sensors

MDPI

Article

An Enhanced User Authentication and Key Agreement Scheme for Wireless Sensor Networks Tailored for IoT

Pooja Tyagi [1], Saru Kumari [1], Bander A. Alzahrani [2], Anshay Gupta [3] and Ming-Hour Yang [4,*]

[1] Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India
[2] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[3] Department of Computer Science and Engineering, HMR Institute of Technology and Management, New Delhi 110036, India
[4] Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan 320314, Taiwan
* Correspondence: mhyang@cycu.edu.tw

Abstract: A security protocol for wireless transmission is essential to defend sensitive information from malicious enemies by providing a variety of facilities such as privacy of the user's information, secure session key, associated authentication, and user-repeal facility when a person's authorizations are suddenly disclosed. Singh et al. proposed an improved user authentication and key agreement system for wireless sensor networks (WSNs). Authors are sure that their protocol is secure from various attacks. Here, we find several security pitfalls in their scheme, such as an offline password-guessing attack, failure to protect the session key, and a man-in-the-middle attack. To remove the identified pitfalls found in Singh et al.'s scheme, we design an enhanced authentication scheme for WSNs tailored for IoT. We prove the reliability of our proposed protocol using the real or random (RoR) model. We also evaluate the proposed scheme with the associated schemes and show its superior efficacy as compared to its counterparts.

Keywords: key agreement; smart card; user authentication; wireless sensor networks

check for updates

Citation: Tyagi, P.; Kumari, S.; Alzahrani, B.A.; Gupta, A.; Yang, M.-H. An Enhanced User Authentication and Key Agreement Scheme for Wireless Sensor Networks Tailored for IoT. *Sensors* **2022**, *22*, 8793. https://doi.org/10.3390/s22228793

Academic Editor: Antonio Guerrieri

Received: 18 October 2022
Accepted: 10 November 2022
Published: 14 November 2022

1. Introduction

A wireless Sensor Network (WSN) consists of sensors or sensor nodes and plays a vital role in the Internet of Things (IoT) applications. The sensor nodes can be used at sensitive places in an unplanned or planned way. These kinds of nodes have the capacity to collect data from their neighboring fields, after which they send the data to nearby base stations (BSs), which process the received data for decision-making. Sensor nodes can communicate with each other via wireless radio communications. In WSNs, the BS (referred to as the gateway node (or GW-node)) is the most effective node, whereas sensors are the least effective nodes in regard to battery power, memory space, and computational ability.

WSNs can be utilized in different unattended fields, such as the army, climatic, medical, and agriculture, for goal monitoring, battleground vigilance, and invader identification. WSNs can also be deployed in various IoT applications such as smart homes, smart supply-chain management, smart cities, smart grids, smart traffic management, and industrial Internet. Due to the unattended surroundings of sensor nodes, an adversary has the ability to immediately capture a sensor node from the goal-tracking area. For this reason, an adversary has a possibility to at once seize a sensor node from the goal field and extract all of the data from its reminiscence, as nodes are not usually tamper-resistant because of their low cost.

The requirement to protect the data stored in WSNs is a crucial issue. Here we discuss some scenarios which necessitate a user verification protocol in WSNs. In WSNs, an intruder can create a bug in the network and can disturb or discontinue the commuted texts. Numerous crucial operations in WSNs, along with the utilization of the battleground surveillance and e-health services, rely on actual data to obtain which the users establish a direct connection with the sensor nodes with the help of the base station. Thus, in applications requiring actual data, the user's authenticity is verified by the sensor nodes and the BS, which requires setting up a secret session key among the users and the accessed nodes so that no intruder can obtain the crucial data from the sensor nodes. Due to this motive, user authentication and key agreement protocols in WSNs are important research fields.

Motivation and Contribution

WSNs' environment is challenging due to their resource-constrained nature and security requirements. The same is true for any IoT application due to its dependency on WSNs. Authentication and key agreement schemes are utilized to handle the application layer in WSNs/IoT. In these schemes, balancing efficiency and security is a big challenge. In this regard, research on establishing authentication and key agreement is in the developing stage. While reviewing some literature on authentication and key agreement scheme for WSN, we came across an article in which we felt scope for improvement. The contributions of this article are as follows:

- We analyze an authentication and key agreement scheme for WSNs and point out its flaws.
- As an enhancement of the analyzed scheme, we propose an authentication and key agreement scheme for WSNs tailored for the IoT.
- We have tried to achieve the maximum possible security features while keeping the minimum possible computational load.

2. Related Work

In 2009, Das [1] proposed two-factor user authentications in WSNs. The author claims that the proposed scheme resists many attacks in WSNs, but it suffers from denial-of-service and node compromise attacks. In 2011, Yeh et al. [2] worked on Das's scheme and proposed an authentication protocol for WSNs using elliptic curve cryptography. Mutual authentication is important to prove the legitimacy of each party, and Yeh et al. [2] found that the scheme in [1] does not provide mutual authentication; they also found that the protocol in [1] suffers from insider attacks, user impersonation attacks, and no provision for changing updating passwords.

In 2013, Xue et al. [3] proposed a temporal credential-based mutual authentication scheme for WSNs. In 2014, Turkanovic et al. [4] suggested a user-mutual authentication key agreement protocol for heterogeneous ad hoc WSNs. This scheme concentrates on the Internet of things (IoT) notion. In 2015, Jiang et al. [5] found some security flaws in the protocol [3] and proposed an improvement of [3]. Jiang et al. found that the scheme in [3] suffers from insider attacks, weak stolen smart card attacks, identity guessing attacks, and tracking attacks. In 2015, He et al. [6] proposed mutual authentication and key agreement protocol for WSNs. He et al. [6] found that the scheme in [3] cannot withstand the security parameters, and protocol [3] suffers from offline password guessing attacks, user impersonation attacks, and sensor node impersonation attacks. He et al. [6] found that the scheme in [3] cannot provide legitimacy of the user.

In 2016, Kumari et al. [7] pointed out that the scheme in [6] has many disadvantages. Kumari et al. [7] showed that protocol [6] suffers from offline password-guessing attacks, session-specific temporary information attacks, the absence of password-changing facilities, and the absence of unauthorized login detection, and it does not provide legitimacy to the user. Kumari et al. [7] proposed a mutual authentication and key agreement scheme for WSNs using chaotic maps. In 2016, Jiang et al. [8] worked on the scheme in [6], and after analysis, they showed that the scheme in [6] fails to provide anonymity for the user. Jiang et al. [8] found that in the protocol [6], an adversary can easily track the user, and also, this scheme cannot stand with stolen smart card attacks. Jiang et al. [8] proposed an untraceable temporal-credential-based authentication scheme for WSNs.

In 2016, Farash et al. [9] noticed that the protocol in [4] does not resist man-in-the-middle attacks, the disclosure of the session key, sensor node impersonation attacks, and the disclosure of secret parameters. Farash et al. [9] also showed that the scheme in [4] does not provide sensor node anonymity, and any adversary who wants to track the user can easily do so. To remove the weakness, Farash et al. [9] proposed an enhanced scheme. In 2016, Amin and Biswas [10] found that the protocol in [4] undergoes offline password-guessing attacks, offline identity-guessing attacks, smart card theft attacks, user impersonation attacks, sensor-node impersonation attacks, and inefficient authentication phase. Amin and Biswas [10] also gave an authentication method after removing the weakness of the scheme in [4], and they claimed the enhanced security of their scheme over protocol in [4]. Amin and Biswas used BAN logic for formal security analysis of the proposed protocol.

In 2016, Amin et al. [11] showed that the scheme in [9] is vulnerable to smart card stolen attacks, offline password-guessing attacks, new smart issue attacks, user impersonation attacks, and known session-specific temporary information attacks. Amin et al. [11] found that the protocol in [9] does not provide user anonymity, and also the secret key of the gateway node is not safe in this protocol. To overcome the disadvantages of the scheme in [9], Amin et al. [11] put forward an improved version of WSNs. In 2016, Chang and Le [12] showed that protocol in [4] is vulnerable to impersonation attacks with node capture, stolen smart card attacks, sensor node spoofing attacks, and stolen verifier attacks, and it fails to ensure backward secrecy. To remove these security issues, Chang and Le [12] proposed an advanced scheme.

In 2017, Wu et al. [13] revealed that the scheme in [10] suffers from sensor capture attacks, session key leakage attacks, user forgery attacks, gateway forgery attacks, and sensor forgery attacks. Wu et al. [13] showed that in the scheme [10], the adversary could track the user easily, and this does not provide mutual authentication between parties. To remove the disadvantages in [10], Wu et al. [13] proposed an authentication scheme for multi-gateway-based WSNs. In 2017, Wu et al. [14] found that the scheme in [5] has many security pitfalls, such as offline password-guessing attacks, user forgery attacks, de-synchronization attacks, and a lack of strong forward security. Wu et al. [14] recommended a stepped-forward version of the protocol in [5]. They claimed their protocol to be secure. In 2017, Dhillon and Kalra [15] proposed multi-factor remote user authentication and key agreement scheme for IoT environments. They said that their proposal is to be defendable against all prospected threats.

In 2018, Amin et al. [16] designed a robust patient monitoring system using wireless medical sensor networks. They are sure that their protocol is secured against obvious violations. They used BAN logic to confirm the mutual authentication feature for their suggested protocol. In 2018, Jangirala et al. [17] designed an authentication and key agreement protocol for the industrial Internet of things. In the scheme of [17], they used the fuzzy extractor method for biometric authentication by the user’s smart card. In 2018, Li et al. [18] pointed out that the protocol in [8] cannot resist known session-specific temporary information attacks and clock synchronization, and this scheme is not applicable to IoT environments. To improve the scheme in [8], Li et al. [18] suggested a three-factor anonymous authentication scheme for WSNs. In 2018, He et al. [19] found that the scheme in [12] suffers from sensor capture attacks. They gave an improved version of [12].

In 2019, Gupta et al. [20] designed an anonymous user authentication and key-establishment scheme for wearable devices. They used BAN logic to justify mutual verification among the gateway/cellular terminal and the wearable gadget of the sensor. In 2019, Ghani et al. [21] proposed an IoT-based scheme for WSNs using a symmetric key. In 2020, Lee et al. [22] discovered that the protocol in [15] is vulnerable to a stolen mobile device attack and a user impersonation attack, and it lacks a provision for the agreement of the session key. In 2021, Mall et al. [23] proposed a physically unclonable function (PUF) based authentication protocol for drone-enabled WSNs. This protocol conducts communication among devices and the cloud by the relocatable drone.

In 2021, Chen et al. [24] suggested a group key agreement protocol for IoT. In this scheme, they introduce an entity known as the device manager. Device managers connect IoT devices with blockchain networks. In 2021, Chen and Liu [25] suggested a three-factor scheme for the IoT that used biological information. They proved their protocol in both a formal and informal manner. In 2021, Ali et al. [26] designed an ECC-based protocol for vehicle-to-vehicle communication in VANETs. In 2021, Sadri and Asaar [27] showed that the scheme in [21] has many security pitfalls, such as user impersonation attacks, malicious gateway attacks, and traceability attacks. To remove weaknesses found in [21], Sadri and Asaar [27] suggested a hash-based scheme for WSNs in IoT with forward secrecy. They analyzed their protocol with both formal and informal methods. In 2021, Rangwani et al. [28] proposed a three-factor scheme for the Industrial Internet of Things (IIoT). They also verified their scheme in both formal and informal ways. In 2021, Nashwan [29] designed a scheme for healthcare IoT. In this scheme, mutual authentication between all nodes is verified using BAN logic.

