus T_3$, $PDK_j^{new} = DK_j^{new} \oplus r_j$, then it updates PDK_j^{new}, and sends $m_5 = \{q_5, T_5\}$ to S_j.
- **Step 2.** The sensor S_j when receives m_5 checks validity of T_5. If it is valid, it computes $q_5^* = h(K_j, TC_j \oplus T_5)$, and checks if q_5^* is equal to q_5. If the equality holds, S_j computes $DK_j^{new} = K_j \oplus T_3$, $TC_j^{new} = h(DK_j^{new}, r_j)$, $PTC_j^{new} = TC_j^{new} \oplus h(DK_j^{new}, ID_j)$, and updates PTC_j^{new}.
- **Step 3.** For updating credentials, SC with T_6 computes $q_6 = h(N_i, K_j \oplus T_6)$, $DK_i^{new} = N_i \oplus T_4$, $TC_i^{new} = h(DK_i^{new}, r_i)$, $PTC_i^{new} = PTC_i \oplus TC_i \oplus TC_i^{new}$, and updates PTC_i^{new}, and sends $m_6 = \{q_6, T_6\}$ to GWN.
- **Step 4.** When GWN receives m_6, it checks the validity of T_6, If it is fresh, it computes $q_6^* = h(N_i, K_j \oplus T_6)$, and checks if $q*_6$ is equal to q_6. If they are equal, it computes $DK_i^{new} = N_i \oplus T_4$, $PDK_i^{new} = DK_i^{new} \oplus r_i$, $TC_i^{new} = h(DK_i^{new}, r_i)$, $PTC_i^{new} = PTC \oplus TC_i \oplus TC_i^{new}$, $BN_i^{new} = r_i \oplus h(PTC_i^{new}, ID_{GWN}, K_u)$, and updates $\{PTC_i^{new}, PDK_i^{new}, BN_i^{new}\}$ and $\{PTC_i^{new}, K_u\}$.

4 Security Analysis of Liu *et al.*'s Scheme

In this section, it will be shown that Liu *et al.*'s scheme is not secure against desynchronization and offline users' long-term random number guessing attacks, as described below.

4.1 Desynchronization Attacks

This protocol is not secure against desynchronization attacks, in a way that if users and GWN would like to

update their parameters, and an adversary prevents some messages to be reached to the other entity, the next communications will be interrupted, and the session key cannot be generated.

When GWN updates parameters as given in Step 1 of Section 3.3 and sends m_5 to the sensor S_j, an adversary can prevent the message m_5 from being reached S_j. As a consequence, in this case, the information in S_j cannot be updated, which means that Step 2 in Section 3.3 cannot be done, while in the GWN, PDK_j^{new} has been updated. Therefore, in the next session, when GWN sends message m_2 to S_j, the sensor checks the integrity of the message by verifying q_2, and since its parameters such as PTC_j^{new} have not been updated, it rejects message m_2 due to the inequality of PTC_j and PTC_j^{new}, and also inequality of TC_j and TC_j^{new}. As a consequence, the next authentication phase will be interrupted. Similarly, when parameters in the user's smart card can be updated as given in Step 3 of Section 3.3 and the message m_6 is sent to the GWN, an adversary can get this message and avoid reaching it to the GWN. Hence, the related parameters such as TC_i cannot be updated in the GWN, while smart card parameters such as TC_i have been modified to TC_i^{new}. Hence, during the next login and authentication phase, a session key cannot be generated since the authentication fails because of the inequality of parameters in each party.

4.2 Offline Users' Long-Term Random Number Guessing Attack

In this attack, an adversary can obtain the long-term random number of the user, r_i, where it is too important in the protocol since this value is fixed during all sessions. Consequently, the adversary can impersonate a user or can violate the sensor's privacy. To do this attack, the adversary follows the steps described below.

- The adversary with sensor capture attack can obtain PTC_j and has PID_j from message m_2, then it can compute $TC_j = PTC_j \oplus PID_j$ and $N_i = PKS_N \oplus h(TC_j, ID_j, T_2)$, where PKS_N and T_2 have been gotten from the m_2 and ID_j from the sensor.
- Then, it can compute $DK_i^{new} = N_i \oplus T_4$, where T_4 is obtained from m_4, and also it can compute $PID_j \oplus ID_j$.
- Next, the adversary chooses ${TC_i}^A$ and checks if the equation $PID_j \oplus ID_j = h({TC_i}^A, T_1, N_i)$ holds. If it is held, the adversary finds the correct TC_i^A and goes to the next step; otherwise, it chooses another ${TC_i}^A$ and repeats this step again.
- After that, the adversary tries to find the value r_i from the equation $PKS_i \oplus N_i = h(TC_i, r_i^A, T_1)$ by selecting a value ${r_i}^A$. If it holds, it finds the correct value ${r_i}^A$, and it will be successful; otherwise, it selects another r_i^A and repeats this step.

In this attack, the adversary needs to compute two hash values, so it can use some space-time trade-off methods such as the rainbow table to reduce the time complexity of its computations [42]. It should be noted this attack will be successful since the value of r_i is not changed in different sessions.

4.3 Known Session-Specific Temporary Information Attack

The protocol is not secure against the known session-specific temporary information attack in a way that if random numbers used during the authentication phase, such as N_i and K_j are known, then the session key is extracted as $SK = h(N_i, K_j)$. Therefore, the main reason for this vulnerability is that the session key is constructed from random numbers in each session.

5 Our Proposed Protocol

In this section, our proposal including the registration and login and authentication phases is described in details.

5.1 The Registration Phase

In this phase, users and sensors are registered by GWN, which is described in the following section.

(1) **User registration phase**: The following steps are done between a user U_i and a GWN through a secure channel.
- **Step 1.** A user U_i chooses a random number r_i, computes $RID_i = h(ID_i, r_i)$ as its random identity, and transfers RID_i to GWN.
- **Step 2.** The GWN selects a random number y_i and a random temporary string, RTS_i, and calculates $TC_i = h(RID_i, h(y_i), K_u)$, $PTC_i = TC_i \oplus h(K_u, RID_i)$, $PID_i = RID_i \oplus h(K_u, RTS_i)$. Then, it stores RTS_i, PTC_i and PID_i in its memory and (K_u, y_i, RTS_i) in SGX. Then, GWN transfers a smart card, including (RTS_i, TC_i), to the user U_i.
- **Step 3.** The user U_i inputs its password PW_i and its identity ID_i, and a secret random number w_i, and computes $B_1 = h(w_i, ID_i, PW_i)$, $B_2 = B_1 \oplus r_i$, $B_3 =$

$h(RID_i, r_i, B_1)$ and $RTC_i = TC_i \oplus h(B_1)$, and U_i stores $\{B_2, B_3, w_i, RTS_i, RTC_i\}$ in its smart card and deletes TC_i.

(2) **Sensor registration phase**: The following steps are done between a sensor S_j and a GWN through a secure channel.

