An Anonymous Authentication and Key Agreement Protocol in Smart Living

Fengyin Li[a], Xinying Yu[b], Yang Cui[a], Siqi Yu[a], Yuhong Sun[a], Yilei Wang[a,*], Huiyu Zhou[c]

[a]*School of Computer Science, Qufu Normal University, Rizhao, 276826, China*
[b]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China*
[c]*School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, United Kingdom*

Abstract

Wireless Sensor Networks (WSNs) play an indispensable role in the application of smart homes, smart healthcare, and precision agriculture. However, WSNs confront privacy risks that hinder its practical applications. The leakage of privacy is one of the key factors to restrict the development of WSNs. Hence, in this paper, we propose an Anonymous Authentication and Key Agreement protocol (AAKA) to accomplish identity authentication and privacy protection. Based on the dynamic sequence number, the shared secret value, and the dynamic random number, the AAKA protocol implements a two-way authentication and keys negotiation among users, gateway, and sensors, which achieves the secure access control of legitimate users to WSNs and ensures the confidential transmission of data over the public channel. We perform the security proof with BAN logic for security evaluation. The performance analysis demonstrates that compared with other WSNs authentication schemes, the AAKA protocol obtained better security features, smaller storage, and more efficient communication. Therefore, it is more suitable for applications in smart living.

Keywords: Smart Living, Wireless Sensor Networks, AAKA protocol, Identity Authentication, Key Agreement

*Corresponding author
Email address: wang_yilei2019@qfnu.edu.cn (Yilei Wang)

1. Introduction

Wireless sensor networks are composed of a large number of resource-constrained and self-organized sensor nodes, mostly used in environmental monitoring, smart healthcare, precision agriculture, and other fields [1, 2]. The pivotal wireless communication and data fusion technologies have become increasingly mature in WSNs. However, the information security issues have become the current research hotspot due to high privacy requirements in WSNs. In the security research of WSNs, identity authentication is one of the key security technologies and is the fundamental basis of data access control [3]. Identity authentication can ensure that legitimate users safely access sensor nodes, preventing unauthorized users from entering the network to obtain sensitive information, which is extensively applied in various domains, for example, Internet of Things (IoT) [4, 5, 6, 7], Mobile Edge Computing (MEC) [8], or medical scenes [9].

The security of WSNs is manifested not only in the legitimacy of communication entities but also in the confidential transmission of sensitive data. The data collected by sensor nodes is particularly sensitive and often related to the privacy of users. However, the data confront the risk of eavesdropping and leakage of privacy since the sensor nodes transmit data wirelessly over the open channel. Therefore, WSNs usually use encryption to protect the sensitive data transmitted on the public channel [10]. However, only using encryption technologies cannot resist some common attacks, such as impersonation attacks, and replay attacks. Consequently, identity authentication and anonymous mechanism are needed for cooperation with an encryption scheme. Identity authentication and key agreement protocol can not only authenticate the identity legitimacy of the communication entities, but also generate a session key, which is used to encrypt the communication data. Anonymity mechanism protects the real identity of communicating entities from being exposed to attackers and enables anonymous communication between entities [11, 12]. Therefore, the design of a secure anonymous identity authentication and key agreement protocol is neces-

sary to provide the basic security for the whole wireless sensor network. This paper presents a lightweight identity authentication and key negotiation protocol following the architecture and security demand of WSNs, which forms a secure foundation for the widespread application of wireless sensor networks in the Internet of things field.

1.1. Related Work

User identity authentication is an important mechanism to ensure environmental security and user privacy in WSNs. Many researchers at home and abroad have successively proposed identity authentication technologies and key agreement protocols based on different principles [13, 14].

Literature [15] reports a superelliptic curve cryptosystem based identity authentication protocol. Its security is founded on the superelliptic curve discrete logarithm problem. The protocol realizes the authentication between two sensors but cannot verify the user's legitimacy. In reference [16], a password-based identity authentication and key consensus scheme is proposed, which realizes mutual authentication among users, gateway, and sensor nodes. However, the user transmits the identity of the sensor node directly to the gateway through the public channel in the authentication phase, which cannot guarantee the anonymity of the node. Literature [17] presents a three-factor anonymous authentication scheme under a wireless sensor network, and uses a fuzzy commitment scheme to process the user's biological information. While the scheme satisfies multiple security goals, it also has a relatively high computational and communication overhead. Literature [18] proposes a new security authentication and key agreement scheme, which uses dynamic pseudonymous identity to ensure user privacy and eliminate redundant calculations to improve efficiency. In literature [19], an improved lightweight identity authentication scheme is proposed to deal with various security vulnerabilities in the scheme proposed by Kumari et al. [20], such as smart card theft attack, denial of service attack, and user traceability. Zhang et al. propose two identity authentication models (USD and UDS) in reference [21] and design authentication schemes for