In 2022, Tanveer et al. [30] suggested a resource-capable scheme for the Industrial Internet of Things (IIoT). The authors claimed that their scheme is suitable for resource-constrained smart devices. In 2022, Kumar et al. [31] designed an RFID-based scheme using PUF for vehicular cloud computing. In 2022, Wu et al. [32] suggested a scheme that depends on a symmetric encryption algorithm and fog computing in the Internet of vehicles. In 2022, Li et al. [33] proposed a protocol for fog-enabled social internet of vehicles.

In 2016, Singh et al. [34] proposed a scheme to resolve the weaknesses of the protocol in [4]. Singh et al. claimed their scheme to be more secure and efficient for a real application environment. However, we show that the protocol in [34] has many security issues. The scheme in [34] suffers from many attacks like offline password-guessing attacks, man-in-the-middle attacks, and attacks on the session key.

Organization

In Section 3, we review Singh et al.'s scheme. The cryptanalysis of Singh et al.'s scheme is shown in Section 4. Section 5 describes our enhanced scheme. Section 6 explains the security analysis of the proposed scheme in both a formal and informal manner. Section 7 contains a comparison of the proposed scheme with some related schemes. The conclusion is in Section 8.

3. Review of Singh et al.'s Scheme

Firstly, we write the notations and their explanations used in this paper in Figure 1.

Symbol	Description
U_i, ID_i	i^{th} User and user's identity
SC	Smart Card
S_j, ID_{sj}, PW_{sj}	j^{th} Sensor Node, its identity and password
PW_i	i^{th} User's password
K_{GW}	Secure password recognized only to Gateway Node
$K_{GW\text{-}Ui}$	Secure password key shared with the User i and gateway node
$K_{GW\text{-}Sj}$	Secure password shared with the sensor node j and gateway node
T	Timestamp
SK	Separately computed session key with private information of both user and sensor node
$\oplus$, $\|$, $h(.)$	XOR, concatenation, a lightweight one way hash function
U_a	Adversary

Figure 1. Notations and description.

3.1. Registration Phase

The system of registration begins after the placement of sensor nodes in the application space. The registration phase is split into two sub-phases. Phase one is between a user and the gateway, and phase two is between the sensor node and the gateway. Figure 2 illustrates all two stages.

User U_i Gateway node GW	Sensor node S_j
Registration phase between user and gateway	Registration phase between gateway and sensor node
Select a random number r_i $P_i = h(r_i \| \| h(PW_i))$ Sends to GW $\{P_i, ID_i, Ts_1\}$ → Checks $\| Ts_1 - T_c \| < \Delta T$ $\alpha_i = h(K_{GW\text{-}U} \| \| ID_i)$ $b_i = \alpha_i \oplus h(P_i \| \| h(PW_i))$ $c_i = h(\alpha_i \| \| h(PW_i) \| \| ID_i)$ ← SC $\{h(.), b_i, c_i, ID_i\}$ Compute $d_i = r_i \oplus h(ID_i \| \| PW_i)$ Add d_i in to SC SC $\{h(.), b_i, c_i, d_i, ID_i\}$	$P_{sj} = h(ID_{sj} \| \| h(PW_{sj}) \| \| Ts_2)$ ← Sends to GW $\{P_{sj}, ID_{sj}, Ts_2\}$ Checks $\| Ts_2 - T_c \| < \Delta T$ $P_{sj}^* = h(ID_{sj} \| \| h(PW_{sj}^*) \| \| Ts_2) =? P_{sj}$ $\beta_j = h(K_{GW\text{-}S} \| \| ID_{sj})$ $b_{sj} = \beta_j \oplus h(ID_{sj} \| \| h(PW_{sj}))$ $c_{sj} = h(\beta_j \| \| h(PW_{sj}) \| \| ID_{sj} \| \| Ts_3)$ Sends to S_j $\{b_{sj}, c_{sj}, Ts_3\}$ → Checks whether $\| Ts_3 - T_c \| < \Delta T$ $\beta_j = b_{sj} \oplus h(ID_{sj} \| \| h(PW_{sj}))$ $c_{sj}^* = h(\beta_j \| \| h(PW_{sj}) \| \| ID_{sj} \| \| Ts_3)$ And store β_j

Figure 2. Registration phase of Singh et al.'s scheme [34].

3.1.1. Registration Between User and Gateway

Identity (ID_i) and secure password (PW_i) are provided to every user. User identity and password hash value are saved in the gateway node. At first, the gateway selects a random key $K_{GW\text{-}U}$. With this key, GW can communicate with the user. The gateway further selects a different key $K_{GW\text{-}S}$. With this particular key, GW can communicate with sensor nodes. The procedure for this phase is as follows:

Step-1: User U_i selects a random number r_i and computes $P_i = h(r_i \| \| h(PW_i))$.

Step-2: User generates time stamp T_{s1} and sends $\{P_i, ID_i, T_{s1}\}$ to GW through a protected channel.

Step-3: Following the received message, the gateway verifies the legitimacy of a time stamp.

If $\| T_{S1} - T_c \| < \Delta T$ exists, then gateway calculates.

- $\alpha_i = h(K_{GW\text{-}U} \| \| ID_i)$;
- $b_i = \alpha_i \oplus h(P_i \| \| h(PW_i))$;
- $c_i = h(\alpha_i \| \| h(PW_i) \| \| ID_i)$;

Step-4: Gateway customizes SC with $\{h(.), b_i, c_i, ID_i\}$ and conveys to the user through a protected channel.

Step-5: User adds $d_i = r_i \oplus h(ID_i \| \| PW_i)$ into SC. Now SC contains $\{h(.), b_i, c_i, d_i, ID_i\}$.

3.1.2. Registration Between Sensor node and gateway

Every sensor node has an identity (ID_{sj}) and a protected password (PW_{sj}). The identity and the hash value of the password for sensor node S_j are also saved in the gateway. The phase consists of the following steps:

Step-1: Sensor node S_j computes $P_{sj} = h(ID_{sj} \| \| h(PW_{sj}) \| \| Ts_2)$ with its ID_{sj} and PW_{sj}.

Step-2: Sensor node dispatch message $\{P_{sj}, ID_{sj}, Ts_2\}$ to the gateway.

Step-3: When information is received, the gateway confirms the validation of a time stamp. If $\| Ts_2 - T_c \| < \Delta T$, then it moves ahead or else sends non-acceptance text to the sensor node.

Step-4: With secret key $K_{GW\text{-}S}$, GW calculates the following values:

- $\beta_j = h(K_{GW\text{-}S} || ID_{sj})$;
- $b_{sj} = \beta_j \oplus h(ID_{sj} || h(PW_{sj}))$;
- $c_{sj} = h(\beta_j || h(PW_{sj}) || ID_{sj} || Ts_3)$;

Step-5: GW sends $\{b_{sj}, c_{sj}, Ts_3\}$ to the sensor node through a non-private channel.

Step-6: After confirmation of obtaining the data, the sensor node verifies the legitimacy of a time stamp. If $|Ts_3 - T_c| < \Delta T$, then move ahead to the succeeding step or else deliver a non-acceptance message to GW.

Step-7: Sensor node calculates $\beta_j = b_{sj} \oplus h(ID_{sj} || h(PW_{sj}))$ and checks $c_{sj}{}^* = h(\beta_j || h(PW_{sj}) || ID_{sj} || Ts_3)$ is equal to c_{sj}; after that, saves β_j into its memory or else sends a failure message to GW.

3.2. Login Phase

After the registration phase, the connection is established between the user and S_j via the GW node. Figure 3 describes the work-flow of the login phase. The steps are as follows:

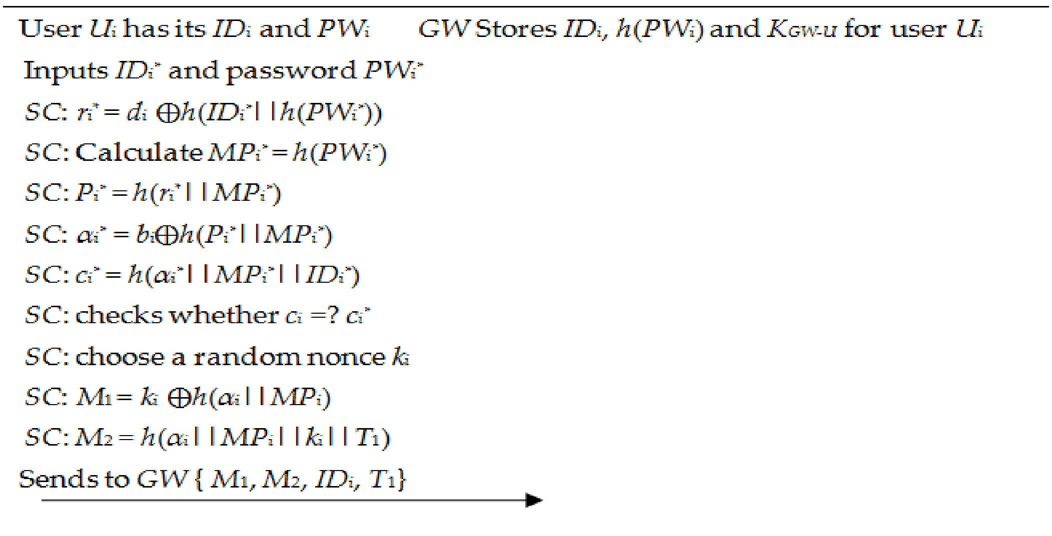

Figure 3. Login phase of Singh et al.'s scheme [34].

Step-1: User U_i inserts his/her card into the insertion area and enters his/her $ID_i{}^*$ and password $PW_i{}^*$.

Step-2: SC calculates $r_i{}^* = d_i \oplus h(ID_i{}^* || PW_i{}^*)$ with the saved value of d_i. Then it calculates $MP_i{}^* = h(PW_i{}^*)$ and $P_i = h(r_i{}^* || MP_i{}^*)$.

Step-3: Then, smartcard calculates $\alpha_i{}^* = b_i \oplus h(P_i || MP_i{}^*)$.

Step-4: SC calculates one more time $c_i{}^* = h(\alpha_i{}^* || MP_i{}^* || ID_i{}^*)$ and verifies whether the original c_i or computed $c_i{}^*$ are the same. If it is not equal, then the login progress will be terminated.

Step-5: If the entered password is exactly the same, the user selects an arbitrary number k_i and calculates $M_1 = k_i \oplus h(\alpha_i || MP_i)$ and $M_2 = h(\alpha_i || MP_i || k_i || T_1)$.

Step-6: User sends $\{M_1, M_2, ID_i, T_1\}$ to GW through an open channel.

3.3. Authentication and Key Agreement Phase

Mutual confirmation among all groups is made after the success of the login phase. This procedure is performed in the authentication and key agreement phase. It takes three steps. The first one is for the user's authority confirmation through GW. The second one represents the GW's lawfulness confirmation by the user and the sensor node. Moreover, the third one is for the user to verify the authentication of the sensor node. The focus of this phase is providing a session key between the user and the sensor node. This phase is illustrated in Figure 4. The whole authentication and key agreement phase is discussed in the following steps.