- **Step 1.** A sensor S_j sends its identity ID_j to GWN.
- **Step 2.** The GWN selects random numbers r_j, y_j and a K_s, and calculates $TC_j = h(y_j, ID_j, r_j)$, $PID_j = h(ID_j, y_j)$, $PTC_j = TC_j \oplus PID_j$, and $Rr_j = r_j \oplus h(ID_j, ID_G, K_s)$. Then, it stores (K_s, ID_j) in its SGX, and Rr_j in its memory, and sends PTC_j to the sensor S_j.

5.2 The Login and Authentication Phase

A mutual authentication between a user U_i, a gateway GWN and a sensor S_j is done, where the details are described in what follows.

- **Step 1.** The user U_i enters ID_i and PW_i when it inserts its smart card. The smart card calculates $B_1^* = h(w_i, ID_i, PW_i)$, $r_i^* = B_2 \oplus B_1^*$, $RID_i^* = h(ID_i, r_i^*)$ and $B_3^* = h(RID_i^*, r_i^*, B_1^*)$ and examines if $B_3^* = B_3$. If it is not held, the smart card rejects the login request; otherwise, it computes $TC_i^* = RTC_i \oplus h(B_1^*)$, and selects a random number x_1 to compute $PTC_i = x_1 \oplus h(TC_i^*)$, $TID_i = h(RID_i^*, TC_i^*, x_1, RTS_i)$, $RID_j = ID_j \oplus h(x_1, TC_i^*)$ and $d_1 = h(TID_i, x_1, ID_j, TC_i^*)$. Then, the user U_i sends
$$m_1 = \{PTC_i, RID_j, d_1, RTS_i\}$$
to GWN.
- **Step 2.** When GWN according to RTS_i from its table finds (RTS_i, PTC_i, PID_i), and sends (RTS_i, PTC_i, PID_i) to SGX. The SGX according to RTS_i calculates $RID_i^* = PID_i \oplus h(K_u, RTS_i)$, and then $TC_i^* = PTC_i \oplus h(K_u, RID_i^*)$. Then, GWN computes $x_1^* = PTC_i \oplus h(TC_i^*,)$, $ID_j = RID_j \oplus h(x_1^*, TC_i^*)$ and $TID_i^* = h(RID_i^*, TC_i^*, x_1^*, RTS_i)$. Then, GWN computes $d_1^* = h(TID_i^*, x_1^*, ID_j^*, TC_i^*)$, and then checks if $d_1^* = d_1$. If it does not hold, it rejects m_1; otherwise, according to ID_j, it finds Rr_j and sends its value to the SGX interface. The SGX has y_j, K_s and Rr_j according to table (ID_j, y_j, K_s), and computes $PID_j = h(ID_j, y_j)$, $r_j = Rr_j \oplus h(ID_j, ID_G, K_s)$ and $TC_j = h(y_j, ID_j, r_j)$, and also the interface SGX chooses a new random number y_j^{new} for ID_j, and computes $PID_j^{new} = h(ID_j, y_j^{new})$, $TC_j^{new} = h(y_j^{new}, ID_j, r_j)$, $PTC_j^{new} = TC_j^{new} \oplus PID_j^{new}$, and sends (TC_j, PTC_j^{new}) to GWN. Then, GWN chooses random numbers x_2, x_3 and x_4 and computes $RTC_j = x_2 \oplus h(TC_j)$, $SK = h(x_1, ID_j, TID_i)$, $RSK = SK \oplus h(x_2, ID_j, TC_j)$, $RTC_j^{new} = PTC_j^{new} \oplus h(TC_j, x_3)$, $Rx_4 = h(TC_j, x_2) \oplus x_4$, $x_5 = h(x_4) \oplus x_3$, and
$$d_2 = h(PTC_j^{new}, x_2, x_3, SK, TC_j),$$
and sends
$$m_2 = \{RSK_s, PID_j, RTC_j, RTC_j^{new}, x_5, d_2, Rx_4\}$$
to the sensor S_j.
- **Step 3.** The sensor S_j with PTC_j computes $TC_j^* = PTC_j \oplus PID_j$, and then computes $x_2^* = RTC_j \oplus h(TC_j^*)$, $SK^* = RSK \oplus h(x_2^*, ID_j, TC_j^*)$, $x_4^* = Rx_4 \oplus h(TC_j^*, x_2^*)$, then $x_3^* = x_5 \oplus h(x_4^*)$, $PTC_j^{new} = RTC_j^{new} \oplus h(TC_j, x_3^*, x_4^*)$ and $PID_j^{new} = PID_j \oplus h(TC_j, x_3^*, x_4^*)$, and accepts SK^* and PTC_j^{new} if $d_2^* = h(PTC_j^{new}, x_2^*, x_3^*, SK^*, TC_j^*)$ is equal to d_2. Then, S_j computes
$$d_3 = h(ID_j, SK^*, PTC_j^{new}),$$
and sends $m_3 = \{d_3\}$ to GWN.
- **Step 4.** The GWN checks if
$$d_3^* = h(ID_j, SK^*, PTC_j^*)$$
is equal to d_3. If not, m_3 will be removed.

  Otherwise, it replaces (PID_j, TC_j) by $(PID_j^{new}, TC_j^{new})$. Then, the SGX generates RTS_i^{new} and y_i^{new}, and calculates $TC_i^{new} = h(RID_i, y_i^{new}, K_u)$, and sends its value to GWN. The GWN computes $PTC_i^{new} = TC_i^{new} \oplus h(x_1^*, RID_i, TC_i^*)$, $d_4 = RTS_i^{new} \oplus h(x_1^*, RID_i^*, TC_i^*)$ and
$$d_5 = h(SK, RTS_i^{new}, TC_i^{new}, x_1^*, RID_i),$$
and sends $m_4 = \{d_4, d_5, PTC_i^{new}\}$ to U_i.
- **Step 5.** The user U_i computes $RTS_i^{new} = d_4 \oplus h(x_1, RID_i^*, TC_i^*)$, $TC_i^{new} = PTC_i^{new} \oplus h(x_1, RID_i^*, TC_i^*)$ and $SK = h(x_1, ID_j, TID_i)$ and computes
$$d_5^* = h(SK, RTS_i^{new}, TC_i^{new}, x_1, RID_i^*),$$
and checks if $d_5^* \stackrel{?}{=} d_5$. If so, then U_i calculates $d_6 = h(RTS_i^{new}, TC_i^{new}, x_1)$, and sends $\{d_6\}$ to GWN, and replaces TC_i with TC_i^{new}, and RTS_i with RTS_i^{new}.
- **Step 6.** The GWN computes
$$d_6^* = h(RTS_i^{new}, TC_i^{new}, x_1),$$
and examines if d_6^* is equal to d_6. If so, it confirms that information on the user side has been updated, and then it updates (RTS_i, TC_i) to $(RTS_i^{new}, TC_i^{new})$.

Remark 1. To avoid desynchronization attacks, it should be mentioned that confirmation of the updating parameters is done with session key confirmation at each entity in our protocol. If an adversary wants to block some messages, the other entity immediately understands since it has not received session key confirmation.

5.3 Password Change Phase

A user U_i can update its password by doing three steps, which are given in the following.

