EURASIP Journal on Information Security

RESEARCH Open Access

A multi-gateway authentication and key-agreement scheme on wireless sensor networks for IoT

Jen-Ho Yang*

Abstract

The Internet of Things (IoT) is designed to let anything connect to the Internet, and the things can be people, computers, and things. On the IoT, the Wireless Sensor Network (WSN) plays an important role because it can be used in many applications such as smart home, intelligent transportation, and intelligent disaster prevention. Since the WSN transmits data in the wireless way, the security problem is a concerning issue in this field. On the WSN, an authentication and key-agreement scheme can let the sensors authenticate to each other and share a common key to encrypt the data. Thus, it can be used to solve the security problem of WSNs. In this paper, I propose a new multi-gateway authentication and key-agreement scheme on WSN for IoT. The proposed scheme adopts a new multi-gateway structure, and thus it allows users and sensors to join in different areas of WSN dynamically. According to the performance analysis, the execution time of the proposed scheme is only $24Th$, where T_h is the execution time of a one-way hash function. In conclusion, the proposed scheme is more efficient than the related works on the WSN for IoT applications.

Keywords Wireless sensor network, IoT, Multi-gateway, Authentication, Key-agreement

1 Introduction

The Internet was initially designed to connect different computers, so that different computers could easily communicate and share data with each other. As time goes on, the Internet was developed to be the Internet of Things (IoT) in recent years [1–4]. The IoT is designed to connect different things, and the "thing" means anything that can be embedded in a chip or sensor to let it have networking or sensing capability. IoT lets anything connect to the Internet, and thus it accomplishes communications among people, computers, and things. There are many applications for the IoT. For example: smart homes, intelligent transportation systems, and smart grid systems. Basically, the IoT architecture can be divided into three layers: a perception layer, a network layer, and an application layer [5]. The three layers are briefly described as follows.

1.1 Perception layer

The perception layer uses devices and sensors that can detect, recognize, and communicate the collected surrounding information such as the temperature, humidity, and lights. And, current mobile communication devices use various kinds of sensors, the devices used in the perception layer can also include smart wearable devices which can collect human body information. Therefore, this layer can be seen as five sense organs in the human neural network.

1.2 Network layer

The network layer acts like a bridge between perception layer and application layer. It is responsible for transmitting the information collected from the physical objects through sensors. The transmission can be wired or

*Correspondence:
Jen-Ho Yang
163982@mail.tku.edu.tw
Department of Artificial Intelligence, Tamkang University, New Taipei City, Taiwan

wireless such as LoRa, ZigBee, or Bluetooth [5]. It also takes the responsibility for connecting the smart things, network devices and networks to each other. Therefore, this layer can be seen as the peripheral nerves in the neural network.

1.3 Application layer

The application layer focuses on how to apply the information collected from the perception and network layers. It defines all applications that use IoT technology or in which IoT has been deployed. The applications of IoT can be smart homes, intelligent transportation, smart grid, and smart healthcare, etc. It has the responsibility to provide the services for the different applications. Therefore, this layer can be seen as the brain in the neural network. And, the brain makes a response after organizing and interpreting the messages from the above layers.

According to the above descriptions, the Wireless Sensor Network (WSN) [6, 7] plays an important role in IoT applications. Basically, the WSN is a group of sensors deployed in different locations of an area. And, each sensor gathers data and sends it to a central location (such as a gateway or a base station) for data saving, viewing, or analyzing. However, the WSN transmits data using wireless ways (such as RFID, ZigBee, and Bluetooth) [8, 9], so the data can be easily gathered by attackers. To solve the security problem, many authentication and key-agreement schemes for the WSN have been proposed in recent years [10–15]. These schemes provide identity authentication among the sensors, users, and the gateway. In addition, the key agreement provides the encryption/decryption key used in the symmetric-key cryptosystem [16]. In these schemes, the sensors collect the surrounding data and transmit it to a gateway. Then, the gateway relays and analyzes the data between sensors and users. Besides, the gateway can be seen as a manager of the WSN, and it is responsible for the parameter settings and identity authentication for the sensors and users.

To accomplish the above goals, Amin and Biswas [17] proposed a secure light-weight scheme for user authentication and key agreement in multi-gateway WSN. In their scheme, the user accesses the data through a local gateway in the local WSN area. To access a foreign WSN in another area, the user can ask the local gateway to communicate with the foreign gateway to obtain the data. Amin and Biswas's scheme is a multi-gateway structure, and it can allow the user to access the data through different WSNs. Therefore, Amin and Biswas's scheme is suitable for IoT applications in a wide area such as smart cities. In 2021, Kwon et al. [18] proposed a secure and lightweight mutual authentication scheme for WSN. Kwon et al.'s scheme provides the mutual authentication and key agreement among the user, the gateway, and a sensor on WSN. Kwon et al. claimed that their scheme is securer and more efficient in comparison with the related schemes.

However, we find that the above two schemes have some problems in practice. In Amin and Biswas's scheme, the gateway does not authenticate the sensor's identity while a new sensor joins in the WSN. That is, an attacker can easily deploy a malicious sensor to gather the data in the WSN. This causes a serious security problem, while the data is confidential in some IoT applications. In addition, Amin and Biswas's scheme has heavy computation and communication loads to accomplish the multi-gateway structure.

On the other hand, I also found that Kwon et al.'s scheme has the following disadvantages. First, the sensors have to register at the gateway before they are deployed in the WSN. And, the gateway needs to transmit the parameters to the sensors through a secure channel. According to the above description, the registration needs to be finished before the sensors are deployed. And, it cannot be done in the wireless network environment, which is not a secure channel. In some applications, the sensors need to be deployed in WSN dynamically. However, Kwon et al.'s scheme is not suitable for these applications because the registration has to be previously performed in a secure channel. Second, Kwon et al.'s scheme is a single gateway structure in WSN. Compared with Amin and Biswas's scheme, it can be applied for a few applications. If we apply Kwon et al.'s scheme to the multi-gateway structure, then the user needs to register to a new gateway again while he wants to access to different WSNs. Therefore, Kwon et al.'s scheme is very inefficient for the multi-gateway scenario. Third, Kwon et al. did not design the steps for data transmissions on WSN after the mutual authentication and key agreement had been done. Therefore, their scheme is not a complete version of WSN.

To solve the above-mentioned problems of the related works, I propose a multi-gateway authentication and key-agreement scheme on WSN for IoT in this paper. The proposed scheme has the following advantages and novelties.

Low computation loads: Compared with related works [17–20], the proposed scheme has less computation and communication loads.

Flexibility: The proposed scheme allows users and sensors to join in different WSN's dynamically. After registering at the system administer (a mainframe of WSN) once, users can access data through gateways from different WSNs without performing the registration again.

Completeness: Unlike Kwon et al.'s scheme [18], the proposed scheme provides two additional algorithms for users to access the data from different areas of WSN based on multi-gateway environ-

ments. Therefore, the proposed scheme is more complete than Kwon et al.'s scheme for the WSN.

Power saving: Based upon the proposed performance analysis, the computation and communication costs of the proposed scheme are less than those of related works. Thus, the proposed scheme can save the sensor's electricity, and it is energy-efficient for the WSN.

Multi-gateway structure: Unlike related works [17–23], the proposed scheme is designed by a multi-gateway structure. Thus, it can be applied to many large-area WSN applications for IoT such as smart cities or intelligent disaster prevention.

According to the above reasons, the proposed scheme is more efficient and practical than the related works for IoT applications.

2 Review of Kwon et al.'s scheme

In this section, I review Kwon et al.'s scheme [18] and point out its flaws. Their scheme is divided into five phases: the sensor node registration, user registration, login and authentication, password update, and sensor node addition phases. The notations are shown in Table 1.

Table 1 The notations of Kwon et al.'s scheme

U_i	The user i
S_j	The sensor j
GW	Gateway
ID_i	The identity of user i
PW_i	The password of user i
PID_i	The pseudo identity of user i
SID_j	The identity of sensor j
k_{GWN}	A long-term secret key of the gateway
KG	A secret key between the gateway and sensor
R_l	A random number generated by the participant l
N_l	A nonce generated by the participant l
SK	The session key
$h(\cdot)$	A secure one-way hash function
$\|\|$	The concatenation operation
$\oplus$	The exclusive-or (XOR) operation

2.1 Sensor node registration phase

In this phase, a sensor node S_j sends a registration request to the gateway GW, and the GW computes secret parameters and sends them to the sensor node in a secure channel. Then, S_j stores the parameters in its storage space. The steps of this phase are described as follows Fig. 1.