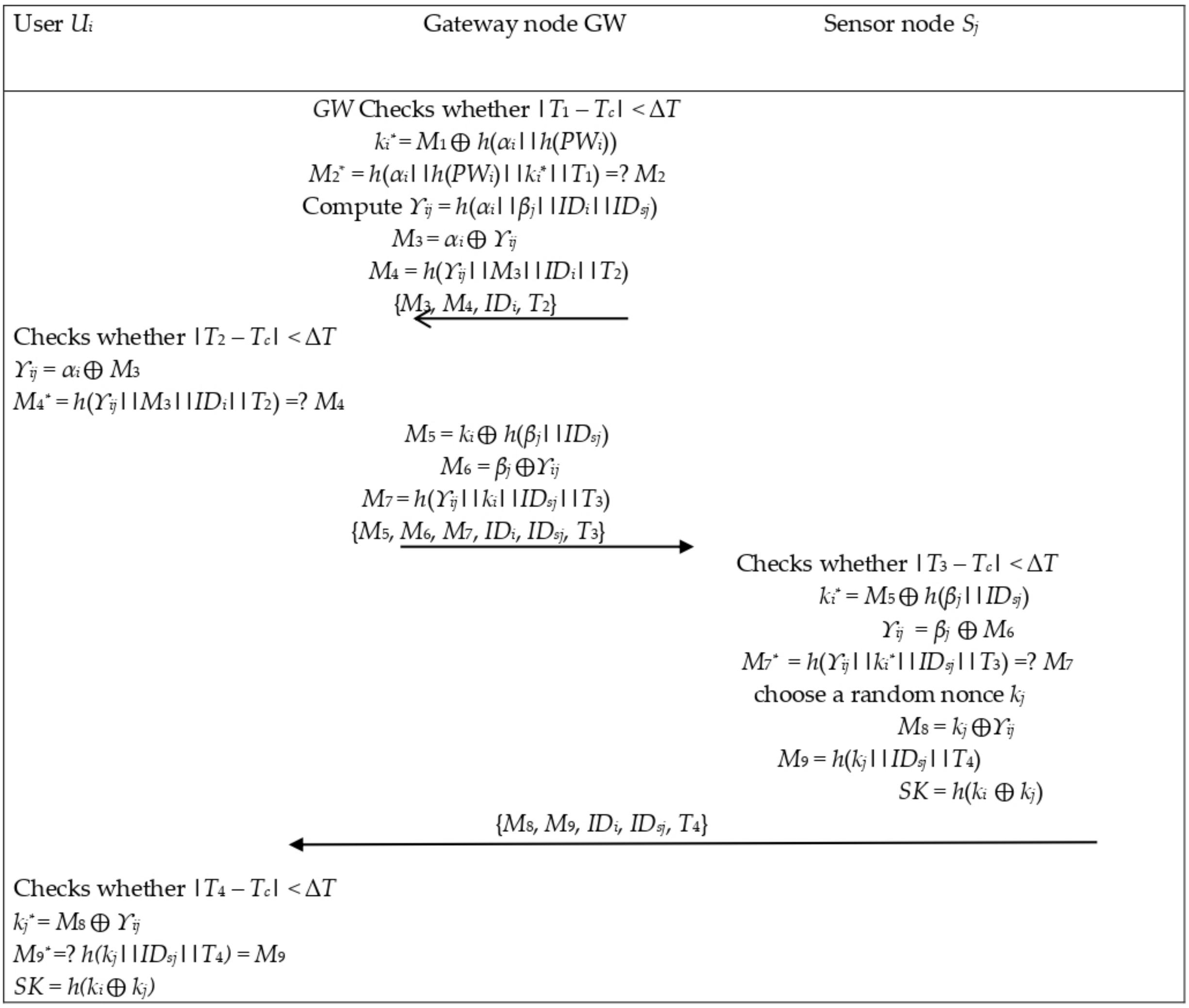

Figure 4. Authentication phase of Singh et al.'s scheme [34].

Step-1: As the gateway obtains a message $\{M_1, M_2, ID_i, T_1\}$ from the user U_i, the gateway verifies the time stamp's validity by calculating $|T_1 - T_c| < \Delta T$. If it is found valid, GW again calculates the upcoming step or else sends a failure message to U_i.

Step-2: With the help of $h(PW_i)$, as per the accepted ID_i, the gateway calculates $k_i^* = M_1 \oplus h(\alpha_i || h(PW_i))$ and after that calculates its own version of $M_2^* = h(\alpha_i || h(PW_i) || k_i^* || T_1)$ and compares it with the received M_2. In case these are the same, then GW validates the user U_i or else sends a failure text to the user.

Step-3: Once the validation of the user is completed, GW calculates $\gamma_{ij} = h(\alpha_i || \beta_j || ID_i || ID_{sj})$, $M_3 = \alpha_i \oplus \gamma_{ij}$, and $M_4 = h(\gamma_{ij} || M_3 || ID_i || T_2)$ and sends $\{M_3, M_4, ID_i, T_2\}$ to the user; here, T_2 represents GW's time stamp.

Step-4: After obtaining $\{M_3, M_4, ID_i, T_2\}$, the user verifies whether $|T_2 - T_c| < \Delta T$ and after that calculates its own version of $\gamma_{ij} = \alpha_i \oplus M_3$ and $M_4^* = h(\gamma_{ij} || M_3 || ID_i || T_2)$. The user checks whether $M_4 =? M_4^*$. If both are the same, then gateway authorization by the user U_i holds. If not, the user discontinues the procedure by sending a failure message to GW.

Step-5: At time T_2 when message is sent to user U_i, GW calculates $M_5 = k_i \oplus h(\beta_j || ID_{sj})$, $M_6 = \beta_j \oplus \Upsilon_{ij}$, and $M_7 = h(\Upsilon_{ij} || k_i || ID_{sj} || T_3)$ after that forwards $\{M_5, M_6, M_7, ID_i, ID_{sj}, T_3\}$ to sensor node S_j.

Step-6: When a message is received from GW, now S_j verifies if $|T_3 - T_c| < \Delta T$ then further calculates owned version of $k_i^* = M_5 \oplus h(\beta_j || ID_{sj})$ by using saved β_j and after this calculates its own version of $\gamma_{ij} = \beta_j \oplus M_6$ and $M_7^* = h(\gamma_{ij} || k_i^* || ID_{sj} || T_3)$ and compares M_7^* with M_7. It checks the values; if both are equal, then GW is verified by S_j, or else S_j transmits a failure text to GW.

Step-7: When authentication of GW is complete, S_j chooses a random number k_j and calculates the session key, which is $SK = h(k_i \oplus k_j)$.

Step-8: In the end, thesensor node S_j calculates $M_8 = k_j \oplus \gamma_{ij}$ and $M_9 = h(k_j || ID_{sj} || T_4)$ and transmits $\{M_8, M_9, ID_i, ID_{sj}, T_4\}$ to user U_i.

Step-9: When the text is received from sensor node S_j, the user verifies the legality of the time stamp $|T_4 - T_c| < \Delta T$a and verifies the validity of S_j by calculating its own version of $k_j = M_8 \oplus \gamma_{ij}$ and $M_9^* = h(k_j || ID_{sj} || T_4)$ and after that analyzes M_9^* with the accepted M_9. It checks if both are the same and furthermore calculates the session key as $SK = h(k_i \oplus k_j)$, then, as a result, efficiently ends the authentication phase.

4. Cryptanalysis of Singh et al.'s Scheme

4.1. Insider Attack

Suppose an insider at GW can obtain a user smart card and access the information saved in SC $\{h(.), b_i, c_i, d_i, ID_i\}$. In the registration phase, when U_i submits $\{P_i, ID_i\}$, the insider guesses the password PW_i and finds r_i in the following way:

$$r_i = d_i \oplus h(ID_i || h(PW_i))$$

after that calculates $P_i^{\#} = h(r_i || h(PW_i))$ and checks whether $P_i = ?\ P_i^{\#}$

The insider guesses the password till he/she achieves the correct password.

4.2. Offline Password Guessing Attack

Secret parameters saved into smart card are $\{h(.), b_i, c_i, d_i, ID_i\}$

An adversary U_a can do guesswork PW_i^* for the password, and now computes $r_i^{\#} = d_i \oplus h(ID_i || PW_i^{\#})$

Then, the adversary finds the value of $P_i^{\#}$ from $P_i^{\#} = h(r_i^{\#} || h(PW_i^{\#}))$. The adversary computes the value

$\alpha_i^{\#} = b_i \oplus h(P_i^{\#} || h(PW_i^{\#}))$. Then, the adversary computes $c_i^{\#} = h(\alpha_i^{\#} || h(PW_i^{\#}) || ID_i)$ and checks whether $c_i^{\#} =?\ c_i$. If it holds, the adversary obtains an exact password PW_i. In any other case, the adversary repeats the process.

4.3. Lack of User Anonymity

In Singh et al.'s scheme, messages $\{M_1, M_2, ID_i, T_1\}$, $\{M_3, M_4, ID_i, T_2\}$, and $\{M_8, M_9, ID_i, ID_{sj}, T_4\}$ directly involve the identity ID_i of a valid user U_i in plain text. By spy monitoring the messages, an adversary recognizes ID_i. Subsequently, Singh et al.'s scheme does not hold the user anonymity property.

4.4. Man-In-The-Middle Attack

During the attack, an adversary U_a tries to know the actual session key.

1. When the user U_i transmits the login message $\{M_1, M_2, ID_i, T_1\}$ to GW via a pubic channel, the adversary U_a intercepts the message and plunders the smart card, then U_a can guess the secret keywords and find the value of α_i. U_a finds $k_i = M_1 \oplus h(\alpha_i || MP_i)$. Let U_a select random nonce $k_i^{\#}$ then modify the parameter M_1 and M_2 as $M_1^{\#} = k_i^{\#} \oplus h(\alpha_i || MP_i)$ and $M_2^{\#} = h(\alpha_i || MP_i || k_i^{\#} || T_1^{\#})$. After that, U_a sends the modified message $\{M_1^{\#}, M_2^{\#}, ID_i, T_1^{\#}\}$ to GW.
2. By gateway, after receiving the message $\{M_1^{\#}, M_2^{\#}, ID_i, T_1^{\#}\}$, the gateway examines the legality of the time stamp by figuring out $|T_1^{\#} - T_c| < \Delta T$. If the legality stays, then there are further attempts to figure out the subsequent steps; if not, a rejection message drops to the user U_i.