- **Step 1.** The user U_i enters ID_i and PW_i, and the smart card computes $B_1^* = h(w_i, ID_i, PW_i)$, $r_i^* = B_2 \oplus B_1^*$, $RID_i^* = h(ID_i, r_i^*)$ and $B_3^* = h(RID_i^*, r_i^*, B_1^*)$, and checks if $B_3^* = B_3$. If they are equal, the smart card sends an authentication to U_i
- **Step 2.** The user U_i enters its new password PW_i^{new}.
- **Step 3.**The smart card calculates $B_1^{new} = h(w_i, ID_i, PW_i^{new})$, $B_2^{new} = B_1^{new} \oplus r_i$, $B_3^{new} = h(RID_i, r_i, B_1^{new})$ and $RTC_i^{new} = TC_i \oplus h(B_1^{new})$, and updates $\{B_2, B_3, RTC_i\}$ to the new values $\{B_2^{new}, B_3^{new}, RTC_i^{new}\}$.

6 Informal Security Analysis

In this subsection, we show that the proposal is secure according to the security model given in Section 2.2.

- **Desynchronization attacks.** To provide security against desynchronization attacks, a technique in which transmitted messages are related to the previous messages is used. With this technique, if an adversary interrupts any messages between a GWN and a user or a sensor and a GWN, both sides will be affected. For instance, in Step 2 of Section 5, the information related to S_j in the GWN have been generated and message m_2 is sent to S_j, and in Step 3 of Section 5, S_j gets this information and updates PTC_j^{new} if d_2 is valid and then sends d_3 as its confirmation of updating this value to the GWN, and only in this case (PID_j, TC_j) can be updated. Similarly, the user updates their parameters $(TC_i^{new}, RTS_i^{new})$ after checking the validity of d_5, and then the GWN checks the validity of d_6 to confirm updating of $(TC_i^{new}, RTS_i^{new})$ at user's side, and then updates these values in its side. Therefore, our proposal is secure against desynchronization attacks.
- **The sensor anonymity.** This feature guarantees that only U_i, S_j, and the GWN know ID_j in each session. In the proposed protocol, RID_j is the encrypted version of ID_j in the form of $RID_j = ID_j \oplus h(x_1, TC_i^*)$ in message m_1 and in the form of $PID_j = h(ID_j, y_j)$ in m_2. As a consequence, the proposed protocol has the sensor anonymity feature.
- **The forward security.** A protocol provides forward security if a session key is not used in other sessions. In our protocol, the GWN checks the validity of d_1 as a session key between U_i and itself, and this value is valid in the current session since x_1, TC_i, and TID_i are changed during different sessions. Similarly, d_2, d_4, d_5, d_6, d_7, and RSK_s are changed in each session, and cannot be used in other sessions. In addition, the secret key SK is generated as $SK = h(x_1, ID_j, TID_i)$, where x_1 and TID_i are changed in each session, and consequently, the session key is different. Hence, our protocol provides forward secrecy.
- **Authentication table leakage and privileged user attacks.** In this attack, an adversary can access the GWN's database to violate the security of the protocol. In our protocol, the SGX is used to protect the sensitive information of users and sensors, such as the master keys K_s and K_u. Since these keys are 1024 bits, their guessing is difficult. In addition, the stored information in the GWN's memory, such as PTC_i and PID_i, are encrypted by K_u to be protected. As a consequence, the proposal provides security against authentication table leakage and privileged user attacks.
- **User traceability attacks.** In this attack, an adversary can trace a user U_i from some fixed parameters in transmitted messages. In our protocol, messages m_1, m_2, m_3, m_4, and other messages are changed during different sessions because of random numbers. For instance, in message m_1, the values x_1, TC_i, and RTS_i are changed in each session, and also, TC_i and RTS_i are updated for the next session. Therefore, the adversary cannot find a connection between the two messages m_1 and m_1' in two different sessions.
- **Replay attacks.** In this attack, the adversary attempts to resend old messages as a new one without being detected by GWN. In our protocol, for instance, RTS_i is used as a random temporary string, and at the end of each session, its value is updated in the form of RTS_i^{new}. Also, a random number x_1 is used in message m_1. For instance, if an adversary resends an old message such as m_1 including d_1, it cannot be accepted by the GWN since RTS_i and TC_i have been updated, and the equality of d_1 and d_1^* cannot be held. As a consequence, the proposed protocol is secure against replay attacks.
- **Stolen smart card attacks.** In this attack, an