Step 1. S_j selects its identity SID_j and generates a random number R_j. Then, S_j sends SID_j and $h(SID_j||R_j)$ to the GW through a secure channel.
Step 2. The GW computes $KS_j = h(h(SID_j||R_j)||k_{GWN})$ and stores SID_j and $h(SID_j||R_j)$ in its database. Then, the GW sends KS_j to the S_j.
Step 3. Finally, S_j stores KS_j in its memory.

2.2 User registration phase

Step 1. U_i inputs ID_i and PW_i in the smart card. Then, U_i transmits ID_i to the GW in a secure channel.
Step 2. GW generates two random numbers, x and R_g. Then, it computes $HID_i = h(ID_i||R_g)$ and $PID_i = HID_i \oplus h(x||k_{GWN})$. After that, GW stores PID_i and x in its database and sends $\{PID_i, HID_i, h(\cdot)\}$ to U_i.
Step 3. U_i generates R_i to compute $APW_i = h(PW_i||R_i)$, $SR_i = R_i \oplus (ID_i||PW_i)$, $SHID_i = HID_i \oplus h(PW_i||ID_i||R_i)$, and $V_i = h(APW_i||ID_i||R_i)$. Finally, U_i stores SR_i, $SHID_i$, V_i, PID_i, and $h(\cdot)$ in the smart card Fig. 2.

2.3 Login and authentication phase

Step 1. U_i inputs ID_i and PW_i into the smart card. Then, the smart card computes $R_1^* = SR_i \oplus h(ID_i||PW_i)$, $APW_i^* = h(PW_i||R_i)$, and $V_i^* = h(APW_i^* ||ID_i|| R_1^*)$. After that, the smart card checks if V_i^* is equal to V_i. If they are equal, then the smart card generates a random nonce N_1 and computes $HID_i = SHID_i \oplus h(PW_i||ID_i||R_i)$, $S_i = SID_j \oplus h(PID_i|HID_i)$, $M_1 = N_1 \oplus h(HID_i||PID_i)$, and $V_1 = h(SID_j|| PID_i|| N_1||HID_i)$. Finally, U_i sends $\{PID_i, S_i, M_1, V_1\}$to GW.

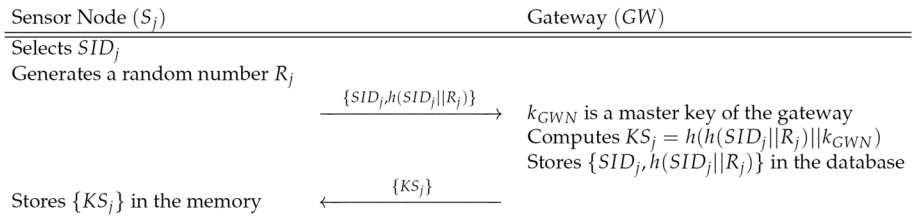

Fig. 1 Sensor node registration phase [18]

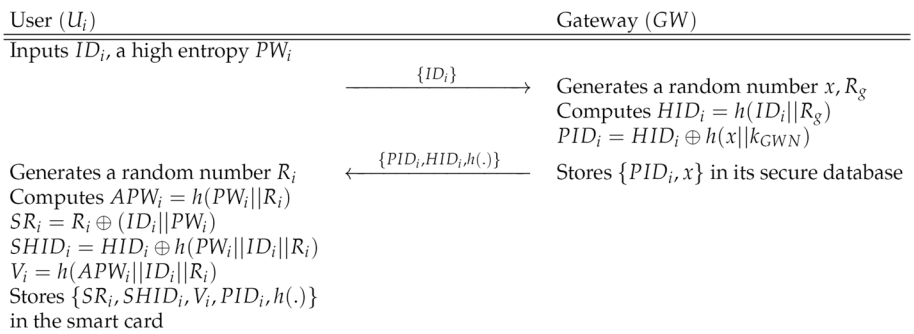

Fig. 2 User registration phase [18]

Step 2. GW computes $HID_i^* = PID_i \oplus h(x||k_{GWN})$, $SID_j^* = S_i \oplus h(PID_i||\ HID_i^*)$, $N_1^* = M_1 \oplus h(HID_i^*||PID_i)$, and $V_1^* = h(SID_j^*||PID_i||\ N_1^*\ ||\ HID_i^*)$. Then, GW checks if V_1^* is equal to V_1. If they are equal, then GW computes $KS_j = h(h(SID_j||R_j)||k_{GWN})$, $M_2 = h(N_2||HID_i) \oplus h(KS_j||PID_i)$, $M_3 = N_1 \oplus h(h(N_2||HID_i)||KS_j)$, and $V_2 = h(PID_i||SID_j||h(N_2||HID_i)||N_1)$. At last, GW sends $\{PID_i, M_2, M_3, V_2\}$to S_j.

Step 3. S_j computes $h(N_2||HID_i)^* = M_2 \oplus h(KS_j||PID_i)$, $N_1^* = M_3 \oplus h(h(N_2||HID_i)^*||PID_i)$, and $V_2^* = h(PID_i||SID_j||h(N_2||HID_i)||\ N_1^*)$. Then, S_j checks if V_2^* is equal to V_2. If they are equal, then S_j computes $SK = h(h(N_2||HID_i)||N_3||N_1)$, $M_4 = N_3 \oplus h(KS_j||N_2)$, and $V_3 = h(SK||N_3||SID_j)$. Finally, S_j sends M_4 and V_3 to GW.

Step 4. GW computes $N_3^* = M_4 \oplus h(KS_j||N_2)$, $SK^* = h(h(N_2||HID_i)||\ N_3^*\ ||N_1)$, and $V_3^* = h(SK^*||\ N_3^*\ ||SID_j)$. Then, GW checks if V_3^* is equal to V_3. If they are equal, then GW generates N_2 and computes $x^{new} = h(x||N_2)$, $PID_i^{new} = HID_i \oplus h(x^{new}||k_{GWN})$, $P_i = PID_i^{new} \oplus h(N_1||HID_i)$, $M_5 = N_2 \oplus h(HID_i||SID_j||\ N_1)$, $M_6 = N_3 \oplus h(N_2||HID_i||\ PID_i^{new})$, and $V_4 = h(N_2||N_3||\ PID_i^{new}\ ||SK)$. Finally, the GW sends $\{P_i, M_5, M_6, V_4\}$to U_i.

Step 5. U_i computes $PID_i^{new} = P_i \oplus h(N_1||HID_i)$, $N_2^* = M_5 \oplus h(HID_i||SID_j||N_1)$, $N_3^* = M_6 \oplus h(N_2^*||HID_i||\ PID_i^{new})$, $SK^* = h(h(N_2^*||HID_i)||\ N_3^*\ ||N_1)$, and $V_4^* = h(N_2^*\ ||\ N_3^*\ ||\ PID_i^{new}\ ||SK^*)$. Then, U_i checks the equality if V_4^* is equal to V_4. If they are equal, then U_i replaces PID_i to PID_i^{new} in the smart card. Otherwise, U_i considers the GW is invalid and denies the transactions Fig. 3.

2.4 Password update phase

Step 1. If U_i wants to change the password, then U_i inputs ID_i and PW_i into the smart card. The smart card computes $R_i^* = SR_i \oplus h(ID_i||PW_i)$, $APW_i^* = h(PW_i||R_i)$, and $V_i^* = h(APW_i||ID_i||\ R_i^*)$. Then, it checks the equality if V_i^* is equal to V_i. If they are equal, then the smart card accepts the request of changing password.

Step 2. U_i inputs a new password PW_i^{new} into the smart card. Then, the smart card selects a random number R_i^{new} to compute $APW_i^{new} = h(PW_i^{new}\ ||\ R_i^{new})$, $SR_i^{new} = R_i^{new} \oplus h\ (ID_i||\ PW_i^{new})$, $SHID_i^{new} = HID_i \oplus h(PW_i^{new}\ ||ID_i||\ R_i^{new})$, and $V_i^{new} = h(APW_i^{new}\ ||ID_i||\ R_i^{new})$. Finally, the smart card stores SR_i^{new}, $SHID_i^{new}$, V_i^{new}, PID_i, and $h(\cdot)$ in its memory.

2.5 Sensor node addition phase

Step 1. S_j^{new} selects its identity SID_j^{new} and a random number R_j^{new}. Then, S_j^{new} computes $h(SID_j^{new}\ ||\ R_j^{new})$ and sends $\{\ SID_j^{new}, h(SID_j^{new}\ ||\ R_j^{new})\}$ to GW through a secure channel.