3. The gateway computes $k_i^{\#*} = M_1^{\#} \oplus h(\alpha_i || h(PW_i))$ and then computes $M_2^{*} = h(\alpha_i || h(PW_i) || k_i^{\#*} || T_1^{\#})$ and checks whether $M_2^{*} = ?\ M_2^{\#}$. If it holds, then the gateway authenticates the user U_i; if not, it sends a rejection message to the user.
4. GW computes $\gamma_{ij} = h(\alpha_i || \beta_j || ID_i || ID_{sj})$, $M_3 = \alpha_i \oplus \gamma_{ij}$, and $M_4 = h(\gamma_{ij} || M_3 || ID_i || T_2)$ and sends $\{M_3, M_4, ID_i, T_2\}$ to the user.
5. Adversary U_a intercepts the message $\{M_3, M_4, ID_i, T_2\}$ and computes the value $\gamma_{ij} = M_3 \oplus \alpha_i$ and changes the gateway's time stamp and parameter M_4 as $M_4^{\#}$. Now U_a delivers $\{M_3, M_4^{\#}, ID_i, T_2^{\#}\}$ to the user U_i. After receiving $\{M_3, M_4^{\#}, ID_i, T_2^{\#}\}$, the user checks whether $|T_2^{\#} - T_c| < \Delta T$ and then computes $\gamma_{ij} = \alpha_i \oplus M_3$ and $M_4^{*} = h(\gamma_{ij} || M_3 || ID_i || T_2^{\#})$ and checks whether $M_4^{*} =? M_4^{\#}$. If it holds, then GW verification by the user holds; otherwise, abort the process.
6. When a message is sent at time T_2 to the user U_i, GW immediately computes $M_5 = k_i^{\#} \oplus h(\beta_j || ID_{sj})$, $M_6 = \beta_j \oplus \gamma_{ij}$, and $M_7 = h(\gamma_{ij} || k_i^{\#} || ID_{sj} || T_3)$, then sends $\{M_5, M_6, M_7, ID_i, ID_{sj}, T_3\}$ to S_j.
7. The adversary U_a intercepts the message $\{M_5, M_6, M_7, ID_i, ID_{sj}, T_3\}$. U_a changes the time stamp and parameter as $M_7^{\#} = h(\gamma_{ij} || k_i^{\#} || ID_{sj} || T_3^{\#})$. Now the adversary U_a sends the message $\{M_5, M_6, M_7^{\#}, ID_i, ID_{sj}, T_3^{\#}\}$ to S_j.
8. When a message is received from the gateway, S_j confirms whether $|T_3^{\#} - T_c| < \Delta T$ and then computes $k_i^{\#} = M_5 \oplus h(\beta_j || ID_{sj})$, $\gamma_{ij} = \beta_j \oplus M_6$, and $M_7^{*} = h(\gamma_{ij} || k_i^{\#} || ID_{Sj} || T_3)$ and checks whether $M_7^{*} =? M_7^{\#}$. If it holds, then the gateway is certified through the sensor node; if not, the sensor node sends a failure text to the gateway.
9. Once the gateway verification is completed, S_j sensor node picks a random number k_j and calculates the session key as $SK = h(k_i^{\#} \oplus k_j)$.
10. S_j computes $M_8 = k_j \oplus \gamma_{ij}$ and $M_9 = h(k_j || ID_{Sj} || T_4)$ then transmits $\{M_8, M_9, ID_i, ID_{sj}, T_4\}$ to the user U_i.
11. The adversary intercepts the message $\{M_8, M_9, ID_i, ID_{sj}, T_4\}$. U_a computes $k_j = M_8 \oplus \gamma_{ij}$, $M_9^{*} = h(k_j || ID_{Sj} || T_4)$ and checks whether $M_9 = ?\ M_9^{*}$. The adversary U_a computes the session key $SK = h(k_i^{\#} \oplus k_j)$. Now U_a chooses random number $k_j^{\#}$ and computes $M_8^{\#} = k_j^{\#} \oplus \gamma_{ij}$ and $M_9^{\#} = h(k_j^{\#} || ID_{Sj} || T_4^{\#})$. U_a transmits the message $\{M_8^{\#}, M_9^{\#}, ID_i, ID_{sj}, T_4^{\#}\}$ to the user U_i.
12. Once the message is received from sensor node S_j, the user confirms the legality of the stamp $|T_4^{\#} - T_c| < \Delta T$. The user examines the effectiveness of the sensor node by figuring out its own version of $k_j^{\#*} = M_8^{\#} \oplus \gamma_{ij}$ and $M_9^{\#*} = h(k_j^{\#*} || ID_{Sj} || T_4^{\#})$ and confirms whether $M_9^{\#} = ?\ M_9^{\#*}$. If it holds, then it calculates the session key as $SK = h(k_i \oplus k_j^{\#})$.

Two session keys are established here: one is between the user and adversary $SK = h(k_i \oplus k_j^{\#})$. The second is $SK = h(k_i^{\#} \oplus k_j)$, which is between the sensor node and the adversary. The adversary makes a fool of both the user and the sensor node by behaving like a middleman.

5. Proposed Scheme

Here, we propose an enhanced user authentication and key agreement scheme for WSNs tailored for IoT. This protocol is divided into four phases: registration, login, authentication and key agreement, and password change. Our scheme sorts out all the identified failures of Singh et al.'s scheme. The architecture of the sensors-enabled IoT network is shown in Figure 5. It depicts that the gateway node facilitates the establishment of a secure communication channel between the user and the sensor node.

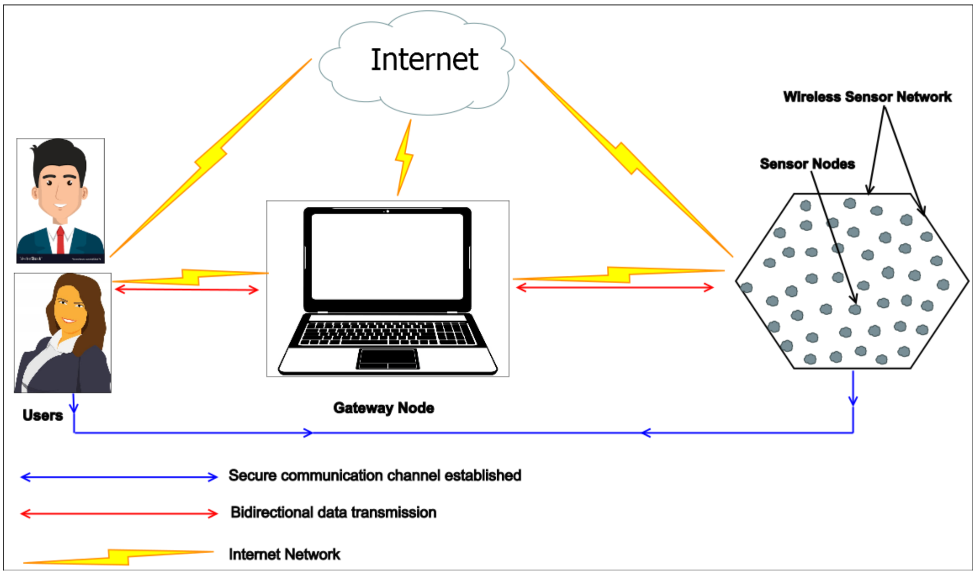

Figure 5. Sensors Enabled IoT Network.

5.1. Registration

Here we split the phase into two sub-phases.

5.1.1. Sensor Registration

Each sensor node S_j has its identity ID_{sj}. This section is performed by the GW offline before the use of sensor nodes in the target area. It contains the following steps:

- For each sensor node S_j, the GW chooses an uncommon identity ID_{sj};
- The gateway node computes a common secret key between GW and S_j

$$K_{GW\text{-}Sj} = h(ID_{sj} \,||\, K_{GW})$$

Ultimately, every sensor node S_j which is used in the target area is preloaded with the information $\{ID_{sj}, K_{GW\text{-}Sj}\}$, and GW also stores ID_{sj} in its database. This phase is shown in Figure 6.

GW
Allot ID_{sj} as identity of S_j
Secret key shared between GW and S_j is $K_{GW\text{-}Sj} = h(ID_{sj} \,||\, K_{GW})$
S_j is preloaded with the information $\{ID_{sj}, K_{GW\text{-}Sj}\}$.
S_j is deployed in target field.
GW also stores ID_{sj} in its database.

Figure 6. Sensor node pre-deployment phase.

5.1.2. User Registration

In this section, a lawful user U_i wishes to register with the GW. As a way to register to the GW, the user U_i wishes to execute the steps which are given below and shown in Figure 7.

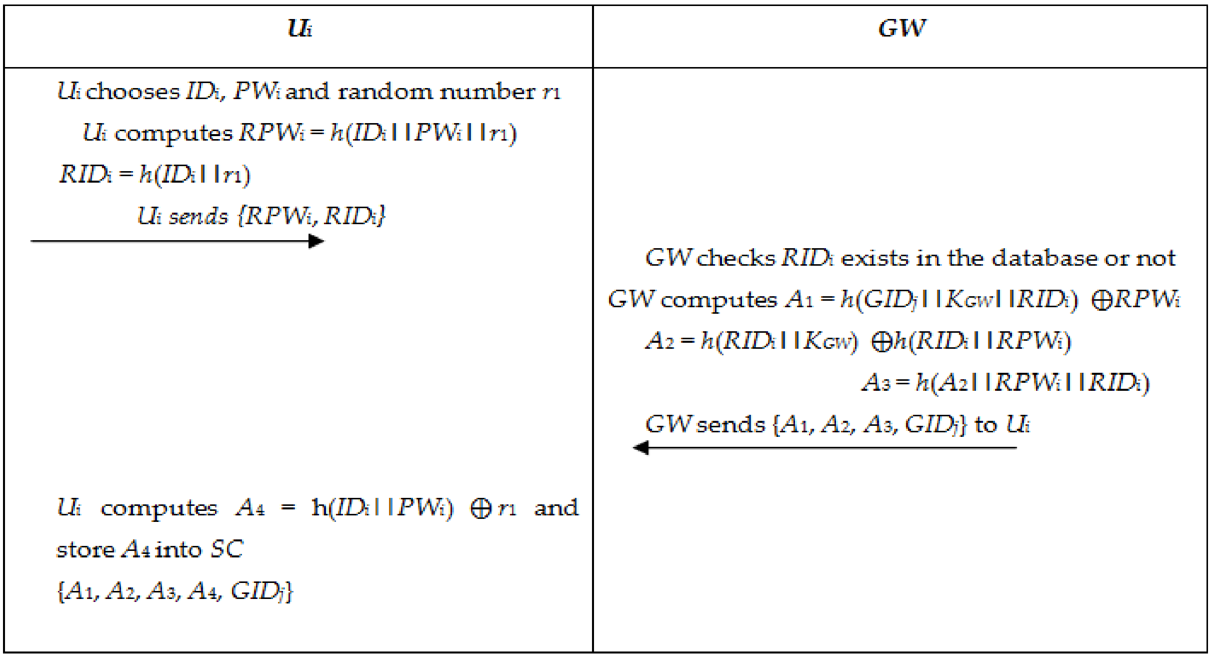

Figure 7. Registration phase between U_i and GW.

Step-1: User U_i selects ID_i, PW_i, and random number r_1. U_i calculates

$$RPW_i = h(ID_i \,||\, PW_i \,||\, r_1)$$

$$RID_i = h(ID_i \,||\, r_1)$$

Now U_i forwards the registration request message $\{RPW_i, RID_i\}$ to GW via a safe channel.

Step-2: GW investigates whether RID_i exists in the database. If it exists, then GW forwards a rejection notification to U_i. If not, GW saves RID_i in the database and computes.

$$A_1 = h(GID_j \,||\, K_{GW} \,||\, RID_i) \oplus RPW_i$$

$$A_2 = h(RID_i \,||\, K_{GW}) \oplus h(RID_i \,||\, RPW_i)$$

$$A_3 = h(A_2 \,||\, RPW_i \,||\, RID_i)$$

GW stores $\{A_1, A_2, A_3, GID_j\}$ into SC and sends SC to U_i by a private channel.

Step-3: U_i computes $A_4 = h(ID_i \,||\, PW_i) \oplus r_1$ and stores A_4 into SC $\{A_1, A_2, A_3, A_4, \mathrm{GID_j}\}$.

5.2. Login Phase

Subsequent to the completion of the registration phase, the user can contact a sensor node by the GW. Comprehensive steps are given underneath.

Step-1: User U_i enters its smart card into the terminal and loads ID_i and PW_i.

Step-2: SC computes $r_1 = A_4 \oplus h(ID_i \,||\, PW_i)$;

And $\mathrm{RPW_i} = h(\mathrm{ID_i} \,||\, \mathrm{PW_i} \,||\, \mathrm{r_1})$, $\mathrm{RID_i} = h(\mathrm{ID_i} \,||\, \mathrm{r_1})$;

And checks $A_3 = ?\ h(A_2 \,||\, RPW_i \,||\, RID_i)$;

Step-3: If they do not match, then the login process will be canceled.

Step-4: If the password entered by the user was correct, then it selects the random number r_u and required sensor ID_{sj} and computes

$$B_1 = A_1 \oplus RPW_i = h(GID_j \,||\, K_{GW} \,||\, RID_i)$$

$$B_2 = B_1 \oplus r_u$$

$$B_3 = h(GID_j \,||\, ID_{sj} \,||\, B_1 \,||\, RID_i \,||\, r_u \,||\, T_1)$$

Finally, the message $M_1 = \{B_2, B_3, GID_j, RID_i, ID_{sj}, T_1\}$ is sent to GW, where T_1 is an ongoing time stamp.

5.3. Authentication and Key Agreement Phase

Subsequent to accepting the login request message by the GW from U_i, subsequent steps are accomplished for mutual authentication and key establishment. The login and authentication phases are shown in Figure 8.