the two models, respectively. However, both schemes have design loopholes. Specifically, at the end of the identity authentication phase, smart card lacks an update to a parameter, which will cause the legitimate user to fail the smart card verification at the next login. Lee etc. present a secure and efficient authentication protocol based on three-factor authentication by taking advantage of biometrics and honey_list technique [22]. Their protocol can provide security even if two of the three factors are compromised. Literature [23] proposes a lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-Based IIoT (LAPTAS). The LAPTAS scheme obtains some excellent features such as perfect forward secrecy, privacy-preserving, biometric template privacy, and revocation support is more resilient against several attacks. Nashwan etc., propose an anonymous access authentication scheme for wireless sensor networks in big data environments [24], which achieves strong security services and performs the forward secrecy feature with a high level of efficiency.

This paper handles the above problem in the protocol [21] and designs an improved anonymous identity authentication and key agreement protocol suitable for WSNs. Specifically, the proposed protocol realizes two-way identity authentication among users, gateways, and sensor nodes based on the dynamic sequence and the shared secret value. In the process of authentication, the three parties negotiate a session key using dynamic random numbers. The session key is utilized for future secure communication, thus realizing the confidential transmission of sensitive data. Finally, the security and functionality analysis shows that the proposed protocol can meet multiple security objectives and resist diversified attacks.

1.2. Road map

The organization of the remaining paper is as follows. Section 2 introduces preliminaries used in this paper, such as hash function, XOR encryption, and BAN logic. The proposed protocol is illustrated in Section 3. Section 4 gives formal security proof using BAN logic. Before concluding the full paper in Section 6, Section 5 provides the performance analysis and compares our protocol

with other related works.

2. Preliminaries

2.1. Hash Function

The hash function is an irreversible operation that provides a way to map an arbitrary input string to a fixed output string. It is widely available for identity authentication and key agreement protocols [25, 26, 27]. The output of the hash function is called a hash value. The length of the hash value depends on the algorithm used, usually 128 to 256 bits. Moreover, the hash function has the following security features:

(1) *One-way function* A hash function is one-way if given a uniform y it is infeasible for a PPT adversary to find a value x such that $h(x) = y$.

(2) *Collision-resistant* A hash function is collision-resistant if given a uniform x it is infeasible for a PPT adversary to find $x' \neq x$ such that $h(x') \neq h(x)$.

(3) *Rapidity* A hash function can quickly calculate the hash value of a given message. Namely, it is easy to calculate $h(x)$ according to the known x, such as linear time.

(4) *Avalanche effect* The change of one bit in the input will cause more than half of the bits in the output to change.

2.2. XOR Encryption Algorithm

The XOR encryption algorithm is often used in the identity authentication protocol due to its simple and fast encryption and decryption. We suppose that $a = b \oplus c$. With any two of these parameters known (say, a and b), the third parameter (i.e., c) can be easily solved. If only one parameter (say, b) is known, it is impossible to solve the other two (i.e., a and c). Therefore, the XOR encryption is widely used in simple encryption operations to realize the private transmission of sensitive information [28].

2.3. BAN Logic

Burrow, Abadi, and Needham proposed the BAN logic in 1989. BAN logic is a widely used formal analysis method to analyze the correctness of authentication protocols and the security of key negotiation, playing an indispensable role in the security analysis of authentication protocols [29, 30, 31]. As a belief-based modal logic, the simplicity and practicality of BAN logic in protocol analysis is the main reason for the widespread interest. In the reasoning process of BAN logic, the beliefs of the subjects participating in the protocol are constantly changing as the message exchange. The first step of BAN logic is to carry out the "idealization" of the protocol, that is, to transform the interaction information in the protocol into the formula recognizable by BAN logic. Then, we make reasonable assumptions according to the specific situation. Finally, we apply the inference rules, idealized protocol, and assumptions to inferring whether or not the protocol achieves the expected goal. The BAN logic is based on the following premise.