Step 2. GW computes $KS_j^{new} = h(h(SID_j^{new}\ ||\ R_j^{new})||k_{GWN})$ and stores $\{\ SID_j^{new}, h(SID_j^{new}\ ||\ R_j^{new})\}$ in its database. Then, the GW sends KS_j^{new} to S_j^{new} through a secure channel.

Step 3. Finally, S_j^{new} stores KS_j^{new} in its memory.

According to the above descriptions, I think Kwon et al.'s scheme has some disadvantages as follows. First, the sensor node registration phase and user registration phase in their scheme have to be performed in a secure channel. That is, the user and sensor registrations cannot be executed in wireless network environments. However, in some WSN applications, the sensors need to be deployed dynamically. Thus, Kwon et al.'s scheme is not suitable for the dynamic network topology. Second, Kwon et al.'s scheme is not a multi-gateway WSN scheme. Compared with related work [17], it can be applied to a few applications. Third, Kwon et al. did not design the steps for data transmissions after the mutual authentication and key agreement had been finished. Thus, Kwon et al.'s scheme is not a complete version. Finally, Kwon et al.'s scheme has many redundant computations. To authenticate the user, the gateway only needs to check if the

Fig. 3 Login and authentication phase [18]

user knows the correct $PID_i = HID_i \oplus h(x||k_{GWN})$, where k_{GWN} is a long-term secret key of the gateway. However, to fulfill this purpose, the user side has to compute $R_1^* = SR_i \oplus h(ID_i||PW_i)$, $APW_i^* = h(PW_i||R_i)$, $V_i^* = h(APW_i^*||ID_i||R_1^*)$, $HID_i = SHID_i \oplus h(PW_i||ID_i||R_i)$, $S_i = SID_j \oplus h(PID_i||HID_i)$, $M_1 = N_1 \oplus h(HID_i||PID_i)$, and $V_1 = h(SID_j||PID_i||N_1||HID_i)$ in Step1 of Subsection 2.1. This is very inefficient and impractical while the user only needs to prove to the gateway that he knows the correct $PID_i = HID_i \oplus h(x||k_{GWN})$.

3 The proposed scheme

To solve the problems of the related works [17, 18], I propose a multi-gateway authentication and key-agreement scheme on WSNs for IOT in this section. The proposed multi-gateway structure is shown in Fig. 4. In the proposed structure, there are several clusters of sensors and gateways. The system administrator (SA) is responsible for setting and storing the system parameters for all participants in the proposed scheme. In addition, the

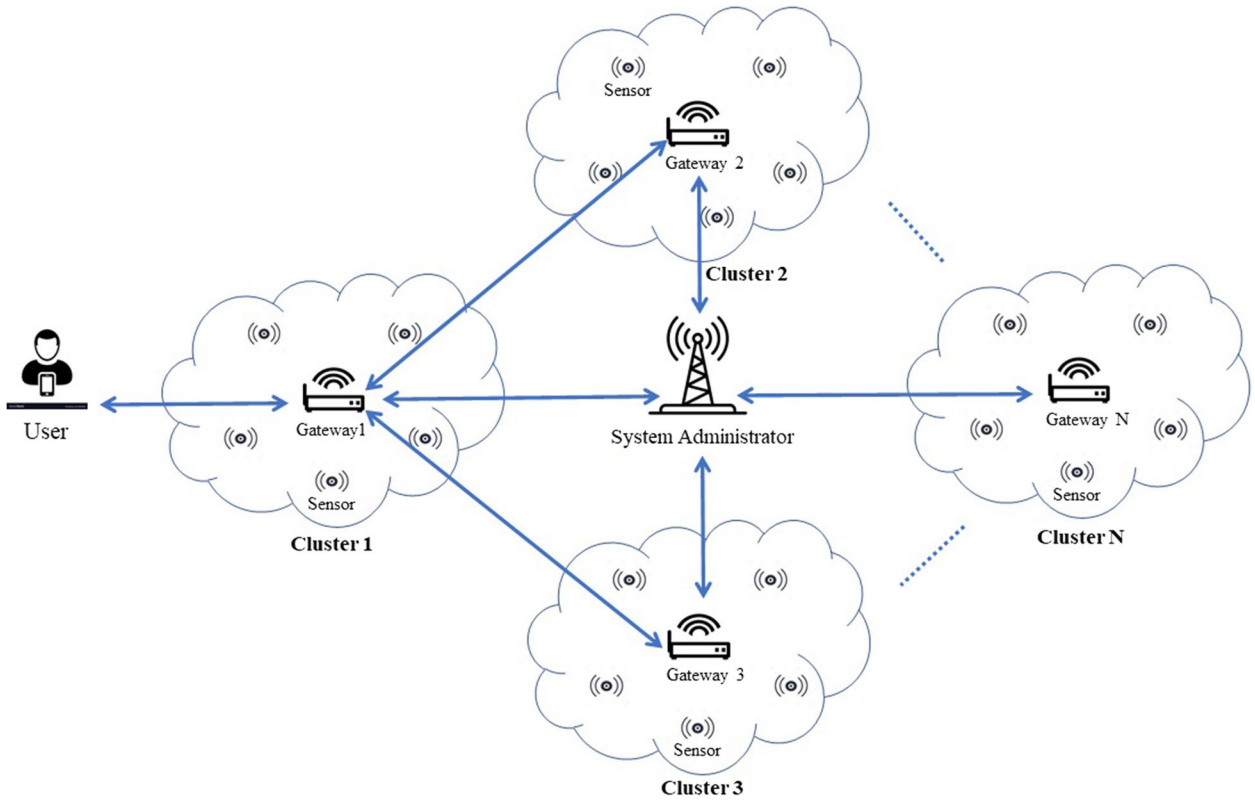

Fig. 4 The proposed multi-gateway WSN structure for IoT

SA can be seen as a mainframe or a base station that keeps the data collected by all sensors. In each cluster, the gateway (GW) is a manager of all sensors, and it collects all data gathered by the sensors. Then, the GW of each cluster sends the data collected from its area to the SA. Besides, the GW is also responsible for authenticating the validities of the user's and sensor's identities. And, the user can get the data collected by the sensors through the GW.

The proposed scheme can be divided into four phases: the system setting phase, sensor registration and authentication phase, user registration and authentication phase, and data access phase. Table 2 shows the parameters used in the proposed scheme.

3.1 System setting phase

In this phase, SA is responsible for setting the parameters of all users, sensors, and gateways. When a user wants to access the wireless sensor network (WSN), he must register with the SA. The steps are shown as follows.

Step 1. Before sensors are deployed, SA randomly selects a secret integer R_{SA} and stores $\{ID_{HG}, X_{HG}, R_{SA}\}$ in the HG. Then, SA stores $\{ID_{S_j}, R_{SA}\}$ to each S_j.

Table 2 The parameters used in the proposed scheme

SA	System administrator
GW	Gateway
HG	Home gateway
FG	Foreign gateway
U_i	The user i
S_j	The sensor j
ID_x	The identity of the participant x
PW_i	The password of user i
TID_{U_i}	The temporary pseudo identity of user i
R_x	A secret integer randomly chosen by user x
X_x	The secret key of the participant x
T_k	The timestamp with the order k
ΔT	The valid time interval of transmitting a timestamp
$h(\cdot)$	A secure one-way hash function such as MD5 or SHA-2 [16]
$\|$	The concatenation operation
$\oplus$	The exclusive-or (XOR) operation
K_{xy}	The symmetric key [16] negotiated by x and y
$E_{K_{xy}}(\bullet)$	The symmetric encryption function using K_{xy} [16]
$D_{K_{xy}}(\bullet)$	The symmetric decryption function using K_{xy} [16]
$DATA$	The data collected by sensors
$EDATA$	The encryption data
$A = ?B$	The expression of "if A is equal to B or not"

Step 2. U_i sends a registration request to the SA. Then, SA sends { ID_i, PW_i, R_{SA} } to U_i in a secure channel. Finally, U_i stores { ID_i, PW_i, R_{SA} } into its mobile device or smart card.

3.2 Sensor registration and authentication phase

In this phase, the sensors are deployed in an area, and each sensor has to register at a home gateway (HG). Then, the HG and a sensor authenticate each other's validities. Note that all steps of this phase can be performed by the HG and sensors automatically without human intervention. The Steps are shown as follows.