U_i	GW	S_j
U_i inputs ID_i^* and PW_i^* SC calculates $r_1^* = A_4 \oplus h(ID_i \| \| PW_i)$ And $RPW_i^* = h(ID_i^* \| \| PW_i^* \| \| r_1^*)$, $RID_i^* = h(ID_i^* \| \| r_1^*)$ And checks $A_3^* = h(A_2 \| \| RPW_i^* \| \| RID_i^*) =? A_3$ SC selects random number r_u and computes $B_1 = A_1 \oplus RPW_i = h(GID_j \| \| K_{GW} \| \| RID_i)$, $B_2 = B_1 \oplus r_u$ $B_3 = h(GID_j \| \| ID_{sj} \| \| B_1 \| \| RID_i \| \| r_u)$ $M_1 = \{B_2, B_3, GID_j, RID_i, ID_{sj}, T_1\}$ sends to GW →		
	GW checks GID_j is right and verifies $\|T_1 - T_c\| < \Delta T$ GW computes $B_1 = h(GID_j \| \| K_{GW} \| \| RID_i)$, $r_u^* = B_1 \oplus B_2$ $B_3 = h(GID_j \| \| ID_{sj} \| \| B_1 \| \| RID_i \| \| r_u^*) =? B_3^*$ GW choose random number r_g and computes $K_{GW\text{-}Sj} = h(ID_{sj} \| \| K_{GW})$ $B_4 = h(K_{GW\text{-}Sj} \| \| ID_{sj} \| \| GID_j) \oplus r_u$, $B_5 = h(r_u) \oplus r_g$ $B_6 = h(K_{GW\text{-}Sj} \| \| r_u \| \| r_g)$ $M_2 = \{ID_{sj}, B_4, B_5, B_6, T_2\}$ sends to S_j →	
		S_j checks ID_{sj} is correct And verifies $\|T_2 - T_c\| < \Delta T$ S_j calculates $r_u^* = B_4 \oplus h(K_{GW\text{-}Sj} \| \| ID_{sj} \| \| GID_j)$ And $r_g^* = B_5 \oplus h(r_u)$ And verifies $B_6^* = h(K_{GW\text{-}Sj} \| \| r_u^* \| \| r_g^*) =? B_6$ S_j selects random number r_s and computes $SK_s = h(r_u \| \| r_g \| \| r_s)$, $B_7 = h(K_{GW\text{-}Sj} \| \| r_g) \oplus r_s$ $B_8 = ID_{sj} \oplus h(r_s \| \| B_7)$, $B_9 = h(SK_s \| \| ID_{sj} \| \| GID_j \| \| r_s)$ ← $M_3 = \{B_7, B_8, B_9, T_3\}$ sends to GW
	GW verifies $\|T_3 - T_c\| < \Delta T$ And computes $r_s^* = B_7 \oplus h(K_{GW\text{-}Sj} \| \| r_g)$ $ID_{sj}^* = B_8 \oplus h(r_s^* \| \| B_7)$ GW checks ID_{sj}^* is in the database or not GW computes $SK_g = h(r_u \| \| r_g \| \| r_s^*)$ And checks $B_9 =? h(SK_g \| \| ID_{sj}^* \| \| GID_j \| \| r_s^*)$ $B_{10} = h(r_u \| \| RID_i) \oplus r_g$, $B_{11} = h(r_u \| \| r_g) \oplus r_s$ $B_{12} = h(SK_g \| \| RID_i \| \| r_g \| \| r_s)$ ← $M_4 = \{B_{10}, B_{11}, B_{12}, T_4\}$	
U_i verifies $\|T_4 - T_c\| < \Delta T$ U_i computes $r_g^* = B_{10} \oplus h(r_u \| \| RID_i)$ $r_s^* = B_{11} \oplus h(r_u \| \| r_g^*)$ $SK_u^* = h(r_u \| \| r_g^* \| \| r_s^*)$ And checks $B_{12} = ?h(SK_u \| \| RID_i \| \| r_g \| \| r_s)$ Hence, session key $SK_u = h(r_u \| \| r_g \| \| r_s)$		

Figure 8. Authentication and key agreement phase.

Step-1: Firstly, GW checks if GID_j is right. After that, GW verifies the validity of the timestamp. If $|T_1 - T_c| < \Delta T$ holds, then GW proceeds to further steps; otherwise, abort the process. GW computes

$$B_1 = h(GID_j || K_{GW} || RID_i)$$

$$r_u{}^* = B_1 \oplus B_2$$

Then checks $B_3{}^* = h(GID_j || ID_{sj} || B_1 || RID_i || r_u{}^* || T_1) = ?\ B_3$

If this does not hold, the user account RID_i will be locked. Otherwise, GW searches for ID_{sj} from the database, chooses a random number r_g, and calculates

$$K_{GW\text{-}Sj} = h(ID_{sj} || K_{GW})$$

$$B_4 = h(K_{GW\text{-}Sj} || ID_{sj} || GID_j) \oplus r_u$$

$$B_5 = h(r_u) \oplus r_g$$

$$B_6 = h(K_{GW\text{-}Sj} || r_u || r_g || T_2)$$

GW sends the message $M_2 = \{ID_{sj}, B_4, B_5, B_6, T_2\}$ to S_j, where T_2 is GW ongoing time stamp.

Step-2: Subsequent to accepting the message, S_j first checks if ID_{sj} is correct; after that, S_j verifies the legality of the time stamp. If $|T_2 - T_c| < \Delta T$ holds, then S_j proceeds to further steps; otherwise, it sends a rejection message to GW.

S_j calculates $r_u{}^* = B_4 \oplus h(K_{GW\text{-}Sj} || ID_{sj} || GID_j)$

and $r_g{}^* = B_5 \oplus h(r_u)$

and verifies $B_6{}^* = h(K_{GW\text{-}Sj} || r_u{}^* || r_g{}^* || T_2) = ?\ B_6$

If the equation is right, S_j selects r_s and computes

$$SK_s = h(r_u || r_g || r_s)$$

$$B_7 = h(K_{GW\text{-}Sj} || r_g) \oplus r_s$$

$$B_8 = ID_{sj} \oplus h(r_s || B_7)$$

$$B_9 = h(SKs || ID_{sj} || GID_j || r_s || T_3)$$

Now S_j sends $M_3 = \{B_7, B_8, B_9, T_3\}$ to GW

Step-3: Subsequent to accepting the message, GW verifies the legality of the time stamp. If $|T_3 - T_c| < \Delta T$ holds, then GW goes ahead to further steps; if not, abort the process.

GW computes $r_s{}^* = B_7 \oplus h(K_{GW\text{-}Sj} || r_g)$

$$ID_{sj}{}^* = B_8 \oplus h(r_s{}^* || B_7)$$

Then investigates $ID_{sj}{}^*$ in the database. If it does not occur, then GW stops the process; otherwise, GW checks $B_9{}^* = h(SK_g || ID_{sj}{}^* || GID_j || r_s{}^* || T_3) =?\ B_9$

GW computes $SK_g = h(r_u || r_g || r_s)$

$$B_{10} = h(r_u || RID_i) \oplus r_g$$

$$B_{11} = h(r_u || r_g) \oplus r_s$$

$$B_{12} = h(SK_g || RID_i || r_g || r_s || T_4)$$

GW sends the message $M_4 = \{B_{10}, B_{11}, B_{12}, T_4\}$ to U_i.

Step-4: Subsequent to accepting the message, U_i investigates the legality of the time stamp. If $|T_4 - T_c| < \Delta T$ holds, then U_i proceeds to further steps; otherwise, it stops the process.

$$U_i \text{ computes } r_g{}^* = B_{10} \oplus h(r_u || RID_i)$$

$$r_s^{*} = B_{11} \oplus h(r_u \,||\, r_g)$$

$$SK_u^{*} = h(r_u \,||\, r_g^{*} \,||\, r_s^{*})$$

Then checks $B_{12}^{*} = h(SK_u^{*} \,||\, RID_i \,||\, r_g^{*} \,||\, r_s^{*} \,||\, T_4) = ?\ B_{12}$
Hence, Session key $SK_u = h(r_u \,||\, r_g \,||\, r_s)$

5.4. Password Change Phase

Step-1: User U_i inserts its SC into the terminal and inputs his/her ID_i and PW_i.

$$SC \text{ computes } r_1 = A_4 \oplus h(ID_i \,||\, PW_i)$$

$$RPW_i = h(ID_i \,||\, PW_i \,||\, r_1) \text{ and } RID_i = h(ID_i \,||\, r_1)$$

Chooses a random number r_u and calculates B_1, B_2, and $B_{13} = h(GID_j \,||\, B_1 \,||\, RID_i \,||\, r_u \,||\, T_5)$. Finally, it sends $M_5 = \{GID_j, RID_i, B_2, B_{13}, T_5\}$ with to GW.

Step-2: GW investigates legality of time stamp T_5 after that calculates B_1, r_u, and checks $B_{13} =?\ h(GID_j \,||\, B_1 \,||\, RID_i \,||\, r_u \,||\, T_5)$

GW computes $B_{14} = h(GID_j \,||\, K_{GW} \,||\, RID_i) \oplus h(r_u \,||\, RID_i)$

$$B_{15} = h(RID_i \,||\, GID_j \,||\, B_{14} \,||\, T_6)$$

Finally, it sends $M_6 = \{B_{14}, B_{15}\}$ to U_i.

Step-3: After receiving M_6, SC checks the validity of time stamp and checks $B_{15} = ?\ h(RID_i \,||\, GID_j \,||\, B_{14} \,||\, T_6)$. If so, U_i inputs a new password PW_i^{new}, and the SC generates an arbitrary number r_1^{new}, and calculates

$$RPW_i^{new} = h(RID_i \,||\, PW_i^{new} \,||\, r_1^{new})$$

$$A_1^{new} = B_{14} \oplus h(r_u \,||\, RID_i) \oplus h(ID_i \,||\, PW_i^{new} \,||\, r_1)$$

$$A_2^{new} = A_2 \oplus h(RID_i \,||\, RPW_i) \oplus h(RID_i \,||\, RPW_i^{new})$$

$$A_3^{new} = h(A_2^{new} \,||\, RPW_i^{new} \,||\, RID_i)$$

Finally, the SC replaces $\{A_1, A_2, A_3\}$ with $\{A_1^{new}, A_2^{new}, A_3^{new}\}$.

6. Security Analysis

Here, we discuss the security of our scheme formally as well as informally.

6.1. Informal Security Analysis

6.1.1. Insider Attack Resistance

When a user sends a registration request message $\{RPW_i, RID_i\}$ to GW, an insider of GW obtains these secret values. Moreover, the insider obtains the parameters stored in SC $\{A_1, A_2, A_3, A_4, GID_j\}$. To find the random number r_1, the adversary needs to guess ID_i^{*} and PW_i^{*} simultaneously because $A_4 = h(ID_i \,||\, PW_i) \oplus r_1$. However, the probability of guessing ID_i^{*} and PW_i^{*} simultaneously is negligible. The adversary cannot find the random number r_1 and cannot guess ID_i^{*} and PW_i^{*}. The adversary cannot verify whether $RPW_i =?\ RPW_i^{*}$ where $RPW_i = h(ID_i \,||\, PW_i \,||\, r_1)$. Hence, an insider cannot guess the user's password.