(1) The subjects participating in the protocol are honest.

(2) The ciphertext block cannot be tampered with, nor can several small ciphertext blocks be used to form a new large ciphertext block.

(3) The two ciphertext blocks in a message are considered to have arrived at two separate times.

(4) Only the subject who holds the key can understand the ciphertext message.

(5) The ciphertext contains redundant information so that the decryptor can judge whether or not he has applied the correct key.

(6) The message contains redundant information so that the subject can judge whether or not the message comes from itself.

Table 1 lists the notations and their respective meanings related to the BAN logic. P and Q represent the subject variables, K denotes the key variables while X and Y depict the formula variables.

Next, we introduce BAN logic rules as follows:

Table 1: BAN logic notations and respective meanings

Notation	Meaning
$P\vert \equiv X$	P believes the truthfulness of X
$P \triangleleft X$	P sees X, i.e. P have received message X
$P\vert \sim X$	P once said X or P had sent message X
$P \Rightarrow X$	P has jurisdiction over X
$\sharp(X)$	X is fresh
$\{X\}_K$	X is encrypted under the key K
$P \overset{K}{\leftrightarrow} Q$	P and Q can communicate with the shared key K
(X, Y)	X or Y is a part of message (X, Y)

Rule1 Message-meaning rule. If P believes that K is the shared key between P and Q, and P receives the message $\{X\}_K$ encrypted by K, then P believes that Q has sent the message X.

$$\frac{P\vert \equiv P \overset{K}{\leftrightarrow} Q, P \triangleleft \{X\}_K}{P\vert \equiv Q\vert \sim X}$$

Rule2 Nonce verification rule. If P believes that X is new and P believes that Q has sent X, then P thinks that Q also believes in X.

$$\frac{P\vert \equiv \sharp(X), P\vert \equiv Q\vert \sim X}{P\vert \equiv Q\vert \equiv X}$$

Rule3 Jurisdiction rule. If P believes that Q has control over X, and P believes that Q believes in X, then P also believes in X.

$$\frac{P\vert \equiv Q \Rightarrow X, P\vert \equiv Q\vert \equiv X}{P\vert \equiv X}$$

Rule4 Belief rule. If P believes in X and Y, then P believes in (X, Y).

$$\frac{P\vert \equiv X, P\vert \equiv Y}{P\vert \equiv (X, Y)}$$

Rule5 Freshness-conjuncatenation rule. If P believes that X is fresh, then P believes that (X, Y) is fresh.

$$\frac{P\vert \equiv \sharp(X)}{P\vert \equiv \sharp(X, Y)}$$

Rule6 Session keys rule. If P believes that X is new and P believes that Q believes in X, then P believes that P and Q secretly negotiate the session key K, where X is the necessary parameter of K.

$$\frac{P| \equiv \sharp(X), P| \equiv Q| \equiv X}{P| \equiv P \overset{K}{\leftrightarrow} Q}$$

3. Anonymous Authentication and Key Agreement Protocol

In this section, we first narrate the classic architecture of the WSN and the functions of its components. Then, we construct the authentication model of the user-gateway-sensor node and provide the general procedure of authenticating. According to the identity authentication model, we design an anonymous identity authentication and key agreement protocol (AAKA) for the WSN, including five phases of the network setup, registration, pre-authentication, identity authentication and key agreement, and password update. Based on the dynamic sequence and the shared secret value, the AAKA protocol implements two-way authentication among users, gateway, and sensor nodes, preventing attackers from impersonating users, gateway, or sensor nodes and providing secure access to WSNs for legitimate users. During the authentication, the protocol applies dynamic random numbers to negotiating a session key among the three types of parties.

3.1. Architecture of WSN

Wireless sensor networks usually contain three communication parties: users, gateways, and sensor nodes. Figure 1 shows the architecture of the WSN.

The sensor node is deployed in the monitoring area, responsible for collecting and converting the data. The collected data is wirelessly transmitted to the gateway node after multi-hop. The gateway node, also known as the sink node, is responsible for connecting the wireless sensor network with the external network, sending monitoring instructions to the sensor nodes, and transmitting perceptual data to users. The user node is the server and computer terminal,

responsible for configuring and managing the wireless sensor network, publishing monitoring tasks, and receiving monitoring data returned by the gateway [32].