Step 1. S_j sends ID_{S_j} to the HG. Then, HG computes $A_{S_j} = h\left(ID_{S_j}||X_{HG}\right)$, $\overline{A_{S_j}} = A_{S_j} \oplus R_{SA}$, and $\widetilde{A_{S_j}} = h\left(A_{S_j}||R_{SA}||T_1\right)$, and it sends $\{\overline{A_{S_j}}, \widetilde{A_{S_j}}, T_1\}$ to S_j.

Step 2. S_j computes $A_{S_j}' = \overline{A_{S_j}} \oplus R_{SA}$ and checks $\widetilde{A_{S_j}} =? h\left(A_{S_j}' || R_{SA} || T_1\right)$ and if T_1 is valid or not by examining ΔT. If $\widetilde{A_{S_j}}$ is equal to $h\left(A_{S_j}' || R_{SA} || T_1\right)$ and T_1 is valid, then S_j ensures that the HG is a legal gateway and $A_{S_j} = A_{S_j}'$. After that, S_j selects R_{Sj0} to compute $\overline{B_{S_j}} = h\left(A_{S_j}\right) \oplus R_{S_j0}$ and $\widetilde{B_{S_j}} = h\left(R_{Sj0} || T_2\right)$. Finally, S_j sends { ID_{S_j}, $\overline{B_{S_j}}$, $\widetilde{B_{S_j}}$, T_2 } to the HG and stores A_{S_j} in its storage space.

Step 3. HG computes $R_{S_j0}' = h(h(ID_{S_j} || X_{HG})) \oplus \overline{B_{S_j}}$ and checks $\widetilde{B_{S_j}} =? h\left(R_{S_j0}' || T_2\right)$ and if T_2 is valid or not by examining ΔT. If $\widetilde{B_{S_j}}$ is equal to $h\left(R_{S_j0}' || T_2\right)$ and T_2 is valid, then HG ensures that S_j is a legal sensor and $R_{S_j0} = R_{Sj0}$. Finally, the HG stores ID_{S_j} in its registration list, which keeps the identities of all legal sensors Fig. 5.

3.3 User registration and authentication phase

In this phase, the user registers at HG to access the WSN. Then, the HG and the user mutually authenticate each other's validities. The steps of this phase are listed as follows.

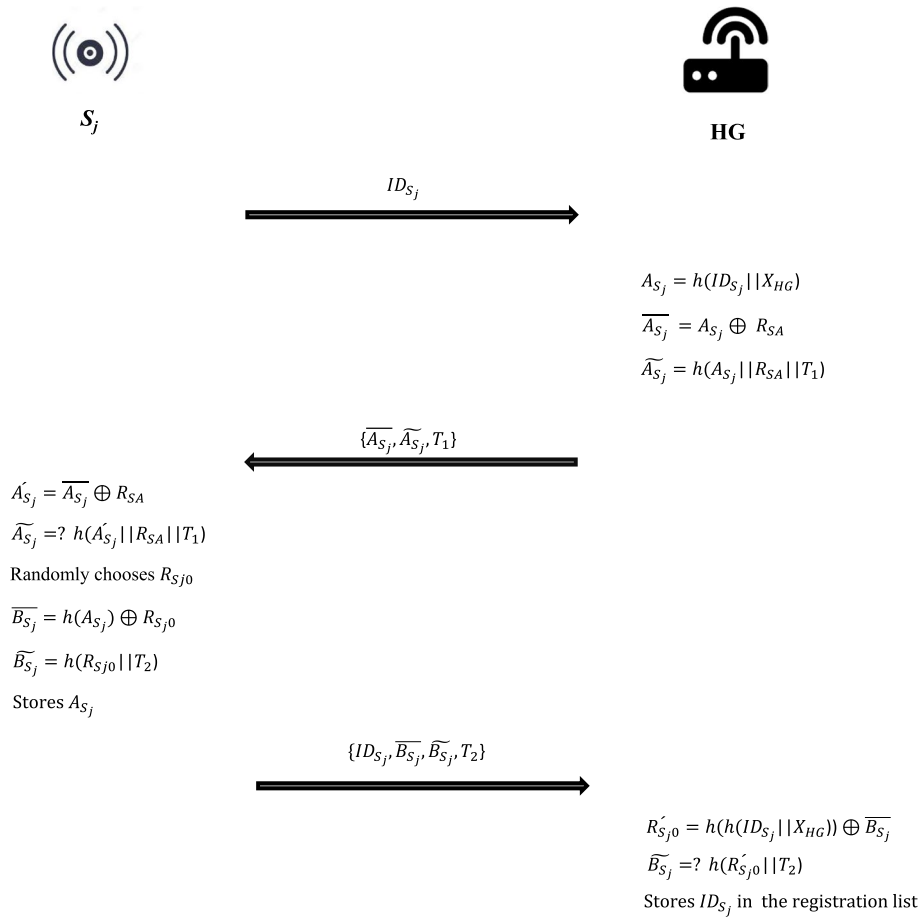

Fig. 5 Sensor registration and authentication phase

Step 1. U_i inputs ID_i and PW_i into the mobile device. If ID_i and PW_i are both correct, then the mobile device randomly chooses TID_{U_i} and sends it to the HG. After that, HG computes $A_{U_i} = h\left(TID_{U_i} || X_{HG}\right)$, $\overline{A_{U_i}} = A_{U_i} \oplus R_{SA}$, and $\widetilde{A_{U_i}} = h\left(A_{U_i} || R_{SA} || T_3\right)$ and sends $\{\overline{A_{U_i}}, \widetilde{A_{U_i}}, T_3\}$ to U_i.

Step 2. U_i computes $A_{U_i} = \overline{A_{U_i}} \oplus R_{SA}$ and checks $\widetilde{A_{U_i}} = ?h\left(A_{U_i} || R_{SA} || T_3\right)$. If $\widetilde{A_{U_i}}$ is equal to $h\left(A_{U_i} || R_{SA} || T_3\right)$ and T_3 is valid, then U_i ensures that HG is a legal gateway and $A_{U_i} = A_{U_i}$. After that, U_i randomly chooses R_{U_i0} to compute $\overline{B_{U_i}} = h\left(A_{U_i}\right) \oplus R_{U_i0}$ and $\widetilde{B_{U_i}} = h\left(R_{U_i0} || T_4\right)$. Then, U_i sends $\{TID_{U_i}, \overline{B_{U_i}}, \widetilde{B_{U_i}}, T_4\}$ to the HG. Finally, U_i stores A_{U_i} in the mobile device.

Step 3. HG computes $R_{U_i0} = h(h(TID_{U_i} || X_{HG})) \oplus \overline{B_{U_i}}$ and checks $\widetilde{B_{U_i}} = ?h\left(R_{U_i0} || T_4\right)$. If $\widetilde{B_{U_i}}$ is equal to $h\left(R_{U_i0} || T_4\right)$ and T_4 is valid, then HG ensures that U_i is a valid user and $R_{U_i0} = R_{U_i0}$. Finally, HG stores TID_{U_i} in the registration list of its database Fig. 6.

3.4 Data access phase

This phase can be divided into three subphases: the user requiring phase, local data access phase and the foreign data access phase. To access data from the WSN, the user sends the requiring message to the HG by performing the user requiring phase. Then, the HG authenticates the user and negotiates a session key between them. If the user wants to access the data from the local area, then the user and HG perform the local data access phase. Otherwise, they perform the foreign data access phase, and the user gets the data from a foreign gateway through the HG. The steps of this phase are listed as follows.

3.4.1 User requiring phase

Step 1. U_i inputs ID_i and PW_i in the mobile device. If ID_i and PW_i are both correct, then the mobile device randomly chooses R_{U_i1} to compute $C_{U_i} = h\left(A_{U_i}\right) \oplus R_{U_i1}$ and $\overline{C_{U_i}} = h\left(R_{U_i1} || ID_{S_j} || T_5\right)$. Then, U_i sends $\{TID_{U_i}, ID_{sj}, C_{U_i}, \overline{C_{U_i}}, T_5\}$ to the HG.