Table 2. Login and authentication phase of our protocol

User(U_i)	**Gateway node (GWN)**	**sensor**(S_j)
Inputs ID_i, PW_i, and computes $B_1^* = h(w_i, ID_i, PW_i)$ $r_i^* = B_1^* \oplus B_2$ $RID_i^* = h(ID_i, r_i^*)$ $B_3^* = h(RID_i^*, r_i^*, B_1^*)$ Checks $B_3^* \stackrel{?}{=} B_3$ $TC_i^* = RTC_i \oplus h(B_1^*)$ Generates a random number x_1 Computes $PTC_i = x_1 \oplus h(TC_i^*)$ $TID_i = h(RID_i, TC_i^*, x_1, RTS_i)$ $RID_j = ID_j \oplus h(x_1, TC_i^*)$ $d_1 = h(TID_i, x_1, ID_j, TC_i^*)$		
$\xrightarrow{m_1=\{PTC_i, RID_j, d_1, RTS_i\}}$		
	Finds (RTS_i, PTC_i, PID_j) according to RTS_i, and sends these values to SGX SGX finds K_u according to RTS_i, and computes $RID_i^* = PID_i \oplus h(K_u, RTS_i)$ $TC_i^* = PTC_i \oplus h(K_u, RID_i^*)$ SGX sends (TC_i^*, RID_i) to GWN, GWN computes $x_1^* = PTC_i \oplus h(TC_i^*)$ $ID_j = RID_j \oplus h(x_1^*, TC_i^*)$ $TID_i^* = h(RID_i^*, TC_i^*, x_1^*, RTS_i)$ $d_1^* = h(TID_i^*, x_1^*, ID_j^*, TC_i^*)$ $d_1^* \stackrel{?}{=} d_1$ GWN finds (Rr_j, ID_j) according to ID_j, and sends these values to SGX SGX finds (K_s, y_j) according to ID_j, and computes $PID_j = h(ID_j, y_j)$ $r_j = Rr_j \oplus h(ID_j, ID_G, K_s)$ $TC_j = h(y_j, ID_j, r_j)$ SGX chooses a new random number y_j^{new} for ID_j, and computes $PID_j^{new} = h(ID_j, y_j^{new})$ $TC_j^{new} = h(y_j^{new}, ID_j, r_j)$ $PTC_j^{new} = TC_j^{new} \oplus PID_j^{new}$ SGX sends (TC_j, PTC_j^{new}) to GWN GWN generates random numbers x_2, x_3 and x_4 Computes $RTC_j = x_2 \oplus h(TC_j)$ $SK = h(x_1, ID_j, TID_i)$ $RSK = SK \oplus h(x_2, ID_j, TC_j)$ $RTC_j^{new} = PTC_j^{new} \oplus h(TC_j, x_3)$ $Rx_4 = h(TC_j, x_2) \oplus x_4$ $x_5 = h(x_4) \oplus x_3$ $d_2 = h(PTC_j^{new}, x_2, x_3, SK, TC_j)$	
		$\xrightarrow{m_2=\{RSK_s, PID_j, RTC_j, RTC_j^{new}, x_5, d_2, Rx_4\}}$
		Computes $TC_j^* = PTC_j \oplus PID_j$ $x_2^* = RTC_j \oplus h(TC_j^*)$ $SK^* = RSK \oplus h(x_2^*, ID_j, TC_j^*)$ $x_4^* = Rx_4 \oplus h(TC_j^*, x_2^*)$ $x_3^* = x_5 \oplus h(x_4^*)$ $PTC_j^{new} = RTC_j^{new} \oplus h(TC_j, x_3^*, x_4^*)$ $PID_j^{new} = RID_j \oplus h(TC_j, x_3^*, x_4^*)$ $d_2^* = h(PTC_j^{new}, x_2^*, x_3^*, SK^*, TC_j^*)$ $d_2^* \stackrel{?}{=} d_2$ $d_3 = h(ID_j, SK^*, PTC_j^{new})$ Updates PTC_j to PTC_j^{new}
		$\xleftarrow{m_3=\{d_3\}}$
	$d_3^* = h(ID_j, SK^*, PTC_j^*)$ $d_3^* \stackrel{?}{=} d_3$ Updates (PID_j, TC_j) to $(PID_j^{new}, TC_j^{new})$ SGX generates two new numbers RTS_i^{new} and y_i^{new} Computes $TC_i^{new} = h(RID_i, y_i^{new}, K_u)$ $d_4 = RTS_i^{new} \oplus h(x_1^*, RID_i^*, TC_i^*)$ $PTC_i^{new} = TC_i^{new} \oplus h(x_1^*, RID_i^*, TC_i^*)$ $d_5 = h(SK, RTS_i^{new}, TC_i^{new}, x_1^*, RID_i^*)$	
$\xleftarrow{m_4=\{d_4, d_5, PTC_i^{new}\}}$		
$RTS_i^{new} = d_4 \oplus h(x_1^*, RID_i^*, TC_i^*)$ $TC_i^{new} = PTC_i^{new} \oplus h(x_1^*, RID_i^*, TC_i^*)$ $SK = h(x_1, ID_j, TID_i)$ $d_5^* = h(SK, RTS_i^{new}, TC_i^{new}, x_1, RID_i)$ $d_5^* \stackrel{?}{=} d_5$ Updates (RTS_i, TC_i) to $(RTS_i^{new}, TC_i^{new})$ Computes $d_6 = h(RTS_i^{new}, TC_i^{new}, x_1)$		
$\xrightarrow{m_5=\{d_6\}}$		
	Computes $d_6^* = h(RTS_i^{new}, TC_i^{new}, x_1)$ $d_6^* \stackrel{?}{=} d_6$ Updates (RTS_i, TC_i) to $(RTS_i^{new}, TC_i^{new})$	

adversary can extract all information stored in the smart card of U_i, $\{B_2, B_3, w_i, RTS_i, RTC_i\}$. The adversary cannot get parameters such as TC_i since this parameter is protected by PW_i, and ID_i. Also, guessing the two parameters PW_i and ID_i is not possible simultaneously. In addition, ID_i is protected by a hash function. Hence, the adversary cannot generate the message m_1. As a consequence, our protocol is secure against stolen smart card attacks.

- **Offline guessing attacks.** An adversary cannot guess ID_i and PW_i from the stored information $\{B_2, B_3, w_i, RTS_i, RTC_i\}$ in the smart card since these parameters (ID_i, PW_i) are protected by hash functions. In addition, the password is updated periodically.
- **Sensor capture attacks.** In this attack, an adversary captures a sensor and extracts the secret information of other sensors, users, and the GWN, and can mount various attacks. In our protocol, when an adversary captures S_j, it obtains PTC_j, and ID_j. Then, the adversary can compute x_2^* and SK^* similar to Step 3, while secret keys of other sensors and users are secure. As a consequence, the protocol is not vulnerable to sensor capture attacks.
- **User impersonation attacks.** In this attack, an adversary would like to generate a valid message m_1 to be accepted by the GWN. In our protocol, the adversary has to compute a valid d_1. For this goal, it needs to know TC_i, TID_i, and ID_j, but it cannot find these values since they are protected by PW_i and ID_i. In addition, the adversary does not have the user's smart card. Therefore, the protocol is secure against impersonation attacks.
- **Resistant to the known session-specific temporary information attacks.** In this attack, the adversary has to generate a valid session key using random numbers in the authentication phase. In our protocol, the session key is composed of x_1, ID_j, and TID_i in the form of $SK = h(x_1, ID_j, TID_i)$, where x_1, ID_j, and TID_i is a random number, the sensor identity, and a random value, respectively. Since ID_j and TID_i are confidential, the adversary with having x_1 and other session-specific temporary information cannot find session keys.

7 The Formal Security Analysis

In this section, the formal security analysis of our proposal using the BAN logic and the ProVerif is given. For this goal, the notations of BAN logic are introduced.

7.1 BAN Logic

The proof is provided by using the BAN logic [11, 43]. The notations of BAN logic are given in Table 3.

- R_1. Nonce verification rule: $\frac{P|\equiv\#(X), P|\equiv Q|\sim\ X}{P|\equiv Q|\equiv X}$
- R_2. Freshness conjuncatenation rule: $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$
- R_3. Seeing rule: $\frac{P\triangleleft(X,Y)}{P\triangleleft X}$
- R_4. Message meaning rule: $\frac{P|\equiv\ P\overset{K}{\leftrightarrow}Q, P\triangleleft\{X\}_k}{P|\equiv Q|\sim\ X}$

Table 3. Notations of BAN logic

Notation	Description
$P \mid\equiv X$	*P believes X*
$P \mid\sim\ X$	*P once said X or P had sent message X*
$P \triangleleft X$	*P sees or receives X*
$P\overset{K}{\rightleftharpoons}X$	*The K is a secret formula which, can be used by P and X to prove their identity to another, because only P and X know the K*
$P \Rightarrow X$	*P has jurisdiction over X*
$\#(X)$	*X is fresh*
$\langle X\rangle_N$	*X is encrypted with N*
$P\overset{K}{\leftrightarrow}Q$	*K is a shared secret key between P and Q*

- R_5. Belief 1: $\frac{P|\equiv Q|\equiv\ (X,Y)}{P|\equiv Q|\equiv\ X}$
- R_6. Belief 2: $\frac{P|\equiv Q|\sim(X,Y)}{P|\equiv Q|\sim\ X}$

7.1.1 Security Goals

The security goals we need to prove are defined as follows.