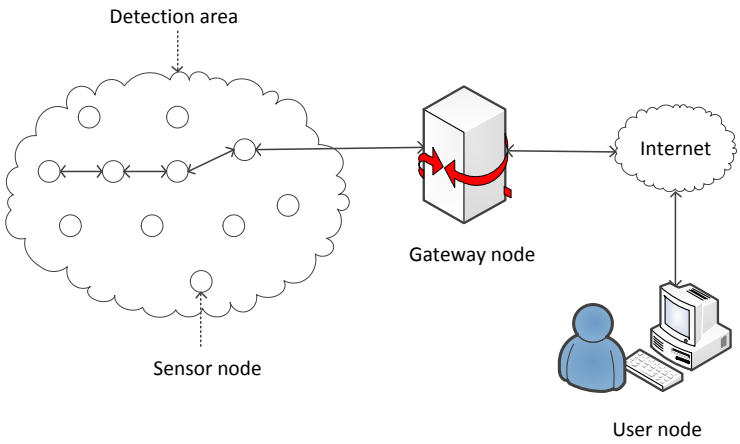

Figure 1: Architecture of wireless sensor network

3.2. Authentication Model

The authentication model of the proposed protocol includes three parties: users, gateway, and sensor nodes. The three parties first authenticate the identity legitimacy whenever users want to access the WSN. The overall authentication process is described as follows and shown in Figure 2.

The user sends an access request to the gateway, and the gateway authenticates the user's legitimacy based on the user's identity registration information such as dynamic sequence. After successful authentication, the gateway generates its identity authentication information using the secret value shared with the target sensor node. And the gateway sends its identity authentication information to the target sensor node. Similarly, the target sensor node authenticates the gateway's legitimacy based on the shared secret value and the information sent by the gateway. After successful authentication, the target sensor node calculates the session key using dynamic random numbers. And the target sensor node generates its authentication information utilizing the shared secret value

and the session key and sends it to the gateway. After the gateway obtains the session key according to the dynamic random numbers, it authenticates the legitimacy of the target sensor node. Providing the authentication is passed, the gateway generates its authentication information again based on the new dynamic sequence and sends it to the user. After the user obtains the session key using dynamic random numbers, he authenticates the gateway's legitimacy. If all three parties pass the identity legitimacy authentication, legitimate users can access the WSN to obtain data encrypted by the session key; otherwise, the session is terminated.

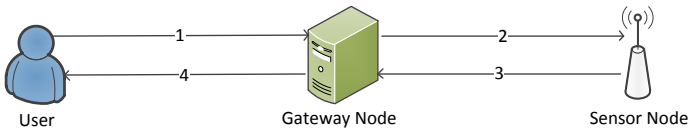

Figure 2: Authentication model

3.3. Network Setup

Before deploying the wireless sensor network, the gateway, sensor nodes, and smart cards perform the following initialization operations. Table 2 lists the notations and their respective meanings applied in this paper.

Step 1. The administrator stores some fundamental operation functions in the memory of smart card SC, gateway node GWN, and sensor node N_j, such as hash function $h()$, XOR operation, connection operation, and random number generator. Then, the administrator selects an identity ID_{SC} and a random number R_{SC} for the smart card SC, and stores $\{ID_{SC}, R_{SC}\}$ to the user's authentication table UT and SC's memory to make it a legal smart card. The user's authentication table UT is stored in GWN as shown in Table 3. Initially, the user's identity information XT and the dynamic sequence TS are empty. Next, the administrator assigns legitimate smart cards to users who demand to register.

Table 2: Notations and respective meanings

Notation	Meaning
U_i, GWN, N_j, SC	The i-th user, gateway node, the j-th sonser node, smart card
ID_i, ID_j, ID_{SC}	User, sensor, and smart cardr's identity
PW_i	U_i's password
X	Gateway's master key
SV_j	Secret value shared between gateway and sensor node
SK	Session key
T_i	Dynamic sequence number
TS_i	Time stamp
ΔT	Tolerable transmission delay
$R_U, R_{GWN}, R_N, R_{SC}$	Random number generated by user, gateway, sonser node, and smart card
$h()$	One-way hash function
$\|\|$	Connection operation
$\oplus$	XOR operation

Table 3: User's authentication table UT stored in GWN

SC's identity	SC's random number	U_i's identity information	Dynamic sequence
ID_{SC_1}	R_{SC_1}	XT_1	TS_1
ID_{SC_2}	R_{SC_2}	XT_2	TS_2
...	...	...	...
ID_{SC_m}	R_{SC_m}	XT_m	TS_m