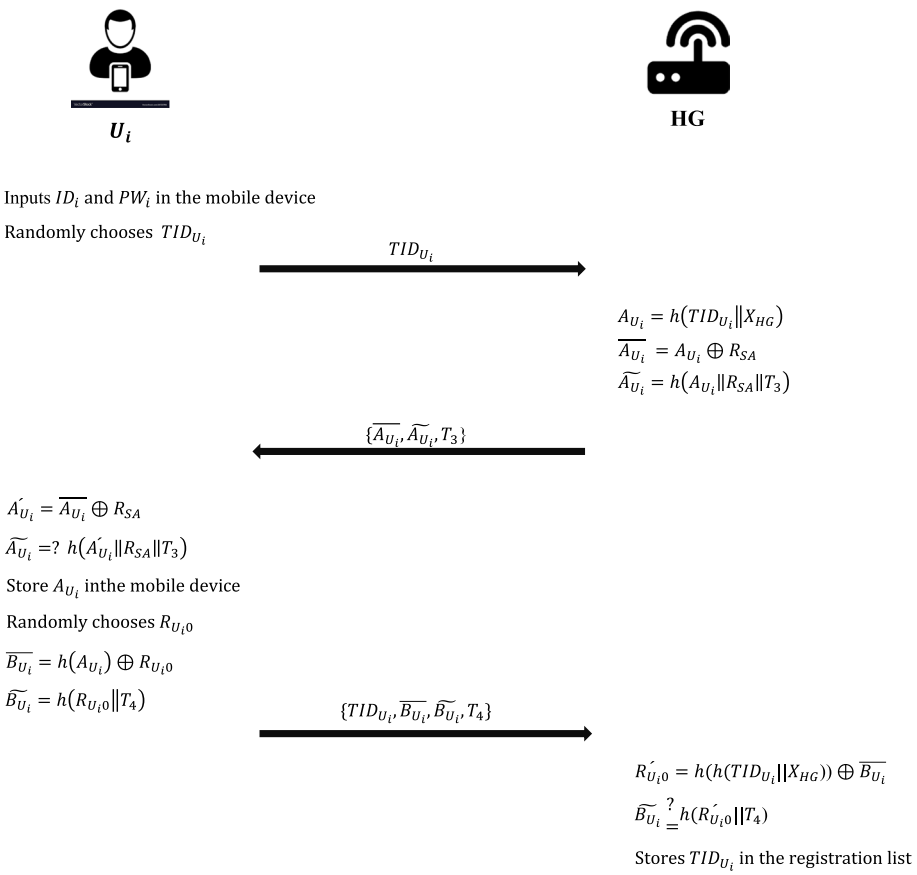

Fig. 6 User registration and authentication phase

Step 2. The HG computes $R_{U_i}1 = h(h(TID_{U_i}||X_{HG}))\oplus C_{U_i}$ and checks $\overline{C_{U_i}} = ?h\left(R_{U_i}1||ID_{S_j}||T_5\right)$. If $\overline{C_{U_i}}$ is equal to $h\left(R_{U_i}1||ID_{S_j}||T_5\right)$ and T_5 is valid, then the HG ensures that U_i is a legal user and $R_{U_i}1 = R_{U_i}1$.

Step 3. HG generates R_{HG1} to compute $K_{UH} = h\left(TID_{U_i}||R_{U_i}1||R_{HG1}\right)$, $C_{HG} = (h(TID_{U_i}||X_{HG}))\oplus R_{HG1}$, and $\overline{C_{HG}} = h\left(R_{HG1}||K_{UH}||T_6\right)$. Then, HG sends $\{C_{HG}, \overline{C_{HG}}, T_6\}$ to U_i.

Step 4. U_i computes $R_{HG1} = h\left(A_{U_i}\right) \oplus C_{HG}$ and $K_{UH} = h\left(TID_{U_i}||R_{U_i}1||R_{HG1}\right)$ and checks $\overline{C_{HG}} = ?h\left(R_{HG1}||K_{UH}||T_6\right)$. If $\overline{C_{HG}}$ is equal to $h\left(R_{HG1}||K_{UH}||T_6\right)$ and T_6 is valid, then U_i ensures that HG is a legal gateway and $K_{UH} = K_{UH}$.

Step 5. HG checks if ID_{sj} is in the local area or not. If it is in the local area, then the HG performs the local data access phase in Subsection 3.4.2. Otherwise, the HG communicates with the FG and performs the foreign data access phase in Subsection 3.4.3 Fig. 7.

3.4.2 Local data access phase

Step1. HG computes $F_{HG} = h(h(ID_{S_j}||X_{HG}))\oplus R_{HG1}$ and $\overline{F_{HG}} = h\left(R_{HG1}||T_7\right)$, then it sends $\{F_{HG}, \overline{F_{HG}}, T_7\}$ to S_j.

Step 2. S_j computes $R_{HG1} = h\left(A_{S_j}\right) \oplus F_{HG}$ and checks $\overline{F_{HG}} = ?h\left(R_{HG1} \,||\, T_7\right)$. If $\overline{F_{HG}}$ is equal to $h\left(R_{HG1} \,||\, T_7\right)$ and T_7 is valid, then S_j ensures that the HG is a legal gateway and $R_{HG1} = R_{HG1}$. After that, S_j generates $R_{S_j}1$ to compute $K_{SH} = h\left(ID_{S_j} \,||\, R_{S_j}1 \,||\, R_{HG1}\right)$, $F_{S_j} = h\left(A_{S_j}\right) \oplus R_{S_j}1$, $\overline{F_{S_j}} = h\left(R_{S_j}1 \,||\, K_{SH} \,||\, T_8\right)$, and $EDATA_1 = E_{K_{SH}}(DATA)$. Finally, S_j sends $\{ID_{S_j}, F_{S_j}, \overline{F_{S_j}}, EDATA_1, T_8\}$ to the HG.

Step 3. HG computes $R_{S_j}1 = h(h(ID_{S_j} \,||\, X_{HG})) \oplus F_{sj}$ and $K_{SH} = h\left(ID_{S_j} \,||\, R_{S_j}1 \,||\, R_{HG1}\right)$ to check $\overline{F_{S_j}} = ?h\left(R_{S_j}1 \,||\, K_{SH} \,||\, T_8\right)$. If $\overline{F_{S_j}}$ is equal to $h\left(R_{S_j}1 \,||\, K_{SH} \,||\, T_8\right)$ and T_8 is valid, then the HG ensures that S_j is a legal sensor and $R_{S_j}1 = R_{Sj1}$ and $K_{SH} = K_{SH}$. Note that K_{SH} is a session key between S_j and the HG.

U_i

HG

Inputs ID_i and PW_i

Randomly chooses R_{U_i}

$C_{U_i} = h(A_{U_i}) \oplus R_{U_i}1$

$\overline{C_{U_i}} = h(R_{U_i}1||ID_{S_j}||T_5)$

$\{TID_{U_i}, ID_{sj}, C_{U_i}, \overline{C_{U_i}}, T_5\}$ →

$R_{U_i}1 = h(h(TID_{U_i}||X_{HG})) \oplus C_{U_i}$

$\overline{C_{U_i}} = ?\ h(R_{U_i}1||ID_{S_j}||T_5)$

Randomly chooses R_{HG1}

$K_{UH} = h(TID_{U_i}||R_{U_i}1||R_{HG1})$

$C_{HG} = (h(TID_{U_i}||X_{HG})) \oplus R_{HG1}$

$\overline{C_{HG}} = h(R_{HG1}||K_{UH}||T_6)$

← $\{C_{HG}, \overline{C_{HG}}, T_6\}$

$R_{HG1} = h(A_{U_i}) \oplus C_{HG}$

$K_{UH} = h(TID_{U_i}||R_{U_i}1||R_{HG1})$

$\overline{C_{HG}} = ?\ h(R_{HG1}||K_{UH}||T_6)$

Fig. 7 User requiring phase

Step 4. To decrypt the collected data from S_j, the HG uses K_{SH} to compute $DATA = D_{K_{SH}}(EDATA_1)$. After that, the HG uses the session key K_{UH} negotiated by Subsection 3.4.1 to compute $EDATA_2 = E_{K_{UH}}\left(DATA||T_9\right)$. Finally, the HG sends $\{ID_{HG},\ EDATA_2,\ T_9\}$ to U_i.

Step 5. U_i computes $(DATA\ ||\ T_9\Big) = D_{K_{UH}}(EDATA_2)$ to get $DATA$ and T_9. And, U_i checks if T_9 is valid or not by ΔT. If T_9 is valid, then U_i accepts the data. Otherwise, U_i discards it Fig. 8.

3.4.3 Foreign data access phase

Step 1. HG generates R_{HF} to compute $F_{HF}=h(ID_{HG}\ ||\ R_{SA})\oplus R_{HF}$ and $\overline{F_{HF}} = h\Big(R_{HF}\ ||\ ID_{S_j}\ ||\ T_7\Big)$. Then, the HG sends $\{\ ID_{HG},\ \ F_{HF}, \overline{F_{HF}},\ \ T_7\ \}$ to FG.