6.1.2. Offline Password Guessing Resistance

The adversary obtains the SC $\{A_1, A_2, A_3, A_4, GID_j\}$ and obtains the parameter stored in it. From $A_4 = h(ID_i \,||\, PW_i) \oplus r_1$, the adversary knows only A_4. To find the random number r_1, the adversary needs to guess ID_i^{*} and PW_i^{*} simultaneously. However, the probability of guessing ID_i^{*} and PW_i^{*} simultaneously is negligible. In other equations, $A_1 = h(GID_j \,||\, K_{GW} \,||\, RID_i) \oplus RPW_i$, $A_2 = h(RID_i \,||\, K_{GW}) \oplus h(RID_i \,||\, RPW_i)$, and $A_3 = h(A_2 \,||\, RPW_i \,||\, RID_i)$ the password is used implicitly. From these equations, if the adversary wants to guess the password PW_i then he/she needs to know ID_i and the random number r_1. In these equations, the random number is not used in any of the equations, and ID_i is used implicitly. The adversary cannot guess the password from these equations. Thus the proposed scheme is safe against offline password-guessing attacks.

6.1.3. Identity Guessing Resistance

The correct value of the user identity (ID_i) is only known to U_i, and the gateway node saves $RID_i = h(ID_i || r_1)$, in which ID_i concatenates with the random number r_1. The user does not use his/her identity for login or for authentication. In the whole scheme, the user identity is used only inside $A_4 = h(ID_i || PW_i) \oplus r_1$. It is not possible for the adversary to find the random number r_1, and it can be easily seen that the adversary needs to accurately guess the PW_i and ID_i simultaneously, but at the same time, it is not possible. Hence, our scheme does not suffer from identity-guessing attacks.

6.1.4. User Forgery Resistance

If the adversary wants to forge the user, then the adversary needs to forge $M_1 = \{B_2, B_3, GID_j, RID_i, ID_{sj}, T_1\}$; the adversary must calculate B_2, B_3. However, in the calculation of $B_2 = B_1 \oplus r_u$ and $B_3 = h(GID_j || ID_{sj} || B_1 || RID_i || r_u || T_1)$, $B_1 = h(GID_j || K_{GW} || RID_i)$ is required. In the calculation of B_1, gateway node secret key K_{GW} is required. Thus it is not desirable for an adversary to forge a user. The user U_i and our scheme are secure against user forgery attacks.

6.1.5. Sensor Capture Resistance

If the adversary captured some sensors, other than S_j, which communicate with U_i, the adversary could not forge $M_3 = \{B_7, B_8, B_9, T_3\}$ since $K_{GW\text{-}Sj}$ is used to construct $B_7 = h(K_{GW\text{-}Sj} || r_g) \oplus r_s$. The sensors are captured by the adversary and have no association with $K_{GW\text{-}Sj}$. So, even though other sensors are seized, U_a cannot execute this attack successfully.

6.1.6. Gateway Forgery Attack

To apply this attack, the adversary wants to forge M_2 or M_4, where $M_2 = \{ID_{sj}, B_4, B_5, B_6, T_2\}$ and $M_4 = \{B_{10}, B_{11}, B_{12}, T_4\}$. To forge the message M_2, adversary must calculate B_4, B_5, and B_6 where $B_4 = h(K_{GW\text{-}Sj} || ID_{sj} || GID_j) \oplus r_u$, $B_5 = h(r_u) \oplus r_g$, and $B_6 = h(K_{GW\text{-}Sj} || r_u || r_g || T_2)$. However, in the calculation of B_4 and B_6, $K_{GW\text{-}Sj}$ shared secret key between GW and sensor node is required. To forge the message M_4, he/she must calculate B_{10}, B_{11}, and B_{12} where $B_{10} = h(r_u || RID_i) \oplus r_g$, $B_{11} = h(r_u || r_g) \oplus r_s$, and $B_{12} = h(SK_g || RID_i || r_g || r_s || T_4)$. However, it is not possible to calculate B_{10}, B_{11}, and B_{12} because random numbers and session keys are required. It is not possible to forge a gateway node. Hence, the proposed scheme is safe against gateway forgery attacks.

6.1.7. De-synchronization Resistance

De-synchronization is a very big security issue in WSNs. Our scheme includes a random number mechanism to assure the originality of interchanged messages and also uses a timestamp mechanism. In each session of our scheme, random numbers r_u, r_g, and r_s are generated by U_i, GW, and S_j, respectively. Hence, our scheme is free from de-synchronization problems.

6.1.8. No Adversarial Session Key Agreement

To change the session key, the adversary needs to change any of the random numbers r_u, r_g, and r_s.

When the message $M_1 = \{B_2, B_3, GID_j, RID_i, ID_{sj}, T_1\}$ is sent to GW, if the adversary wants to agree on the session key with GW and S_j, then U_a selects a random number r_u. Now, the adversary needs to calculate B_2 and B_3. $B_2 = B_1 \oplus r_u$ and $B_1 = A_1 \oplus RPW_i = h(GID_j || K_{GW} || RID_i)$. In $B_1 = A_1 \oplus RPW_i$, U_a cannot calculate B_1 from this because, in Section 5.2, the adversary cannot guess the user's password. In $B_1 = h(GID_j || K_{GW} || RID_i)$,

K_{GW} is $GW's$ secret key which is only known by GW. The adversary cannot calculate B_1 with this. In the calculation of $B_3 = h(GID_j || ID_{sj} || B_1 || RID_i || r_u || T_1)$, he/she must know B_1 and random number r_u. As discussed above, we conclude that U_a cannot calculate B_1 and U_a also cannot calculate B_3. In message M_1, the adversary cannot make any type of changes.

When message $M_2 = \{ID_{sj}, B_4, B_5, B_6, T_2\}$ is sent to S_j, If the adversary wants to change the session key, then U_a selects a random number r_g^*. Now, the adversary needs to calculate B_5 and B_6 where $B_5 = h(r_u) \oplus r_g$. U_a does not know the random number r_u selected by U_i, then he/she cannot calculate B_5. In $B_6 = h(K_{GW\text{-}Sj} || r_u || r_g || T_2)$, $K_{GW\text{-}Sj}$ is a shared secret key between GW and S_j. The adversary cannot calculate B_6. In message M_2, the adversary cannot make any type of change.

When message $M_3 = \{B_7, B_8, B_9, T_3\}$ is sent to GW, if U_a wants to change the session key, then U_a selects a random number r_s^*. Now, the adversary needs to calculate B_7, B_8, and B_9. The adversary needs to know $K_{GW\text{-}Sj}$ and r_g to calculate $B_7 = h(K_{GW\text{-}Sj} || r_g) \oplus r_s$, but $K_{GW\text{-}Sj}$ shares the secret key only between GW and S_j, so the adversary cannot calculate B_7. To calculate $B_8 = ID_{sj} \oplus h(r_s || B_7)$, U_a needs to know B_7. Above, we conclude that U_a cannot calculate B_7 and the adversary cannot calculate B_8. U_a needs to know SKs in order to calculate $B_9 = h(SKs || ID_{sj} || GID_j || r_s || T_3)$ where $SKs = h(r_u || r_g || r_s)$. U_a does not know the random numbers r_u and r_g, and the adversary cannot calculate B_9.

Hence, our proposed scheme is safe from adversarial session key agreement.

6.1.9. Man-In-The-Middle Attack

To apply a man-in-the-middle attack, the adversary works as a middleman between the user and the sensor node. In this attack, one session key is conducted between the user and adversary, and another session key is established between the adversary and sensor node. Both the user and the sensor node believe they are communicating with each other, but in this attack, both are communicating with the adversary.

When message $M_1 = \{B_2, B_3, GID_j, RID_i, ID_{sj}, T_1\}$ is sent to GW, then the adversary intercepts it and tries to find random number r_u where $r_u = B_2 \oplus B_1$ and $B_1 = A_1 \oplus RPW_i = h(GID_j || K_{GW} || RID_i)$. The adversary does not know RPW_i and K_{GW}. As a result, he/she cannot find r_u and cannot able to apply this attack at this end.

Similarly, when the sensor node sends message $M_3 = \{B_7, B_8, B_9, T_3\}$ to GW, then the adversary needs to know the random number $r_s = B_7 \oplus h(K_{GW\text{-}Sj} || r_g)$. However, the adversary does not know $K_{GW\text{-}Sj}$ and r_g. So he/she cannot be able to find the random number r_s.

Hence, our proposed scheme is safe from a man-in-the-middle attack.

6.1.10. Stolen Smart Card Resistance

Suppose SC of the user has been lost, then all the information stored in SC obtains by an adversary. In our proposed scheme SC has the parameters $\{A_1, A_2, A_3, A_4, GID_j\}$ where $A_1 = h(GID_j || K_{GW} || RID_i) \oplus RPW_i$, $A_2 = h(RID_i || K_{GW}) \oplus h(RID_i || RPW_i)$, $A_3 = h(A_2 || RPW_i || RID_i)$, and $A_4 = h(ID_i || PW_i) \oplus r_1$. However, without knowing (ID_i, r_1), U_a cannot obtain the user's password. An adversary cannot obtain any secret information from it. Hence our proposed protocol resists stolen smart card attacks.

6.1.11. User Anonymity Provision

Our scheme protects ID_i with $h(ID_i || r_1)$. It also protects PW_i with $h(ID_i || PW_i || r_1)$. Thus in order to obtain ID_i, a random number r_1 is needed, and to obtain $U_i's$ password $U_i's$ identity and random number r_1 need to be known. Moreover, even if a stolen smart card is obtained by the adversary, U_a cannot obtain ID_i from $A_4 = h(ID_i || PW_i) \oplus r_1$ since ID_i is protected by $h(ID_i || PW_i) \oplus r_1$. The adversary cannot find the identity and password of the user. This proves that our suggested protocol provides user anonymity.

6.1.12. Mutual Authentication Provision

GW checks $B_4 = h(K_{GW\text{-}Sj} || ID_{sj} || GID_j) \oplus r_u$ to verify U_i, and $B_9 = h(SK_g || ID_{sj} || GID_j || r_s || T_3)$ to verify S_j, S_j checks ID_{Sj} and $B_6 = h(K_{GW\text{-}Sj} || r_u || r_g || T_2)$ to authenticate GW directly and U_i indirectly. U_i checks $B_{12} = h(SK_g || RID_i || r_g || r_s || T_4)$ to justify GW directly and S_j indirectly. So, either pair of parties achieves mutual authentication.

6.1.13. Password Updating/Changing Provision

Suppose a legitimate user has his/her smart card stolen. Suppose the information is acquired by the adversary who saves in SC. Suppose the adversary revealed the information which is saved in SC. To change the password, it is necessary for the adversary to know the existing password PW_i verification. Moreover, it is not possible to find the old password because the password is protected with $RPW_i = h(ID_i || PW_i || r_1)$. In this way, an adversary needs to reckon the existing password before updating another password.

6.2. Formal Security Analysis

Here, we do a formal security analysis of our scheme with the help of a random oracle model. In this section, we use the Real or Random (RoR) [35] model to prove that the proposed protocol is secure. In the RoR model, the attacker is given the right to query and uses the interactive question and answer with a random oracle to verify the security of the proposed scheme. There are two participants in the proposed protocol: Π_I^m and Π_S^n represent the m-thIoT device instance and the n-th trusted server instance respectively. In addition, for formal security analysis, we define the following query model for attacker A.

$Execute(O)$: A by executing this query, he can intercept the messages transmitted by IoT devices and trusted server servers on the public channel, where $O = \{\Pi_I^m, \Pi_S^n\}$.

$Send(O,\ M)$:By executing the query, A can send message M to O and receive a response from O.

$Hash(string)$:By executing the query, A can enter a string and return its hash value.

$Test(O)$: A flips a coin c by executing this query. If $c = 1$, A can obtain the correct session key. If $c = 0$, A can obtain a random string with the same length as the session key.