Step 2. The gateway node GWN secretly stores the master key X, assigns an identity ID_j to the sensor node N_j, and calculates the secret value SV_j (see Eq 1) shared with N_j. Before deploying the sensor node N_j in the monitoring area, GWN stores $\{ID_j, SV_j\}$ in the sensor node's authentication table NT and N_j's memory to make it a legal sensor node. The sensor node's authentication table NT is stored in GWN as shown in Table 4. Finally, the administrator deploys the legal sensor nodes in the environment.

$$SV_j = h(ID_j||X) \tag{1}$$

Table 4: Sensor node's authentication table NT stored in GWN

N_j's identity ID_j	Secret value SV_j shared with N_j
ID_1	SV_1
ID_2	SV_2
...	...
ID_n	SV_n

3.4. Registration

In this phase, a new user can register itself at the gateway node. Firstly, the user selects the identity and password to generate a registration request and sends the request to the gateway. Next, the gateway generates the user's identity registration information and stores it in the authentication table UT. Finally, the user stores his identity registration information in the smart card. The specific process of this phase is detailed below and shown in Figure 3.

Step1. The user U_i generates his registration request.

Firstly, the user U_i inserts the smart card SC into a card reader for obtaining the SC's identity ID_{SC}. Then, U_i inputs his identity ID_i, password PW_i, and random number R_U and calculates the registration request RPW_i and REG_i as follows. Finally, U_i sends the registration request $\{ID_{SC}, RPW_i, REG_i\}$ to the gateway node GWN.

$$RPW_i = h(PW_i||R_U) \tag{2}$$

$$REG_i = ID_i \oplus (R_{SC}||RPW_i) \tag{3}$$

Step2. The gateway node GWN generates the user U_i's identity registration information.

GWN checks whether or not ID_{SC} exists in the authentication table UT after receiving the U_i's registration request $\{ID_{SC}, RPW_i, REG_i\}$. If not, GWN rejects U_i's registration request; otherwise, GWN retrievals the matching random number R_{SC} in the authentication table UT according to ID_{SC}. Then, GWN checks the total number (sum) of legitimate registration requests and access requests it handled and sets the dynamic sequence $TS_i = sum$. Next, GWN calculates U_i's identity registration information as follows according to the registration request. Finally, GWN sends the identity registration information $\{RSP_i\}$ to U_i and stores $\{XT_i, TS_i\}$ in the authentication table UT.

$$ID_i = REG_i \oplus h(R_{SC}||RPW_i) \tag{4}$$

$$US = h(ID_i||X) \tag{5}$$

$$UR = US \oplus h(ID_i||RPW_i) \tag{6}$$

$$SE = TS_i \oplus h(US||ID_i) \tag{7}$$

$$UV = h(ID_i||US||RPW_i) \oplus TS_i \tag{8}$$

$$RSP_i = h(ID_i||R_{SC}) \oplus (UR||SE||UV) \tag{9}$$

$$XT_i = h(X||TS_i) \oplus ID_i. \tag{10}$$

Step3. The user U_i stores his identity registration information in the smart card SC.

U_i first restores UR, SE, and UV as Eq 11 after receiving $\{RSP_i\}$. Then, U_i calculates RE using equation 12 according to the random number R_U, identity ID_i, and password PW_i. Finally, U_i stores the identity registration information $\{UR, SE, UV, RE\}$ into SC and deletes $\{ID_{SC}, R_{SC}\}$.