Step 2. After getting $\{ID_{HG},\ \ F_{HF}, \overline{F_{HF}},\ \ T_7\}$, FG computes $R_{HF}^{'} = h\Big(ID_{HG}\Big\|R_{SA}\Big) \oplus F_{HF}$ and checks $\overline{F_{HF}}$ =?$h\Big(R_{HF}^{'}||ID_{S_j}||T_7\Big)$. If $\overline{F_{HF}}$ is equal to $h\Big(R_{HF}^{'}||ID_{S_j}||T_7\Big)$ and T_7 is valid, then the FG ensures that HG is a legal gateway and $R_{HF} = R_{HF}^{'}$. After that, the FG generates R_{FH} and R_{FG1} to compute $K_{FH}=h(R_{HF}^{'}||R_{FH})$, $F_{FG} = h(h(ID_{S_j}||X_{FG}))\oplus R_{FG1}$, and $\overline{F_{FG}} = h\Big(R_{FG1}||T_8\Big)$. Finally, the FG sends $\{F_{FG},\ \ \overline{F_{FG}},\ \ T_8\}$ to S_j.

Step 3. S_j computes $R_{FG1} = h\Big(A_{S_j}\Big) \oplus F_{FG}$ and checks $\overline{F_{FG}}$ =?$h\Big(R_{FG1}^{'}\ ||\ T_8\Big)$. If $\overline{F_{FG}}$ is equal to $h\Big(R_{FG1}^{'}\ ||\ T_8\Big)$ and T_8 is valid, then S_j ensures that the FG is a legal gateway and $R_{FG1} = R_{FG1}^{'}$. After that, S_j generates R_{Sj1} to compute $K_{SF} = h\Big(ID_{S_j}\ ||$

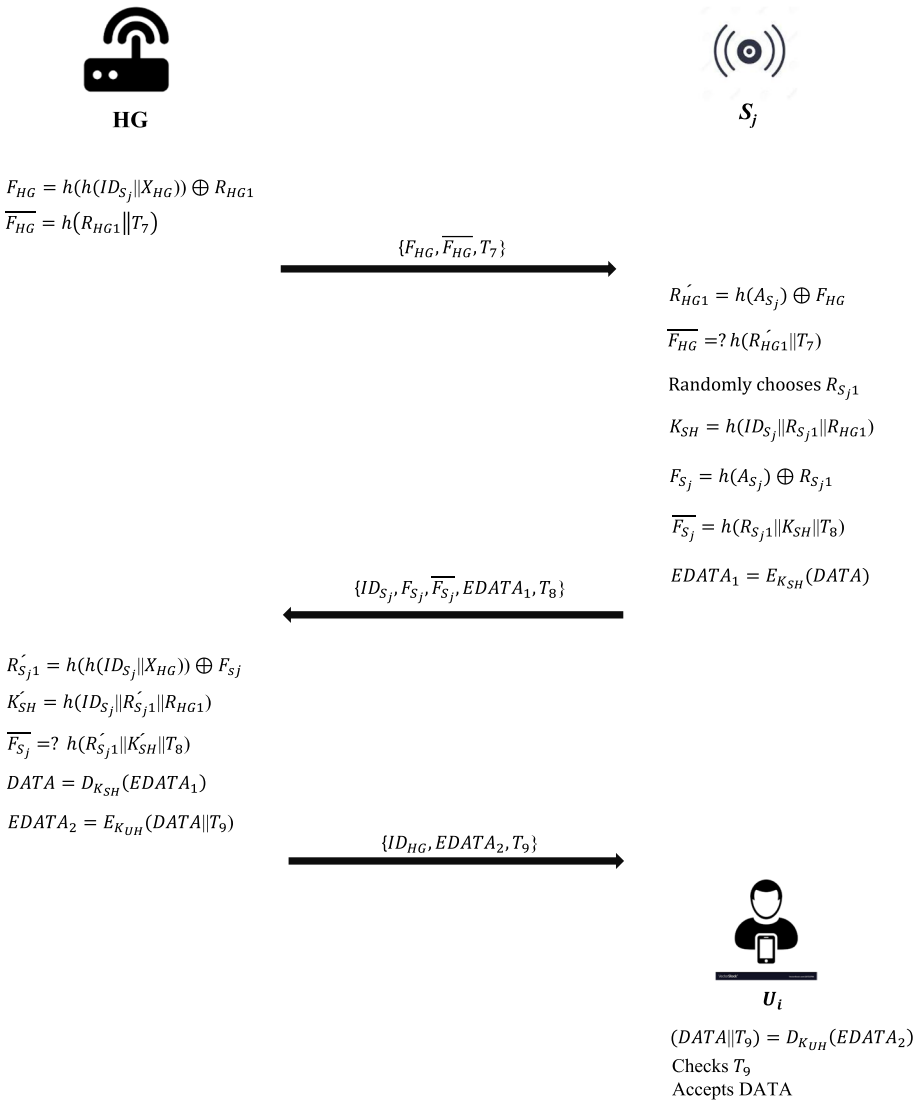

Fig. 8 Local data access phase

R_{S_j1} || R_{FG1}), $F_{S_j} = h\left(A_{S_j}\right) \oplus R_{Sj1}$, $\overline{F_{S_j}} = h\left(R_{S_j1}\right.$ || $\left.K_{SF}\right.$ || T_9), and $EDATA_1 = E_{K_{SF}}(DATA)$. Finally, S_j sends $\{ ID_{S_j}, F_{sj}, \overline{F_{S_j}}, EDATA_1, T_9 \}$ to the FG.

Step 4. FG computes $R_{S_j1} = h(h(ID_{S_j}||X_{FG}))\oplus F_{sj}$ and $K_{SF} = h\left(ID_{S_j}||R_{S_j1}||R_{FG1}\right)$ to check $F_{S_j} = ?h\left(R_{S_j1}\right.$ $\left.||K_{SF}||T_9\right)$. If F_{S_j} is equal to $h\left(R_{S_j1}||K_{SF}||T_9\right)$ and T_9 is valid, then the FG ensures that S_j is a legal sensor and $R_{S_j1} = R_{Sj1}$.

Step 5. FG computes $DATA = D_{K_{SF}}(EDATA_1)$, $EDATA_2 = E_{K_{HF}}(DATA)$, $F_{FH} = h(ID_{HF}$ || $R_{SA}) \oplus R_{FH}$, and $\overline{F_{FH}} = h\left(R_{FH}\right.$ || $\left.K_{HF}\right.$ || T_{10}). Then, the FG sends $\{ ID_{FG}, F_{FH}, \overline{F_{FH}}, EDATA_2, T_{10} \}$ to the HG.

Step 6. HG computes $R_{FH} = h\left(ID_{HF}||R_{SA}\oplus F_{FH}\right)$ and $K_{HF} = h\left(R_{HF}||R_{FH}\right)$ to check $\overline{F_{FH}} = ?h\left(R_{FH}\right.$ $\left.||K_{HF}||T_{10}\right)$. If $\overline{F_{FH}}$ is equal to $h\left(R_{FH}||K_{HF}||T_{10}\right)$ and T_{10} is valid, then the HG ensures that the FG is a legal gateway and $R_{FH} = R_{FH}$ and $K_{HF} = K_{HF}$. After that, the HG computes $DATA = E_{K_{HF}}(EDATA_2)$ and $EDATA_3 = E_{K_{UH}}\left(DATA||T_{11}\right)$ to send $\{ID_{HG}, EDATA_3, T_{11}\}$ to U_i.

Step 7. U_i computes $(DATA\,||\,T_{11}) = D_{K_{UH}}(EDATA_3)$ to get $DATA$ and T_{11}. And, U_i checks if T_{11} is valid or not by ΔT. If T_{11} is valid, then U_i accepts the data. Otherwise, U_i discards it Fig. 9.

Compared with related works [17, 18], the proposed scheme has the following advantages. First, the proposed scheme allows the user and the sensor to register and join dynamically with WSN without using a secure channel. Thus, the proposed scheme is more suitable for the dynamic network topology of IoT. Second, the proposed scheme is a multi-gateway structure, and thus it can be applied to various IoT applications. Third, unlike the related works [17, 18], the proposed scheme has local and foreign data access phases to show how to access data securely in the multi-gateway WSN. Thus, the proposed scheme is more completed and practical than related works. Fourth, the proposed scheme is only designed by the light-weight operations (XOR operation and one-way hash function). Therefore, the proposed scheme has low computation and communication loads.