Goal 1. $GWN \mid\equiv U_i \mid\sim\ x_1$
Goal 2. $GWN \mid\equiv U_i \mid\sim\ ID_j$
Goal 3. $GWN \mid\equiv U_i \mid\sim\ RID_i$
Goal 4. $S_j \mid\equiv GWN \mid\sim\ x_2$
Goal 5. $S_j \mid\equiv GWN \mid\sim\ SK$
Goal 6. $GWN \mid\equiv S_j \mid\equiv\ SK$
Goal 7. $S_j \mid\equiv GWN \mid\sim\ x_3$
Goal 8. $S_j \mid\equiv GWN \mid\sim\ PTC_j^{new}$
Goal 9. $GWN \mid\equiv S_j \mid\equiv\ PTC_j^{new}$
Goal 10. $U_i \mid\equiv GWN \mid\sim\ \{RTS_i^{new}, TC_i^{new}, SK\}$
Goal 11. $U_i \mid\equiv GWN \mid\equiv\ \{RTS_i^{new}, TC_i^{new}, SK\}$
Goal 12. $GWN \mid\equiv U_i \mid\equiv\ \{RTS_i^{new}, TC_i^{new}, SK\}$

7.1.2 Suppositions

The following suppositions used in the proof are listed in what follows.

$\mathbf{s_1}$: $U_i \mid\equiv \#(x_1)$
$\mathbf{s_2}$: $U_i \mid\equiv U_i\overset{TC_i}{\longleftrightarrow}GWN$
$\mathbf{s_3}$: $GWN \mid\equiv GWN\overset{TC_i}{\longleftrightarrow}U_i$
$\mathbf{s_4}$: $GWN \mid\equiv \#(x_2, x_3)$
$\mathbf{s_5}$: $GWN \mid\equiv GWN\overset{TC_j}{\longleftrightarrow}S_j$
$\mathbf{s_6}$: $S_j \mid\equiv S_j\overset{TC_j}{\longleftrightarrow}GWN$
$\mathbf{s_7}$: $S_j \mid\equiv \#(x_2)$

7.1.3 Idealisation

In this section we present an idealized form of our protocol as follows.

$U_i \rightarrow GWN : m_1 = \{l_1, l_2, l_3\}$
$l_1 : \{\langle x_1, T_1 \rangle_{TC_i}\}$
$l_2 : \{\langle ID_j, RID_i, x_1 \rangle_{TC_i}\}$
$l_3 : \{\langle ID_j \rangle_{h(x_1, TC_i)}\}$
$GWN \rightarrow S_j : m_2 = \{l_4, l_5\}$
$l_4 : \{\langle x_2, SK, ID_j \rangle_{TC_j}\}$
$l_5 : \{\langle x_2 \rangle_{TC_j}\}$
$S_j \rightarrow GWN\ m_3 = \{l_6\}$
$l_6 : \{\langle SK, ID_j \rangle_{h(TC_j)}\}$
$GWN \rightarrow S_j : m_4 = \{l_7, l_8, l_9\}$
$l_7 : \{\langle PTC_j^{new} \rangle_{h(TC_j, x_3)}\}$
$l_8 : \{\langle x_4 \rangle_{TC_j}\}$
$l_9 : \{\langle x_3, PTC_j^{new} \rangle_{TC_j}\}$
$S_j \rightarrow GWN\ m_5 = \{l_{10}\}$
$l_{10} : \{\langle PTC_j^{new}, ID_j \rangle_{TC_j}\}$
$GWN \rightarrow U_i : m_6 = \{l_{11}, l_{12}\}$
$l_{11} : \{\langle RTS_i^{new}, TC_i^{new} \rangle_{TC_i^*}\}$
$U_i \rightarrow GWN : m_7 = \{l_{13}\}$
$l_{13} : \{\langle RTS_i^{new}, h(x_1) \rangle_{TC_i^{new}}\}$

7.1.4 Proof

In this subsection, the idealized version of our protocol, suppositions, and BAN logic rules are used to prove the aforementioned security goals.
According to m_1 and R_3 we have:
$P_1 : GWN \triangleleft l_1$
$P_2 : GWN \triangleleft l_2$
$P_3 : GWN \triangleleft l_3$
Based on P_1, l_1, s_3, and R_4 we have:
$P_4 : GWN \mid\equiv U_i \mid\sim x_1$ (*Goal* 1)
According to P_2, s_3, and R_4 we have:
$P_5 : GWN \mid\equiv U_i \mid\sim l_2$
Based on l_2, P_5 and R_6 we have:
$P_6 : GWN \mid\equiv U_i \mid\sim ID_j$ (*Goal* 2)
$P_7 : GWN \mid\equiv U_i \mid\sim RID_i$ (*Goal* 3)
According to m_2 and R_3 we have:
$P_8 : GWN \triangleleft l_4$
$P_9 : GWN \triangleleft l_5$
According to P_9, l_5, s_6 and R_4 we have:
$P_{10} : S_j \mid\equiv GWN \mid\sim x_2$ (*Goal* 4)
According to P_8, s_6 and R_4 we have:
$P_{11} : S_j \mid\equiv GWN \mid\sim l_4$
Based on l_4, P_{11} and R_6 we have:
$P_{12} : S_j \mid\equiv GWN \mid\sim SK$ (*Goal* 5)
Based on m_3 and R_3 we have:
$P_{13} : GWN \triangleleft l_6$
According to P_{13}, l_6, s_5 and R_4 we have:
$P_{14} : GWN \mid\equiv S_j \mid\sim SK$
Based on P_{14}, l_6, s_4 and R_2 we have:
$P_{15} : GWN \mid\equiv \#SK$
Based on P_{14} and P_{15} and R_1 we have:
$P_{16} : GWN \mid\equiv S_j \mid\equiv SK$ (*Goal* 6)
Based on m_4 and R_3 we have:
$P_{17} : S_j \triangleleft l_7$
$P_{18} : S_j \triangleleft l_8$
$P_{19} : S_j \triangleleft l_9$
In line with P_{18}, l_8, s_6 and R_4 we have:
$P_{20} : S_j \mid\equiv GWN \mid\sim x_4$ (*Goal* 7)
In line with P_{19}, s_6 and R_4 we have:
$P_{21} : S_j \mid\equiv GWN \mid\sim l_9$
In line with l_7, P_{17} and R_6 we have:
$P_{22} : S_j \mid\equiv GWN \mid\sim PTC_j^{new}$ (*Goal* 8)
In line with m_5 and R_3 we have:
$P_{23} : GWN \triangleleft l_{10}$
In line with P_{23}, m_5, s_5 and R_4 we have:
$P_{24} : GWN \mid\equiv S_j \mid\sim PTC_j^{new}$
In line with P_{24}, m_5, s_8 and R_1 we have:
$P_{25} : GWN \mid\equiv S_j \mid\equiv PTC_j^{new}$
(*Goal* 9)
According to m_6 and R_3 we have:
$P_{26} : U_i \triangleleft l_{11}$
$P_{27} : U_i \triangleleft l_{12}$
According to P_{26}, s_2 and R_4 we have:
$P_{28} : U_i \mid\equiv GWN \mid\sim l_{11}$
According to P_{28}, l_{11} and R_6 we have:
$P_{29} : U_i \mid\equiv GWN \mid\sim (RTS_i^{new}, TC_i^{new}, SK)$
(*Goal* 10)
According to s_1, l_{11} and R_2 we have:
$P_{30} : U_i \mid\equiv \#l_{11}$
Based on P_{31} and R_1 we have:
$P_{31} : U_i \mid\equiv GWN \mid\equiv (RTS_i^{new}, TC_i^{new}, SK)$
(*Goal* 11)
In line with m_7 and R_3 we have:
$P_{32} : GWN \triangleleft l_{13}$
According to P_{32}, s_4 and R_1 we have:
$P_{33} : GWN \mid\equiv U_i \mid\equiv (RTS_i^{new}, TC_i^{new}, SK)$
(*Goal* 12)

7.2 Security Analysis Using ProVerif

In this subsection, ProVerif as the security verification tool is employed to evaluate the security of the proposal. For this aim, the definitions of the protocol, variables, channels and other parameters are given in Table 4.