Theorem 1. *In the R.O.R model, suppose A can execute the queries of* $Execute(O)$, $Send(O,\ M)$, $Hash(string)$, *and* $Test(Z)$, *the probability P of A breaking the protocol in polynomial time is:* $Adv_A^P(\varepsilon) \le \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2Adv_A^{\Omega}(\varepsilon)$. *Here,* q_h *refers to the number of times the hash is executed,* q_p *refers to the number of times PUF is executed. Hash and PUF refer to scope space of hash function* $H(\cdot)$ *and PUF function* $PUF(\cdot)$. $Adv_A^{\Omega}(\varepsilon)$ *represents the advantage of A cracking the symmetric cipher* Ω, *for a sufficiently small number* γ, *then* $Adv_A^{\Omega}(\varepsilon) < \gamma$.

Proof. We defined five rounds of the game $GM_0 - GM_4$ to simulate the attack process of A. In the process of proving, $Succ_A^{GM_i}(\varepsilon)$ represents the probability that A can win multiple rounds of the game, $Adv_A^P(\varepsilon)$ means that A can break the advantage of protocol. The proof steps are as follows:

GM_0 : In the ROR model, GM_0 game is a real attack on the authentication key exchange protocol proposed by A, and A flips the coin c at the beginning of the game. Therefore, we obtain the following results:

$$Adv_A^P(\varepsilon) = |2Pr[Succ_A^{GM_0}(\varepsilon)] - 1|$$

GM_1: With GM_0 being different from GM_1 by executing the *Execute* query, A can intercept the messages $\{h(ID_A),\ Auth_{req},\ TS_1,\ h(ID_A, TS_1)\}$,$\{C_{A,i},\ P_{A,i},\ TS_2,\ h(h(ID_A), C_{A,i},\ P_{A,i}),\ h(K_{A,i})\}$, and $\{P_{A,i},\ TS_3,\ h(h(ID_A), TS_3, K_{A,i},\ R_{A,i+1}), (R_{A,i+1}), (R_{A,i+1})_{h(K_{A,i})}\}$ transmitted on the public channel. Then, A will perform a *Test* query to calculate the session key $h(K_{A,i})$, but the message intercepted on the public channel cannot help A calculate SK. Therefore, the probability of A winning GM_1 by eavesdropping information will not increase. So we obtain:

$$Pr[Succ_A^{GM_1}(\varepsilon)] = Pr[Succ_A^{GM_0}(\varepsilon)]$$

GM_2 : Different from GM_1, GM_2 adds $Hash$ query and $Send$ query. In the intercepted messages $\{C_{A,i}, P_{A,i}, TS_2, h(h(ID_A), C_{A,i}), h(K_{A,i})\}$ and $\{P_{A,i}, TS_3, h(h(ID_A), TS_3, R_{A,i+1}), (R_{A,i+1}), (R_{A,i+1})_{h(K_{A,i})}\}$, the parameters $h(h(ID_A), C_{A,i}), h(K_{A,i}), h(K_{A,i})$ and $\{h(h(ID_A), TS_3, K_{A,i}, R_{A,i+1})$ are based on the one-way hash function. In addition, $h(K_{A,i})$ is different in each communication; the hash function will not collide. Therefore, according to the birthday paradox [36], we can obtain

$$|Pr[Succ_A^{GM_2}(\varepsilon)] - Pr[Succ_A^{GM_1}(\varepsilon)]| \le \frac{q_h^2}{2|Hash|}$$

GM_3 :The difference between GM_3 and GM_2 is that GM_3 adds PUF query. A executes $Send$ and PUF queries. Because the physical function PUF has security attributes. Therefore, we can obtain

$$|\Pr[Succ_A^{GM_3}(\varepsilon)] - \Pr[Succ_A^{GM_2}(\varepsilon)]| \le \frac{q_P^2}{2|PUF|}$$

GM_4 :In this game, A tries to crack the encrypted message $(R_{A,i+1})_{h(K_{A,i})}$. In the security model in Section 3.2, it is defined that the attacker cannot crack the memory of the server, A cannot obtain $h(K_{A,i})$, so A cannot calculate $(R_{A,i+1})$. According to the security of Ω symmetric encryption algorithm, we can obtain

$$|Pr[Succ_A^{GM_4}(\varepsilon)] - Pr[Succ_A^{GM_3}(\varepsilon)]| \le Adv_A^{\Omega}(\varepsilon)$$

Because the probability of success and failure of A is equal, so the probability that A can guess the session key is

$$Pr[Succ_A^{GM_4}(\varepsilon)] = 1/2.$$

According to the above formula, we can obtain

$$\frac{1}{2}Adv_A^P(\varepsilon) = \left|Pr[Succ_A^{GM_0}(\varepsilon)] - \frac{1}{2}\right|$$

$$= |Pr[Succ_A^{GM_0}(\varepsilon)] - Pr[Succ_A^{GM_4}(\varepsilon)]|$$

$$\le \sum_{i=0}^{3} |Pr[Succ_A^{GM_{i+1}}(\varepsilon)] - Pr[Succ_A^{GM_i}(\varepsilon)]|$$

$$= \frac{q_h^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + Adv_A^{\Omega}(\varepsilon)$$

Therefore, the probability that A can crack the protocol is:

$$Adv_A^P(\varepsilon) \le \frac{q_h^2}{|Hash|} + \frac{q_P^2}{|PUF|} + 2Adv_A^{\Omega}(\varepsilon)$$

7. Comparisons with other Related Schemes

7.1. Comparison of Security and Functionality Features

All the schemes [15,19,21,22,27,34] which are used in comparison suffer from security problems. The scheme in [34] suffers from insider attacks, offline password-guessing attacks, user forgery attacks, and session key disclosure attacks. This scheme does not provide user anonymity. The scheme in [15] suffers from user forgery attacks and stolen smart card attacks. The scheme in [19] does not provide user anonymity. The scheme in [21] suffers from insider attacks, user forgery attacks, sensor capture resistance, gateway forgery attacks, and password-changing provision. The scheme in [22,27] suffers from an insider attacks. Our proposed scheme resists all the security attacks which are mentioned in Figure 9. Our scheme provides functional features which cannot be seen in the related schemes [15,19,21,22,27,34].

Security Properties	Singh et al.'s	Dhillon-Kalra's	He et al.'s	Ghani et al.'s	Lee et al.'s	Sadri-Asaar's	Ours
Insider attack resistance	No	Yes	Yes	No	No	No	Yes
Offline password-guessing resistance	No	Yes	Yes	Yes	Yes	Yes	Yes
Identity-guessing resistance	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User forgery resistance	No	No	Yes	No	Yes	Yes	Yes
Sensor capture resistance	Yes	Yes	Yes	No	Yes	Yes	Yes
Gateway forgery attack	Yes	Yes	Yes	No	Yes	Yes	Yes
De-synchronization resistance	Yes	Yes	Yes	Yes	Yes	Yes	Yes
No adversarial session key agreement	No	Yes	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle attack	No	Yes	Yes	Yes	Yes	Yes	Yes
Stolen smart card resistance	Yes	No	Yes	Yes	Yes	Yes	Yes
User anonymity provision	No	Yes	No	Yes	Yes	Yes	Yes
Mutual authentication provision	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Password updating/changing provision	Yes	Yes	Yes	No	Yes	Yes	Yes

Figure 9. Comparison of security and functional features [15,19,21,22,27,34].

7.2. Comparison of Computation Cost

Figure 10 defines cryptographic functions and their running time for comparison of computation cost. Figures 10 and 11 together show the comparison of the computation cost of our scheme with schemes in [15,19,21,22,27,34].

Cryptographic function	Description	Running Time (ms)
T_h	Time cost of a hash function on user/gateway node	0.03993 ms
T_E	Time cost of elliptic curve scalar-point multiplication on server/gateway node	2.5044 ms
T_{hs}	Time cost of hash function on sensor node	3 ms
T_{Es}	Time cost of elliptic curve scalar-point multiplication on sensor node	21 ms

Figure 10. Cryptographic function and their description for computation cost.

On the user side, the scheme in [19] has the highest computation cost, while the scheme in [21] has the lowest computation cost. Protocol [22] and protocol [34] has equal computation cost. Our proposed scheme and the scheme in [27] have the second-highest computation cost. At the gateway node side, the scheme in [21] has the lowest computation cost, while the scheme in [15,19,22] has the same third-lowest computation cost. Our proposed scheme has the highest computation cost from the gateway node side. On the sensor node side, the scheme in [19] has the highest computation cost, while the scheme in [21] has the lowest computation cost. Our suggested scheme computation cost is slightly greater than the scheme in [22]. It is depicted in Figure 12 that the total computation cost of our scheme is slightly greater than the total computation cost of [22]. The scheme in [19] has the highest computation cost. The scheme in [21] has the lowest computation cost.

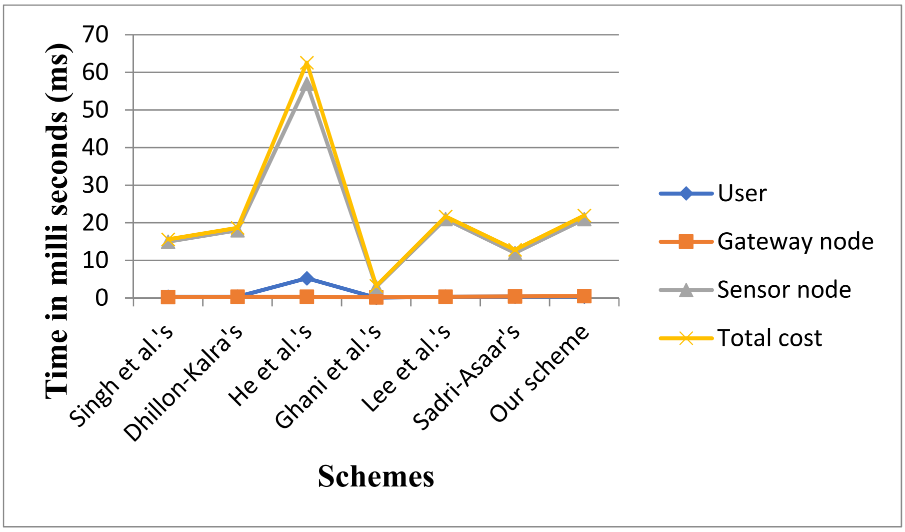

Figure 11. Comparison of computation cost.

	User	Gateway Node	Sensor Node	Total Cost
Singh et al.'s	$9T_h$ 0.35937 *ms*	$6T_h$ 0.23958 *ms*	$5T_{hs}$ 15 *ms*	$9\ T_h + 6\ T_h + 5\ T_{hs}$ 15.59895 *ms*
Dhillon–Kalra's	$8T_h$ 0.31944 *ms*	$8T_h$ 0.31944 *ms*	$6T_{hs}$ 18 *ms*	$8T_h + 8T_h + 6T_{hs}$ 18.63888 *ms*
He et al.'s	$6T_h$+$2T_E$ 5.24838 *ms*	$8T_h$ 0.31944 *ms*	$5T_{hs}$+$2T_{Es}$ 57 *ms*	$14T_h$+$2T_E$+$5T_{hs}$+$2T_{Es}$ 62.56782 *ms*
Ghani et al.'s	$2T_h$ 0.07986 *ms*	$4T_h$ 0.15972 *ms*	$1T_h$ 3 *ms*	$7T_h$ 3.23958 *ms*
Lee et al.'s	$9T_h$ 0.35937 *ms*	$8T_h$ 0.31944 *ms*	$7T_{hs}$ 21 *ms*	$17T_h$+$7T_{hs}$ 21.67881 *ms*
Sadri–Asaar's	$10T_h$ 0.3993 *ms*	$11T_h$ 0.43923 *ms*	$4T_h$ 12 *ms*	$25T_h$ 12.83853 *ms*
Our scheme	$10T_h$ 0.3993 *ms*	$13T_h$ 0.51909 *ms*	$7T_{hs}$ 21 *ms*	$23T_h$+$7T_{hs}$ 21.91839 *ms*

Figure 12. Comparison of computation cost [15,19,21,22,27].