4 Security and performance analyses

In this section, I perform some attacks on the proposed scheme to analyze its security. In addition, the performance analyses of the related works are also provided. The proposed attacks are shown as follows.

4.1 Outsider attack

Assuming that an outside attacker wants to get R_{U_i1} and R_{HG1} to compute the session key $K_{UH} = h\left(TID_{U_i}||R_{U_i1}\right.$ $\left.||R_{HG1}\right)$, and the outsider will wiretap the communications between the user and HG in Subsection 3.4.1 to obtain $\{TID_{U_i}, ID_{sj}, C_{U_i}, \overline{C_{U_i}}, T_5\}$ and $\{C_{HG}, \overline{C_{HG}}, T_6\}$. To get R_{U_i1}, the attacker has to compute the equation $R_{U_i1} = h(h(TID_{U_i}||X_{HG}))\oplus C_{U_i}$. However, it is impossible because the attacker does not know $A_{U_i} = h\left(TID_{U_i}||X_{HG}\right)$ which is a secret authentication information stored in the user's devices. For the same reason, the attacker cannot obtain R_{HG1} from the equation $R_{HG1} = h\left(A_{U_i}\right) \oplus C_{HG}$.

Based upon the similar analysis, an attacker cannot get the session keys $K_{SH} = h\left(ID_{S_j}\right.$ || R_{S_j1} || $\left.R_{HG1}\right)$, $K_{SF} = h\left(ID_{S_j}\right.$ || R_{S_j1} || $\left.R_{FG1}\right)$, and $K_{FH} = h(R_{HF}$ || $R_{FH})$ because he does not know the secret authentication information A_{U_i} and A_{S_j}. According to the above analysis, the outsider attack is infeasible for the proposed scheme.

4.2 Insider attack

Assuming that an inside attacker U_i (a malicious user) wants to get the secret key X_{HG} of the home gateway (HG), and the insider will try to compute X_{HG} from $A_{U_i} = h\left(TID_{U_i}\right.$ || $\left.X_{HG}\right)$ stored in his device. However, this attack is infeasible because X_{HG} is protected by a secure one-way hash function $h(\cdot)$. Computing X_{HG} from $h\left(ID_{S_j}\right.$ || $\left.X_{HG}\right)$ is impossible [16].

On the other hand, if an attacker has controlled a sensor S_j and obtained its authentication information $A_{S_j} = h\left(ID_{S_j}\right.$ || $\left.X_{HG}\right)$, then he tries to compute X_{HG} from $h\left(ID_{S_j}\right.$ || $\left.X_{HG}\right)$. However, this attack is also impossible because X_{HG} is protected by the one-way hash function $h(\cdot)$. According to the above analyses, the insider attack is infeasible for the proposed scheme.

4.3 Impersonating attack

Assuming that an attacker does not register at the SA and tries to impersonate a legal user U_i, and the attacker is going to execute the steps of Subsection 3.3 to register at the HG. Thus, the attacker intercepts $\{ \overline{A_{U_i}}, A_{U_i}, T_3 \}$ in Step 1 of Subsection 3.3, and then he uses the fake authentication information $A_{U_i\_fake}$ to compute $\overline{B_{U_i}} = h\left(A_{U_i\_fake}\right) \oplus R_{U_i0}$ and $\widetilde{B_{U_i}} = h\left(\widetilde{R_{U_i0}}\right.$ || $\left.T_4\right)$. After that, the attacker sends $\{ \widetilde{TID_{U_i}}, \overline{B_{U_i}}, \widetilde{B_{U_i}}, T_4 \}$ to the HG. After receiving $\overline{B_{U_i}}$ and $\widetilde{B_{U_i}}$, the HG computes $R_{U_i0} = h(h(A_{U_i})) \oplus \overline{B_{U_i}}$ and checks $\widetilde{B_{U_i}} = ?h\left(R_{U_i0}\right.$ || $\left.T_4\right)$, where $A_{U_i} = h\left(TID_{U_i}\right.$ || $\left.X_{HG}\right)$. However, the HG will find that $\widetilde{B_{U_i}}$ is not equal to $h\left(R_{U_i0}\right.$ || $\left.T_4\right)$ because $A_{U_i\_fake} \neq A_{U_i}$ and $R_{U_i0} \neq R_{U_i0}$. That is, the HG

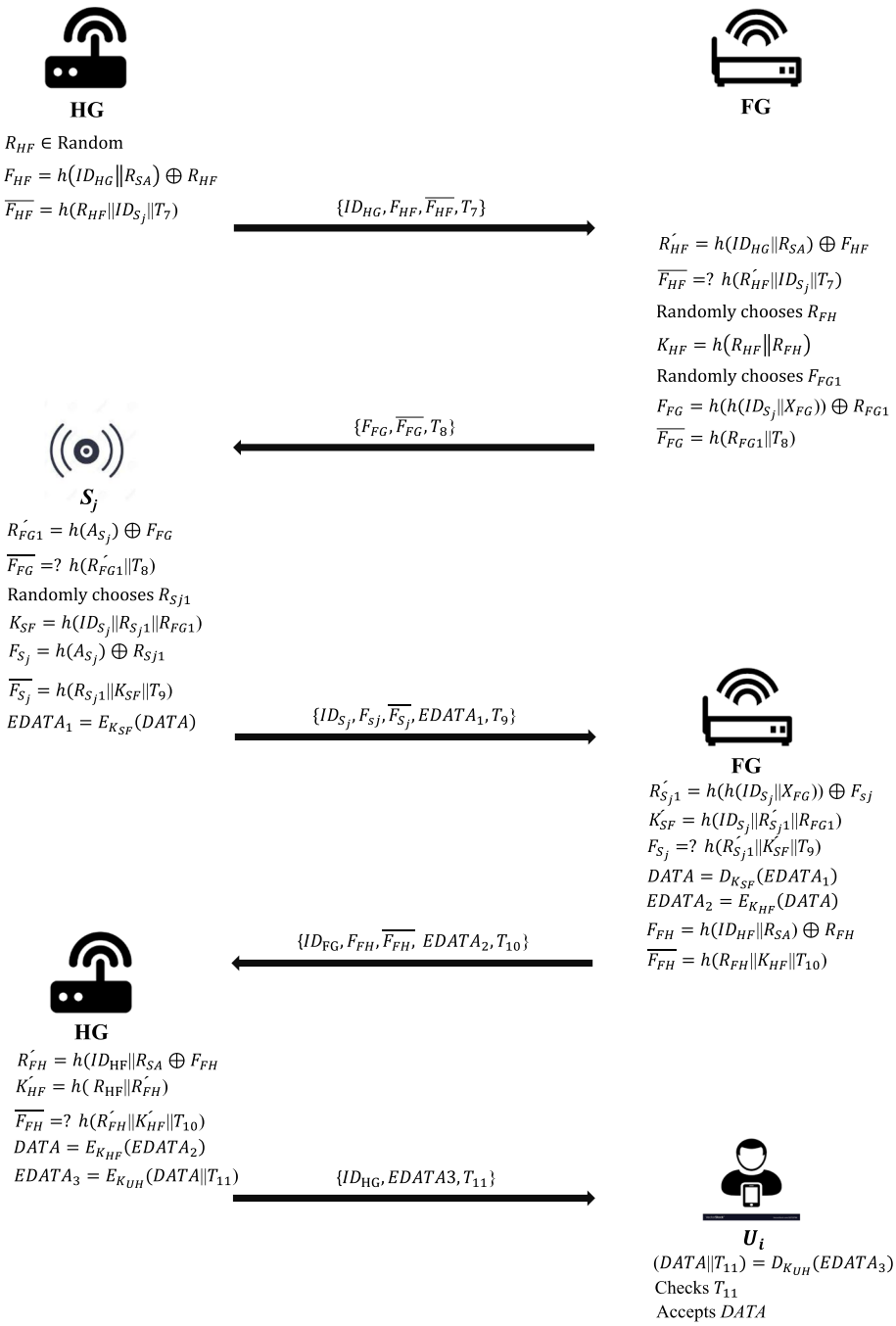

Fig. 9 Foreign data access phase

will know the attacker is not a legal user and deny his registration.

Similarly, impersonating a legal user in Subsection 3.4.1 is impossible because an attacker does not know the authentication information A_{U_i}. The HG can discover an attacker by checking $\overline{C_{U_i}} =? h\left(R'_{U_i1} \,||\, ID_{S_j} \,||\, T_5\right)$ in Step 2 of Subsection 3.4.1. According to the above analyses, the impersonating attack is infeasible for the proposed scheme.