Then, the queries are given in Table 5, and finally the results are in Table 6.

The results presented in Table 6 indicate that the authentication process done by users, GWNs, and sensors are successful, so the session key is secure.

8 Performance Analysis

8.1 Computational Overhead

Comparison of our protocol with related protocols in terms of computational cost at the user side, the gateway side, and the sensor side for the login and authentication phase is summarized in Table 7. It should be noted in the calculating computation cost

Table 4. Definitions, channels, variables and events

```
(*--channels--*)
free privatechannel 1: channel  [private].
free privatechannel 2: channel  [private].
free publicchannel 1: channel.
free publicchannel 2: channel.
(*-- constants --*)
free IDsc: bitstring [private].
free IDi: bitstring [private].
free IDGwn: bitstring [private].
free IDj: bitstring [private].
Free SK: bitstring [private].
(*-- shared key --*)
(*-- secret key --*)
free REi: bitstring [private].
free PWi: bitstring [private].
(*--functions--*)
fun xor (bitstring, bitstring): bitstring.
equation forall p: bitstring, q: bitstring; xor(xor(p, q), q)= p.
fun concat(bitstring, bitstring): bitstring.
fun h(bitstring): bitstring.
(*-- events--*)
event startUi (bitstring).
event endUi (bitstring).
event startGW (bitstring).
event endGW (bitstring).
event startSN (bitstring).
event endSN (bitstring).
event startSK (bitstring).
event endSK (bitstring).
```

Table 5. Queries

```
(*-- queries --*)
query IDi: bitstring; inj { event(endUi(ID¬i)) ==> inj {
                    event (startUi(IDi)).
query IDGwn: bitstring; inj { event(endGW(IDGwn)) ==> inj {
                    event (startGW(IDGwn)).
query IDj: bitstring; inj { event(endSN(IDj)) ==> inj {
                    event (startSN(IDj)).
query SK: bitstring; inj { event(endSK(SK)) ==> inj {
                    event (startSK(SK)).
(* query attacker (key ij) query attacker (key ji) *)
(*-- process--*)
Process
((!Ui)|(!GW)|(!SN))
```

Table 6. Result

```
Query inj { event(endUi(ID¬i)) ==> inj { event (startUi(IDi)) is true.
Query inj { event(endGW(IDGwn)) ==> inj { event (startGW(IDGwn))  is true.
Query  inj { event(endSN(IDj)) ==> inj { event (startSN(IDj))  is true.
Query inj { event(endSK(SK)) ==> inj { event (startSK(SK))  is true.
```

of protocols, the most time-consuming operations such as hash evaluation is considered, and the time for XOR operations is negligible to be considered. In Table 7, T_H, T_R, T_e/T_d, T_{se}/T_{sd}, T_{dh}, T_m denote the time required for the hash, Rep, asymmetric encryption and decryption, symmetric encryption and decryption, data hiding and scalar point multiplication operations, respectively.

Table 7. Computation overhead in authentication

protocol	User (U_i)	gateway (GWN)	Sensor(S_j)	Total cost
Gao *et al.* protocol [44]	$9T_H + T_{se} + 2T_{dh}$	$6T_H + T_{se} + T_{sd}$	$4T_H + T_{sd}$	$19T_H + T_{se} + 2T_{sd} + 2T_{dh}$
Fatima *et al.* protocol [45]	$5T_H + T_e$	$8T_H + T_d$	$5T_H$	$18T_H + T_e + T_d$
Jabbari and Mohasefi protocol [46]	$13T_H + T_R + 2T_m$	$10T_H$	$4T_H + 2T_m$	$27T_H + 4T_m + T_R$
Liu *et al.* protocol [11]	$12T_H$	$18T_H$	$8T_H$	$38T_H$
Yu and Park protocol [47]	$11T_H + T_R$	$12T_H$	$6T_H$	$29T_H + T_R$
Our protocol	$12T_H$	$22T_H$	$7T_H$	$41T_H$

As given in Table 7, the computational overhead at the user side in our protocol contains 12 hash operations as given in Steps 1 and 5 of Section 5.2. As a consequence, the computational cost at the user side is $12T_H$. In addition, the computational cost of the GWN includes 17 hash operations in Step 2, 4 hash operations in Step 4, and one hash operation in Step 6 as given in Section 5.2. Hence, the total cost at the GWN is $22T_H$. Furthermore, the computational cost of S_j includes 7 hash operations as given in the Step 3 of Section 5.2. Therefore, the total computational cost in our protocol is $41T_H$. As a consequence, this value is slightly increased compared to that of baseline schemes.

8.2 Communication Overhead

The communication cost of protocols includes the size of messages exchanged between entities. Since the message m_1 in our protocol is $m_1 = \{PTC_i, RID_j, d_1, RTS_i\}$, so its size is $4|H(.)|$, the message m_2 is our protocol is $m_2 = \{RSK_s, PID_j, RTC_j, RTC_j^{new}, x_5, d_2, Rx_4\}$, so $|m_2|= 7|H(.)|$. The messages m_3, m_4, and m_5 are $\{d_3\}$, $\{d_4, d_5, PTC_i^{new}\}$ and $\{d_6\}$, respectively. Hence, their sizes are $|m_3|= |H(.)|$, $|m_4|= 3|H(.)|$, and $m_5 = |H(.)|$, respectively. Therefore, the communication cost in our protocol is $16|H(.)|$.

8.3 Experimental Results

In this part, the efficiency of our protocol is compared with related schemes [11, 44–47]. These protocols are implemented on a personal computer (Intel(R) Core TMi7-4710HQ 2.50 GHz processor, 4 GB memory and Windows 8 operating system) using MIRACL library [48]. For the security level of 2^{80}, it is assumed that T_H and T_R takes 0.5 ms, T_e/T_d and T_m take 50.3 ms, T_{se}/T_{sd} takes 0.5 ms, and also T_{dh} takes 1.2 ms. Let $|H(.)|= 160$ bits and $|T|= 32$ bits. The total computational cost of our protocol is 20.5 ms, while its value for related protocols is summarized in Table 8. In addition, the communication overhead of our protocol is 2560 bits $= 16|H(.)|= 16 \times 160$. Therefore, this value for related protocols and ours is given in Table 8.