$$(UR||SE||UV) = RSP_i \oplus h(ID_i||R_{SC}) \tag{11}$$

$$RE = R_U \oplus h(ID_i||PW_i) \tag{12}$$

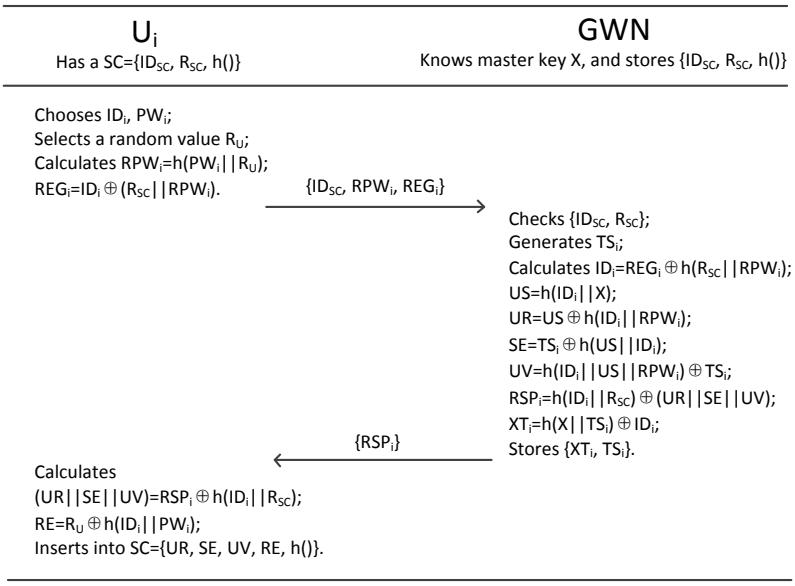

Figure 3: User registration phase

3.5. Pre-authentication

In this phase, the smart cards pre-authenticate the legitimacy of users when users want to access the WSN. Specifically, the smart card generates the user's access request S_1 and sends it to the gateway only if the identity and password inputted by the user pass the pre-authentication. At this phase, the pre-authentication function of the smart card reduces the communication and calculation overhead between the user and the gateway and effectively resists denial of service attacks. The specific process of this phase is detailed below and shown in Figure 4.

Step1. The smart card SC pre-authenticates the user U_i's validity according to the identity and password.

Firstly, U_i keys identity ID_i^*, password PW_i^*, and identity ID_j of the target sensor node N_j after inserting SC into a card reader. Then, SC calculates R_U^*, RPW_i^*, US^*, TS_i^*, and UV^* as follows according to the parameter RE, UR, SE and the user's identity ID_i^* and password PW_i^*. Finally, SC checks $UV^* \stackrel{?}{=} UV$. If equal, SC authenticates U_i as a legitimate user and continues the process; otherwise, SC terminates the session.

$$R_U^* = RE \oplus h(ID_i^* || PW_i^*) \tag{13}$$

$$RPW_i^* = h(PW_i^* || R_U^*) \tag{14}$$

$$US^* = UR \oplus h(ID_i^* || RPW_i^*) \tag{15}$$

$$TS_i^* = SE \oplus h(US^* || ID_i^*) \tag{16}$$

$$UV^* = h(ID_i^* || US^* || RPW_i^*) \oplus TS_i^* \tag{17}$$

Step2. The smart card SC generates the user U_i's access request.

SC chooses a time stamp T_1 and computes the access request information M_1 and M_2. Then, SC sends the access request $S_1 = \{M_1, M_2, T_1, TS_i\}$ to the gateway node GWN.

$$M_1 = (ID_j || R_U) \oplus h(US || ID_i || TS_i) \tag{18}$$

$$M_2 = h(ID_i || TS_i || US || T_1 || ID_j) \tag{19}$$

3.6. Identity Authentication and Key Agreement

In this phase, the AAKA protocol completes anonymous authentication and key agreement. Specifically, the user and the gateway implement a two-way authentication based on the dynamic sequence. The gateway and the sensor node achieve a two-way authentication based on the shared secret value. While authenticating, the three parties negotiate a session key based on their dynamic random numbers. The negotiated key ensures the confidential transmission of

later data over the public channel. The specific process of this phase is detailed below and shown in Figure 4.

Step1. The gateway authenticates the legitimacy of the user's identity and calculates its own authentication information.

After receiving the user U_i's access request S_1, the gateway GWN authenticates U_i's legitimacy according to S_1 and the user's identity registration information stored in table UT. If the authentication is successful, GWN calculates its identity authentication information S_2 and sends it to the target sensor node N_j. The specific process is as follows:

① *The gateway GWN authenticates the user U_i's legitimacy.* GWN first checks the validity of the time stamp T_1 after receiving the access request $S_1 = \{M_1, M_2, T_1, TS_i\}$. If the time difference between T_1 and the current time T_c exceeds the tolerable transmission delay ΔT, GWN will reject the request; otherwise, GWN retrieves the corresponding XT_i from table UT according to the dynamic sequence TS_i. Then, GWN calculates $ID_i^{'}$, $US^{'}$, $ID_j^{'}||R_U^{'}$, and $M_2^{'}$ as follows. Finally, GWN checks $M_2^{'} \stackrel{?}{=} M_2$. If equal, GWN authenticates U_i as a legitimate user and continues the process; otherwise, GWN terminates the session.