4.4 Men-in-the-middle attack

Assuming that an attacker intercepts the communications between the sensor and the HG in Step 2 of Subsection 3.2, then he tries to alter $\{ \overline{A_{S_j}}, A_{S_j}, T_1 \}$ and resends the

Table 3 The performance analysis of the related works

Roles Papers	User	Gateway	Sensor	Total
[17]	$15Th$	$15Th$	$5Th$	$33Th$
[18]	$13Th$	$18Th$	$6Th$	$37Th$
[19]	$13Th+2T_{ecm}$	$13Th$	$4Th+2T_{ecm}$	$30Th+4T_{ecm}$
[20]	$5Th+3T_{ecm}+2T_{sym}$	$5Th+3T_{ecm}+2T_{sym}$	$3Th+2T_{ecm}$	$13Th+8T_{ecm}+4T_{sym}$
Proposed scheme	$8Th$	$13Th$	$3Th$	$24Th$

tampered message to the sensor. The attacker may compute $\overline{A_{S_j}}" = A_{S_j}{}' \oplus R_{SA}"$, and $\tilde{A}_{S_j}" = h\left(A_{S_j}{}' \,||\, R_{SA}{}' \,||\, T_1'\right)$, and then he sends $\{\overline{A_{S_j}}", \tilde{A}_{S_j}", T_1{}'\}$ to S_j. However, S_j will discover that $\{\overline{A_{S_j}}', A_{S_j}', T_1"\}$ is a fake message by computing $A_{S_j}' = \overline{A_{S_j}}" \oplus R_{SA}$ and $\tilde{A}_{S_j}" = ?h\left(A_{S_j}' \,||\, R_{SA} \,||\, T_1\right)$. Obviously, $\tilde{A}_{S_j}"$ is not equal to $h\left(A_{S_j}' \,||\, R_{SA} \,||\, T_1\right)$ because $\tilde{A}_{S_j}" = h\left(A_{S_j}{}' \,||\, R_{SA}{}' \,||\, T_1'\right) \neq h\left(A_{S_j}' \,||\, R_{SA} \,||\, T_1\right)$. According to the above analysis, the proposed scheme can prevent from the men-in-the-middle attack.

4.5 Replay attack

Assuming that an attacker intercepts $\{\overline{A_{S_j}}, \tilde{A}_{S_j}, T_1\}$, and then he resends it to S_j without any alteration in Subsection 3.2. However, the sensor will discover $\{\overline{A_{S_j}}, A_{S_j}, T_1\}$ is re-sent by an attacker because S_j must check if T_1 is valid or not by examining ΔT. Definitely, the re-sending time will exceed the valid time interval ΔT, and thus the sensor can discover this attack. Based upon the similar analysis, an attacker cannot perform this kind of attack on the user registration and authentication phase and data access phase because the timestamps are used in the proposed scheme. According to the above analysis, the proposed scheme can prevent from the replay attack.

4.6 Performance analysis

The performance analyses of the related works [17–20] and the proposed scheme are shown in Table 3. Note that the proposed scheme and [17] are multi-gateway structures for WSN and the others are not. In addition, only the proposed scheme provides the data access phase after mutual authentication and key agreement have been done. Under the same benchmark, the performance analysis of the data access phase of the proposed scheme is eliminated. In Table 3, T_h, T_{ecm}, and T_{sym} are the execution time of the one-way hash function, point multiplication of the elliptic curve cryptosystem [16], and symmetric encryption/decryption, respectively. According to [18], the execution time of T_h, T_{ecm}, and T_{sym} are about 0.00032, 0.0171, and 0.0056 seconds, respectively. That is, $T_{ecm} > T_{sym} > T_h$. Moreover, the measurement does not consider the execution time of XOR operation because its computation cost is negligible. According to Table 3, the proposed scheme is more efficient than related works.

5 Conclusions

In this paper, I proposed a multi-gateway authentication and key-agreement scheme on wireless sensor networks for IoT. Unlike related works, the proposed scheme allows the user and the sensor to register at different gateways dynamically, and thus the user and the sensor can securely join in different areas of wireless sensor networks. In addition, the proposed multi-gateway structure can be applied to more IoT applications in comparison with single-gateway related works. According to the proposed performance analysis, the computation costs of the proposed scheme are much less than those of the related works. In conclusion, the proposed scheme is more efficient and practical than related works for IoT applications.

Acknowledgements
No acknowledgement.

Author's contributions
All works are completed by Jen-Ho Yang. The author(s) read and approved the final manuscript.

Author's information
Jen-Ho Yang, associate professor, Department of Digital Multimedia Design, Kainan University, Taiwan.

Funding
No funds, grants, or other support was received.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests
The authors indicate no potential conflicts of interest.

Received: 2 May 2022 Accepted: 20 February 2023
Published online: 08 March 2023

References

1. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014)
2. F. Wortmann, K. Flüchter, Internet of things: Technology and value added. Bus. Inf. Syst. Eng. **57**(3), 221–224 (2015)
3. M.S. Farash, M. Turkanović, S. Kumari, M. Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. Ad Hoc Netw. **36**(1), 152–176 (2016)
4. O. Bello, S. Zeadally, Intelligent device-to-device communication in the internet of things. IEEE Syst. J. **10**(3), 1172–1182 (2016)
5. A. Osseiran, O. Elloumi, J.S. Song, J.F. Monserrat, Internet of things. IEEE Communications Standards Magazine **1**(2), 84 (2017)
6. R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus, TinyPK: Securing sensor networks with public key technology. SASN **4**, 59–64 (2004)
7. S. Lee, J. Lee, H. Sin, S. Yoo, S. Lee, J. Lee, Y. Lee, S. Kim, An energy-efficient distributed unequal clustering protocol for wireless sensor networks. World Acad. Sci. Eng. Technol. **48**, 12–23 (2008)
8. B. Vaidya, J. Silva and J. Rodrigues, "Robust dynamic user authentication scheme for wireless sensor networks", Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks, pp. 88-91, 2009
9. R. Song, Advanced smart card based password authentication protocol. Comput. Standards Interfaces **32**, 321–325 (2010)
10. A. Das, P. Sharma, S. Chatterjee, J. Sing, A dynamic password based user authentication scheme for hierarchical wireless sensor networks. J. Netw. Comput. Appl. **35**, 1646–1656 (2012)
11. D. Sun, J. Li, Z. Feng, Z. Cao, G. Xu, On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. Pers. Ubiquit. Comput. **17**, 895–905 (2013)
12. K. Xue, C. Ma, P. Hong, R. Ding, A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J. Netw. Comput. Appl. **36**, 316–323 (2013)
13. M. Turkanovic, B. Brumen, M. Hölbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Netw. **20**, 96–112 (2014)
14. R. Amin, G. Biswas, Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wireless Personal Commutations **84**, 1–24 (2015)
15. R. Amin, G. Biswas, A novel user authentication and key agreement protocol for accessing multi-medical server usable in TMIS. J. Med. Syst. **39** (2015 [Online].). https://doi.org/10.1007/s10916-015-0217-3
16. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition, Wiley, New York, 1996
17. R. Amin, G. Biswas, A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw. **36**(1), 58–80 (2016)
18. D.K. Kwon, S.J. Yu, J.Y. Lee, S.H. Son, Y.H. Park, WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks. Sensors **21**(3), 936–958 (2021)
19. F. Wu, L. Xu, S. Kumari, X. Li, A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security. J. Ambient. Intell. Humaniz. Comput. **8**, 101–116 (2017)
20. M.F. Moghadam, M. Nikooghadam, M.A.B.A. Jabban, M. Alishahi, L. Mortazavi, A. Mohajerzadeh, An efficient authentication and key agreement scheme based on ECDH for wireless sensor network. IEEE Access **8**, 73182–73192 (2020)
21. C. Stergiou, K.E. Psannis, B.B. Gupta, Y. Ishibashi, Security, Privacy & Efficiency of sustainable cloud computing for big data & IoT. Sustainable Computing: Informatics and Systems **19**, 174–184 (2018)
22. V. Adat, B.B. Gupta, Security in internet of things: Issues, challenges, taxonomy, and architecture. Telecommun. Syst. **47**, 423–441 (2018)
23. A. Tewari, B. Gupta, Secure timestamp-based mutual authentication protocol for IoT devices using RFID tags. Int. J. Semant. Web Inf. Syst. **16**(3), 20–34 (2020)

Publisher's Note