8.4 Security Features Comparison

In Table 9, the security features of the proposed protocol and related ones [11, 44–47] are given. According to the results of Table 9, the existing protocols cannot resist various attacks. All protocols except for [11] cannot support dynamic authentication credentials to provide more security features. In addition, protocols given in [44, 45] cannot guarantee forward security. Also they cannot provide security against authentication table leakage attacks. As a consequence, the proposed protocol satisfies more security features than them.

9 Conclusion

In this paper, we showed that Liu *et al.*'s authentication scheme is not secure against desynchronization and offline users' long-term random number guessing attacks. Then, a modified authentication scheme using the SGX and the DAC was proposed in a way that it is secure against the aforementioned attacks with the informal security analysis, BAN logic, and ProVerif. Then, its performance analysis clarified that the modified scheme from the point of computation cost is as efficient as Liu *et al.* 's scheme, while its communication cost has been increased slightly. Finally, performance evaluation demonstrates that the proposal not only is practical but also has security requirements of IoT authentication protocols.

10 Declarations

Ethical Approval

Ethical approval does not apply to this manuscript as it does not contain any studies with human participants or animals.

Conflict of Interests

The authors state that they have no conflict of interest.

Table 8. Communication and computation cost in the authentication process

Protocol	Total execution time (ms)	Communication cost
Gao *et al.* [44]	13.7	≈ 300 bytes
Fatima *et al.* [45]	109	≈ 300 bytes
Jabbari and Mohasefi [46]	215.2	≈ 444 bytes
Liu *et al.* [11]	19	≈ 284 bytes
Yu and Park [47]	15	≈ 276 bytes
Our protocol	20.5	≈ 320 bytes

Table 9. comparison of security features

Security features	Jabbari and Mohasefi's protocol [46]	Yu and Park's protocol [47]	Fatima *et al.* scheme [45]	Gao *et al.* protocol [44]	Liu *et al.*'s scheme [11]	Our protocol
Resistant to the replay attack	Y	Y	Y	Y	Y	Y
Resistant to the user impersonation attack	Y	Y	Y	Y	Y	Y
Resistant to the offline users' long-term random number guessing attack	Y	N	Y	Y	N	Y
Resistant to the stolen smart card attack	N	N	Y	Y	Y	Y
Resistant to the desynchronization attack	Y	N	Y	Y	N	Y
Resistant to the known session-specific temporary information attack	N	N	Y	Y	N	Y
Provide forward secrecy	Y	Y	Y	Y	Y	Y
Provide authentication table leakage attack	Y	N	Y	N	Y	Y
Provide dynamic authentication credential	N	N	N	N	Y	Y

Note: Y and N denote yes and no, respectively.

Funding

No funds, grants, or other support was received.

Author Contributions

All authors contributed to the study conception and design, read and approved the final manuscript.

Availability of Data and Materials

Data sharing does not apply to this manuscript as no data sets were generated or analyzed during the current study.

References

[1] L. Zhu X. Du M. Shen, X. Tang and M. Guizani. Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities. *IEEE Internet of Things Journal*, 6(5):7702–7712, 2019.

[2] N. Kumar X. Jia, D. He and K.-K. R. Choo. Authenticated key agreement scheme for fog-driven iot healthcare system. *Wireless Networks*, 25(8):4737–4750, 2019.

[3] K. Wu M. Cao S. Jiang H. Fu, G. Manogaran and A. Yang. Intelligent decision-making of online shopping behavior based on internet of things. *nternational Journal of Information Management*, 50:515–525, 2020.

[4] M. S. Farash T. Shon S. A. Chaudhry, H. Naqvi and M. Sher. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. *Journal of Supercomputing*, 74:3504–3520, 2015.

[5] A. Jabbari and J. Bagherzadeh. A revised key agreement protocol based on chaotic maps. *Nonlinear Dynamics*, 78(1):669–680, 2014.

[6] S. Fan M. Ma, D. He and D. Feng. Certificateless searchable public key encryption scheme secure against keyword guessing attacks for smart healthcare. *Journal of Information Security and Applications*, 50(102429), 2020.

[7] M. M. Modiri; J. Mohajeri; M. Salmasizadeh. Gslha: Group-based secure lightweight handover authentication protocol for m2m communication. *The ISC International Journal of Information Security (ISeCure 2020)*, 12(2):101–111, 2020.

[8] V. Chegeni; H. Haj Seyyed Javadi; M. R. Moazami Goudarzi; A. Rezakhani. Providing a hybrid cryptography algorithm for lightweight authentication protocol in rfid with urban traffic usage case. *The ISC International Journal of Information Security (ISeCure 2021)*, 13(1):73–85, 2021.

[9] F. B. Bayatiani; H. Mala. A lightweight rfid grouping proof protocol with forward secrecy and resistant to reader compromised attack. *The ISC International Journal of Information Security (ISeCure 2023)*, 15(3):1–12, 2023.

[10] R. Zhang X. Liu and M. Zhao. A robust authentication scheme with dynamic password for wireless body area networks. *Computer Networks*, 116:220–234, 2019.

[11] J. Ma X. Liu, Z. Guo and Y. Song. A secure authentication scheme for wireless sensor networks based on dac and intel sgx. *IEEE Internet of Things Journal*, 9(5):3533–3547, 2021.

[12] M. L. Das. Two-factor user authentication in wireless sensor networks. *IEEE transactions on wireless communications*, 8(3):1086–1090, 2009.

[13] M. K. Khan and K. Alghathbar. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks. *Sensors*, 10(3):2450–2459, 2010.

[14] D. Makrakis B. Vaidya and H. T. Mouftah. Improved two-factor user authentication in wireless sensor networks. In *Proc. of the 6-th International Conference on international conference on wireless and mobile computing, networking and communications*, pages 600–606, Niagara Falls, ON, Canada, 11-13 October 2010. IEEE.

[15] W. Jeon Y. Lee J. Kim, D. Lee and D. Won. Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. *Sensors*, 14(4):6443–6462, 2014.

[16] Z. Xiong J. Li, Y. Ding and S. Liu. An improved two-factor mutual authentication scheme with key agreement in wireless sensor networks. *KSII Transactions on Internet and Information Systems*, 11(11):5556–5573, 2017.

[17] K. Lee K. Park S. Yu, J. Lee and Y. Park. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors*, 18(10):doi: 10.3390/s18103191, 2018.

[18] M. J. Sadri and M. R. Asaar. A lightweight anonymous two-factor authentication protocol for wireless sensor networks in internet of vehicles. *Internatonal Journal of Communication Systems*, 33(14):e4511, 2020.

[19] G. P. Biswas M. K. Khan L. Leng R. Amin, S. K. H. Islam and N. Kumar. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Computer Networks*, 101:42–62, 2016.

[20] M. Nikooghadam A. Ostad-Sharif, H. Arshad and D. Abbasinezhad-Mood. Three party secure data transmission in iot networks through design of a lightweight authenticated key agreement scheme. *Future Generation Computer Systems*, 100(2):882–892, 2019.