$$ID_i^{'} = h(X||TS_i) \oplus XT_i \tag{20}$$

$$US^{'} = h(ID_i^{'}||X) \tag{21}$$

$$ID_j^{'}||R_U^{'} = M_1 \oplus h(US^{'}||ID_i^{'}||TS_i) \tag{22}$$

$$M_2^{'} = h(ID_i^{'}||TS_i^{'}||US^{'}||T_1||ID_j^{'}) \tag{23}$$

② *After verifying the user U_i's legitimacy, the gateway GWN updates U_i's identity registration information in table UT.* Firstly, GWN chooses a timestamp T_2 and a random number R_{GWN}, checks the total number (sum^{new}) of the legitimate registration and access requests it handled, and sets the new dynamic sequence $TS_i^{new} = sum^{new}$. Then, GWN computes XT_i^{new} using Eq 24

and updates U_i's identity registration information as $\{XT_i^{new}, TS_i^{new}\}$ in table UT.

$$XT_i^{new} = h(X||TS_i^{new}) \oplus ID_i \tag{24}$$

③ *The gateway GWN calculates its identity authentication information* S_2 *and sends it to the target sensor node* N_j. First, GWN retrieves the corresponding secret value SV_j from table NT according to N_j's identity $ID_j^{'}$. Then, GWN calculates the authentication information c, M_3, and M_4 as follows. Finally, GWN sends the authentication information $S_2 = \{M_3, M_4, T_2\}$ to N_j.

$$c = TS_i \oplus TS_i^{new} \tag{25}$$

$$M_3 = (R_U^{'}||R_{GWN}) \oplus h(SV_j^{'}) \tag{26}$$

$$M_4 = h(ID_j^{'}||R_{GWN}||T_2||SV_j^{'}) \tag{27}$$

Step2. The sensor node authenticates the legitimacy of the gateway and calculates the session key as well as its identity authentication information.

After receiving the gateway GWN's authentication information S_2, the target sensor node N_j authenticates the legitimacy of GWN according to S_2 and the shared secret value SV_j. If the authentication is successful, N_j calculates a session key SK and its identity authentication information S_3. Then, N_j sends S_3 to GWN. The specific process is as follows:

① *The target sensor node* N_j *authenticates the gateway* GWN*'s legitimacy.* hlN_j first checks the validity of the time stamp T_2 after receiving GWN's authentication information $S_2 = \{M_3, M_4, T_2\}$. If T_2 is invalid, N_j will reject the session; otherwise, N_j computes $R_U^{''}||R_{GWN}^{'}$ and $M_4^{'}$ according to S_2 and the secret value SV_j shared with GWN. Then, N_j checks $M_4^{'} \stackrel{?}{=} M_4$. If equal, N_j believes in GWN's legitimacy and continues the process; otherwise, N_j terminates the session.

$$R_U^{''}||R_{GWN}^{'} = M_3 \oplus h(SV_j) \tag{28}$$

$$M_4^{'} = h(ID_j||R_{GWN}^{'}||T_2||SV_j) \tag{29}$$

② *After verifying the gateway GWN's legitimacy, the target sensor node* N_j *calculates a session key* SK *and its identity authentication information* S_3. Firstly, N_j chooses a timestamp T_3 and a random number R_N, and computes the session key SK using Eq 30 according to the random numbers $R_U^{''}, R_{GWN}^{'}, R_N$. Then, N_j computes its identity authentication information M_5 and M_6 as follows. Finally, N_j sends the authentication information $S_3 = \{M_5, M_6, T_3\}$ to GWN.

$$SK = h(R_U^{''}||R_{GWN}^{'}||R_N) \tag{30}$$

$$M_5 = R_N \oplus h(SV_j) \tag{31}$$

$$M_6 = h(SK||SV_j||R_N||T_3||ID_j) \tag{32}$$

Step3. The gateway authenticates the legitimacy of the sensor node identity and calculates the session key and authentication information.