Our proposed scheme can be a little bit more costly than other related schemes, but our scheme has passed various hurdles in security checks which makes it user-friendly. Our scheme neither uses complex cryptographic operations nor does it add much computational load when compared to its counterparts. Moreover, the running time of an operation is directly proportional to the power consumption required to run that operation. Therefore, the proposed scheme is a power-efficient protocol.

7.3. Comparison of Communication Cost

In pursuance of comparing the communication cost of the suggested protocol with the relevant protocols, we consider the length of the elliptic curve scalar-point multiplication function, and the random number is 160 bits. We suppose the length of the identities, such as ID_i and ID_{sj}, and every coordinate point from the output of the elliptic curve scalar-point multiplication function is 80 bits. Let the output of the message authentication code be 160 bits. We suppose that each element is 160 bits in the elliptic curve group. Here, we have the hash (h(.)) function SHA2-256 with the output of length 256 bits. We consider the length of the timestamp as 32 bits. In Figure 13 and in Figure 14, we show the communication costs of the three entities in our proposed scheme and the related schemes [15,19,21,22,27,34].

	User	Gateway Node	Sensor Node	Total Costs
Singh et al.'s	624 bits	1584 bits	704 bits	2912 bits
Dhillon–Kalra's	1312 bits	864 bits	2320 bits	4496 bits
He et al.'s	800 bits	800 bits	912 bits	2512 bits
Ghani et al.'s	784 bits	992 bits	368 bits	2144 bits
Lee et al.'s	704 bits	928 bits	2000 bits	3632 bits
Sadri–Asaar's	848 bits	1024 bits	512 bits	2384 bits
Our scheme	960 bits	1680 bits	800 bits	3440 bits

Figure 13. Comparison of communication cost [15,19,21,22,27,34].

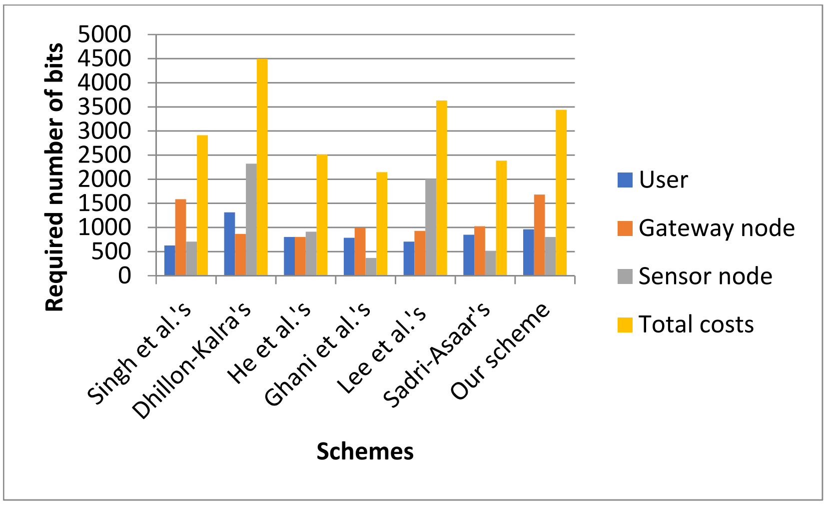

Figure 14. Comparison of communication cost.

From Figures 13 and 14, we see that, on the user side, the communication cost of the protocol [34] is 624 bits which is the minimum, and Dhillon and Kalra's scheme has the highest communication cost of 1312 bits. Our suggested scheme has the second-highest communication cost of 960 bits. At the gateway node side, our proposed scheme has the highest communication cost of 1680 bits, and the scheme in [19] has the lowest communication cost of 800 bits. At the sensor node aspect, our suggested scheme has a communication cost of 800 bits, while the protocol in [21] has the lowest communication cost of 368 bits. Dhillon and Kalra's scheme has the highest communication cost of 2320 bits. The total communication cost of Dhillon and Kalra's scheme is the highest, and in the proposed scheme, it is the third-highest. The scheme in [21] has the lowest communication cost.

8. Conclusions

We have analyzed Singh et al.'s authentication and key agreement scheme for WSNs and found some security pitfalls in it. Then we developed an improved authentication and key agreement scheme for WSNs tailored for IoT. The informal analysis of the proposed scheme indicates its resistance to various sorts of adversarial activities. The formal security of the proposed scheme with the RoR model further supports its security. In the end, we have compared the performance of our scheme with that of the related schemes. For the proposed scheme, we have tried to control the cost along with maintaining security.

Author Contributions: P.T.: conceptualization, methodology, and writing—original draft preparation. S.K.: conceptualization and supervision. B.A.A.: methodology and formal security analysis. A.G.: implementation. M.-H.Y.: methodology, informal security analysis, and final editing. All authors have read and agreed to the published version of the manuscript.

Funding: The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, has funded this project under grant no. (FP-089-43). Ming-Hour Yang is supported by MOST 111-2221-E-033-047-MOST 111-2218-E-A49-013-MBK-MOST 111-2218-E-002-038-. Saru Kumari is supported by Chaudhary Charan Singh University, Meerut Uttar Pradesh, India under "Research and Development Scheme" grant sanctioned vide Ref. No. Dev./1043 dated 29 June 2022 and by the State Government of Uttar Pradesh, India under the "Research and Development" scheme grant sanctioned vide the Government order no.-89/2022/1585/sattar-4-2022/001-4-32-2022 dated 10 November 2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors have no conflict of interest.

References

1. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090. [CrossRef]
2. Yeh, H.-L.; Chen, T.-H.; Liu, P.-C.; Kim, T.-H.; Wei, H.-W. A Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *Sensors* **2011**, *11*, 4767–4779. [CrossRef]
3. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323. [CrossRef]
4. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks* **2014**, *20*, 96–112. [CrossRef]
5. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2014**, *8*, 1070–1081. [CrossRef]
6. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf. Sci.* **2015**, *321*, 263–277. [CrossRef]
7. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Arshad, H.; Khan, M.K. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Futur. Gener. Comput. Syst.* **2016**, *63*, 56–75. [CrossRef]
8. 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. *J. Netw. Comput. Appl.* **2016**, *76*, 37–48. [CrossRef]
9. Farash, M.S.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Networks* **2016**, *36*, 152–176. [CrossRef]
10. Amin, R.; Biswas, G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80. [CrossRef]
11. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [CrossRef]
12. Chang, C.-C.; Hsueh, W.-Y.; Cheng, T.-F. A Dynamic User Authentication and Key Agreement Scheme for Heterogeneous Wireless Sensor Networks. *Wirel. Pers. Commun.* **2016**, *89*, 447–465. [CrossRef]
13. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.-K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85. [CrossRef]
14. Wu, F.; Xu, L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Netw. Appl.* **2015**, *10*, 16–30. [CrossRef]
15. Dhillon, P.K.; Kalra, S. Secure multi-factor remote user authentication scheme for Internet of Things environments. *Int. J. Commun. Syst.* **2017**, *30*, e3323. [CrossRef]
16. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Futur. Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]
17. Srinivas, J.; Mishra, D.; Mukhopadhyay, S. A Mutual Authentication Framework for Wireless Medical Sensor Networks. *J. Med Syst.* **2017**, *41*, 80. [CrossRef]
18. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.K.; Choo, K.-K.R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comput. Appl.* **2017**, *103*, 194–204. [CrossRef]
19. He, J.; Yang, Z.; Zhang, J.; Liu, W.; Liu, C. On the security of a provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718756311. [CrossRef]
20. Gupta, A.; Tripathi, M.; Shaikh, T.J.; Sharma, A. A lightweight anonymous user authentication and key establishment scheme for wearable devices. *Comput. Networks* **2018**, *149*, 29–42. [CrossRef]
21. Ghani, A.; Mansoor, K.; Mehmood, S.; Chaudhry, S.A.; Rahman, A.U.; Saqib, M.N. Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. *Int. J. Commun. Syst.* **2019**, *32*, e4139. [CrossRef]
22. Lee, H.; Kang, D.; Ryu, J.; Won, D.; Kim, H.; Lee, Y. A three-factor anonymous user authentication scheme for Internet of Things environments. *J. Inf. Secur. Appl.* **2020**, *52*, 102494. [CrossRef]

23. Mall, P.; Amin, R.; Obaidat, M.S.; Hsiao, K.-F. CoMSeC++: PUF-based secured light-weight mutual authentication protocol for Drone-enabled WSN. *Comput. Networks* **2021**, *199*, 108476. [CrossRef]
24. Chen, C.-M.; Deng, X.; Gan, W.; Chen, J.; Islam, S.K.H. A secure blockchain-based group key agreement protocol for IoT. *J. Supercomput.* **2021**, *77*, 9046–9068. [CrossRef]
25. Chen, C.-M.; Liu, S. Improved Secure and Lightweight Authentication Scheme for Next-Generation IoT Infrastructure. *Secur. Commun. Netw.* **2021**, *2021*, 1–13. [CrossRef]
26. Ali, I.; Chen, Y.; Ullah, N.; Kumar, R.; He, W. An Efficient and Provably Secure ECC-Based Conditional Privacy-Preserving Authentication for Vehicle-to-Vehicle Communication in VANETs. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1278–1291. [CrossRef]
27. Sadri, M.J.; Asaar, M.R. An efficient hash-based authentication protocol for wireless sensor networks in Internet of Things applications with forward secrecy. *Int. J. Commun. Syst.* **2021**, *34*, e4823. [CrossRef]
28. Rangwani, D.; Sadhukhan, D.; Ray, S.; Khan, M.K.; Dasgupta, M. A robust provable-secure privacy-preserving authentication protocol for Industrial Internet of Things. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 1548–1571. [CrossRef]
29. Nashwan, S. An End-to-End Authentication Scheme for Healthcare IoT Systems Using WMSN. *Comput. Mater. Contin.* **2021**, *68*, 607–642. [CrossRef]
30. Tanveer, M.; Alkhayyat, A.; Khan, A.U.; Kumar, N.; Alharbi, A.G. REAP-IIoT: Resource-Efficient Authentication Protocol for the Industrial Internet of Things. *IEEE Internet Things J.* **2022**. [CrossRef]
31. Kumar, V.; Kumar, R.; Jangirala, S.; Kumari, S.; Kumar, S.; Chen, C.-M. An Enhanced RFID-Based Authentication Protocol using PUF for Vehicular Cloud Computing. *Secur. Commun. Networks* **2022**, *2022*, 1–18. [CrossRef]
32. Wu, T.-Y.; Guo, X.; Chen, Y.-C.; Kumari, S.; Chen, C.-M. SGXAP: SGX-Based Authentication Protocol in IoV-Enabled Fog Computing. *Symmetry* **2022**, *14*, 1393. [CrossRef]
33. Li, Z.; Miao, Q.; Chaudhry, S.A.; Chen, C.-M. A provably secure and lightweight mutual authentication protocol in fog-enabled social Internet of vehicles. *Int. J. Distrib. Sens. Netw.* **2022**, *18*, 15501329221104332. [CrossRef]
34. Singh, A.; Awasthi, A.K.; Singh, K. Cryptanalysis and Improvement in User Authentication and Key Agreement Scheme for Wireless Sensor Network. *Wirel. Pers. Commun.* **2016**, *94*, 1881–1898. [CrossRef]
35. Canetti, R.; Goldreich, O.; Halevi, S. The random oracle methodology, revisited (preliminary version). In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 24–26 May 1998; pp. 209–218. [CrossRef]
36. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Advances in Cryptology—EUROCRYPT 2000*; Springer: Berlin, Heidelberg, 2000; pp. 156–171. [CrossRef]