[21] C.-C. Lee C.-T. Chen and I.-C. Lin. Efficient and secure three-party mutual authentication key agreement protocol for wsns in iot environments. *PLoS ONE*, 15(4):https://doi.org/10.1371/journal.pone.0232277, 2020.

[22] C. C. Chang and H. D. Lee. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Transactions on wireless communications*, 15(1):357–366, 2016.

[23] N. Kumar R. Amin, S. K. H. Islam and K.-K. R. Choo. An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. *Journal of network and computer applications*, 104:133–144, 2018.

[24] W.-Y. Hsueh C.-C. Chang and T.-F. Cheng. A dynamic user authentication and key agreement scheme for heterogeneous wireless sensor networks. *Wireless Personal Communications*, 89(2):447–465, 2016.

[25] Y. Sun Z. Yang, J. Lai and J. Zhou. A novel authenticated key agreement protocol with dynamic credential for wsns. *ACM Transactions on Sensor Networks (TOSN)*, 15(2):1–27, 2019.

[26] M. L. Das S. Agrawal and J. Lopez. Detection of node capture attack in wireless sensor networks. *IEEE Systems Journal*, 13(1):238–247, 2018.

[27] D. Fu and X. Peng. Tpm-based remote attestation for wireless sensor networks. *Tsinghua Science and Technology*, 21(3):312–321, 2016.

[28] W. Hu H. Tan and S. Jha. A remote attestation protocol with trusted platform modules (tpms) in wireless sensor networks. *Security and Communication Networks*, 8(13):2171–2188, 2015.

[29] R. A. Balisane and A. Martin. Trusted execution environment-based authentication gauge (teebag). In *Proc. of the 2016 New Security Paradigms Workshop (NSPW2016)*, pages 61–67, New York, NY, USA, 26-29 September 2016. ACM.

[30] C. A. Maziero R. C. R. Condé and N. C. Will. Using intel sgx to protect authentication credentials in an untrusted operating system. In *Proc. of the 2018 IEEE Symposium on Computers and Communications (ISCC2018)*, pages 00158–00163., Natal, Brazil, 25-28 June 2018. IEEE.

[31] H. Sun and S. Xiao. Dna-x: Dynamic network authentication using sgx. In *Proc. of the 2nd International Conference on Cryptography, Security and Privacy (ICCSP 2018)*, pages 110–115, Guiyang, China, 16-19 March 2018. ACM.

[32] N. Asokan K. Kostiainen and J.-E. Ekberg. Credential disabling from trusted execution environments. In *Proc. of the 15th Nordic Confer-*

ence on Secure IT Systems: Information Security Technology for Applications (NordSec2010), pages 110–115, Espoo, Finland, 27-29 October 2010. Springer, Berlin, Heidelberg.

[33] K. Kostiainen and N. Asokan. Credential life cycle management in open credential platforms. In *Proc. of the 6th ACM workshop on Scalable trusted computing (STC2011)*, pages 65–70, Chicago Illinois, USA, 17 October 2011. ACM.

[34] J.-F Lalande G. Arfaoui, S. Gharout and J. Traoré. Practical and privacy-preserving tee migration. In *Proc. of Information Security Theory and Practice: 9th IFIP WG 11.2 International Conference (WISTP 2015)*, pages 93–98, Heraklion, Crete, Greece, 24-25 August 2015. Springer, Berlin, Heidelberg.

[35] R. N. Akram C. Shepherd and K. Markantonakis. Establishing mutually trusted channels for remote sensing devices with trusted execution environments. In *Proc. of the 12th International Conference on Availability, Reliability and Security (ARES 2017)*, pages 1–10, Reggio Calabria, Italy, 29 August-1 September 2017. ACM.

[36] R. N. Akram C. Shepherd and K. Markantonakis. Remote credential management with mutual attestation for trusted execution environments. In *Proc. of the 12th International Conference on Availability, Reliability and Security (ARES 2017)*, pages 1–10, Reggio, Calabria, Italy, 29 August- 1 September 2017. ACM.

[37] Y. Omori and T. Yamashita. Extended inter-device digital rights sharing and transfer based on device-owner equality verification using homomorphic encryption. *IEICE TRANSACTIONS on Information and Systems*, 103(6):1339–1354, 2020.

[38] M. Wazid J. Srinivas, A. K. Das and N. Kumar. Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things. *IEEE Transactions on Dependable and Secure Computing*, 17(6):1133–1146, 2018.

[39] E. A. Dabbish T. S. Messerges and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on computers*, 51(5):541–552, 2002.

[40] F. D. Garcia J. V. Bulck D. Gruss K. Murdock, D. Oswald and F. Piessens. Plundervolt: Software-based fault injection attacks against intel sgx. In *Proc. of the IEEE Symposium on Security and Privacy (SP)*, pages 1466–1482, San Francisco, CA, USA, 18-21 May 2020. IEEE.

[41] J. V. Bulck et al. Foreshadow: Extracting the keys to the intel sgx kingdom with transient out-of-order execution. In *Proc. of the 27th USENIX Security Symposium*, pages 991–1008, Baltimore, MD, USA, 15–17 August 2018. IEEE.

[42] P. Oechslin. Making a faster cryptanalytic time-memory trade-off. In *Proc. of the 23rd Annual International Cryptology Conference on Advances in Cryptology-CRYPTO 2003*, pages 617–630, Santa Barbara, California, USA, August 17-21 2003. Springer Berlin Heidelberg.

[43] R.M. Needham M. Burrows M, M. Abadi. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[44] Z. Xia G. Gao, Z. Feng. Energy efficient three-factor authentication in wireless sensor networks with resisting insider attacks. *IEEE Transactions on Green Communications and Networking*, 7(3):https://doi.org/10.1145/3607142, 2023.

[45] K. Mahmood S. Shamshad M.A. Saleem M.F. Ayub M. N. Fatima, M.S. Obaidat. Privacy-preserving three-factor authentication protocol for wireless sensor networks deployed in agricultural field. *ACM Transactions on Sensor Networks*, page https://doi.org/10.1145/3607142, 2023.

[46] A. Jabbari and J. B. Mohasefi. User-sensor mutual authenticated key establishment scheme for critical applications in wireless sensor networks. *Wireless Networks*, 27:227–248, 2020.

[47] S. J. Yu and Y. Park. Slua-wsn: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks. *Sensors*, 20(15):4143–1354, 2020.

[48] MIRACL Cryptographic Library: Multiprecision Integer and Rational Arithmetic C/C++ Library. Available at https://www.shamus.ie.

Mustafa Isam Ahmed Al-Baghdadi received his B.S. degree in Communication Engineering from Iraq University College, Basra, Iraq in 2014, and now he is a M.Sc. student in Electrical Engineering from Science and Research Branch, Islamic Azad University. His research interests include Network Security.

Maryam Rajabzadeh Asaar received her M.Sc. and Ph.D. degrees in Electrical Engineering from Sharif University of Technology. She is an assistant professor at Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University. Her research interests include Cryptographic Protocols and Network Security.