Based on the authentication information S_3 of the target sensor node N_j, the gateway GWN computes the session key and authenticates N_j's legitimacy. If the authentication is successful, GWN calculates its identity authentication information S_4 and sends it to the user U_i. The specific process is as follows:

① *The gateway GWN computes the session key according to the authentication information* S_3 *of the target sensor node* N_j. GWN first checks the validity of the timestamp T_3 after receiving the authentication information $S_3 = \{M_5, M_6, T_3\}$. If T_3 is invalid, GWN will reject the session; otherwise, GWN restores N_j's random number $R_N^{'}$ according to the shared secret value $SV_j^{'}$ and M_5. Then, GWN computes the session key $SK^{'}$ according to $R_U^{'}, R_{GWN}, R_N^{'}$.

$$R_N^{'} = M_5 \oplus h(SV_j^{'}) \tag{33}$$

$$SK^{'} = h(R_U^{'}||R_{GWN}||R_N^{'}) \tag{34}$$

② *The gateway GWN authenticates the legitimacy of the target sensor node* N_j. Firstly, GWN computes $M_6^{'}$ as Eq 35 according to the session key $SK^{'}$ and the shared secret value $SV_j^{'}$. Then, GWN checks $M_6^{'} \stackrel{?}{=} M_6$. If equal, GWN authenticates N_j as a legitimate sensor node and continues the process; otherwise, GWN terminates the session.

$$M_6^{'} = h(SK^{'}||SV_j^{'}||R_N^{'}||T_3||ID_j^{'}) \tag{35}$$

③ *The gateway GWN calculates its identity authentication information* S_4 *after verifying the legitimacy of the sensor node* N_j. Firstly, GWN chooses a timestamp T_4 and computes its identity authentication information M_7, M_8, and M_9. Then, GWN sends its authentication information $S_4 = \{M_7, M_8, M_9, T_4\}$ to U_i.

$$M_7 = c \oplus US^{'} \tag{36}$$

$$M_8 = (R_{GWN}||R_N^{'}) \oplus h(R_U^{'}||US^{'}) \tag{37}$$

$$M_9 = h(SK^{'}||ID_i^{'}||TS_i^{new}||US^{'}||T_4) \tag{38}$$

Step4. The user calculates the session key and authenticates the legitimacy of the gateway.

Based on the authentication information S_4 of the gateway GWN, the user U_i computes the session key and authenticates the legitimacy of GWN. If the authentication is successful, the legitimate U_i can access WSN to obtain the required data. The session key is used to encrypt the sensitive data to guarantee confidential communication between users and sensor nodes. The specific process is as follows:

① *The user U_i computes the session key according to the authentication information S_4 of the gateway GWN.* U_i first checks the validity of the timestamp T_4 after receiving the authentication information $S_4 = \{M_7, M_8, M_9, T_4\}$. If T_4 is invalid, U_i will reject the session; otherwise, U_i performs the following to obtain the session key $SK^{''}$.

$$c^{'} = M_7 \oplus US \tag{39}$$

$$TS_i^{new^{'}} = c^{'} \oplus TS_i \tag{40}$$

$$R_{GWN}^{'}||R_N^{''} = M_8 \oplus h(R_U||US) \tag{41}$$

$$SK^{''} = h(R_U||R_{GWN}^{'}||R_N^{''}) \tag{42}$$

② *The user U_i authenticates the legitimacy of the gateway GWN.* Firstly, U_i computes $M_9^{'}$ according to the session key $SK^{''}$, identity ID_i, and the new dynamic sequence $TS_i^{new^{'}}$. Then, U_i checks $M_9^{'} \stackrel{?}{=} M_9$. If equal, U_i believes in GWN's legitimacy and continues the process; otherwise, U_i terminates the session.

$$M_9^{'} = h(SK^{''}||ID_i||TS_i^{new^{'}}||US||T_4) \tag{43}$$

③ *The user U_i updates the parameters stored in the smart card SC.* U_i selects a random number R_U^{new} and computes the new parameters $\{SE, RE, UR, UV\}$ as follows. Finally, the parameters in SC are replaced as $\{SE^{new}, RE^{new}, UR^{new}, UV^{new}, h()\}$.

$$SE^{new} = SE \oplus c^{'} \tag{44}$$

$$RE^{new} = RE \oplus R_U \oplus R_U^{new} \tag{45}$$

$$RPW_i^{new} = h(PW_i||R_U^{new}) \tag{46}$$

$$UR^{new} = US \oplus h(ID_i||RPW_i^{new}) \tag{47}$$

$$UV^{new} = h(ID_i||US||RPW_i^{new}) \oplus TS_i^{new} \tag{48}$$

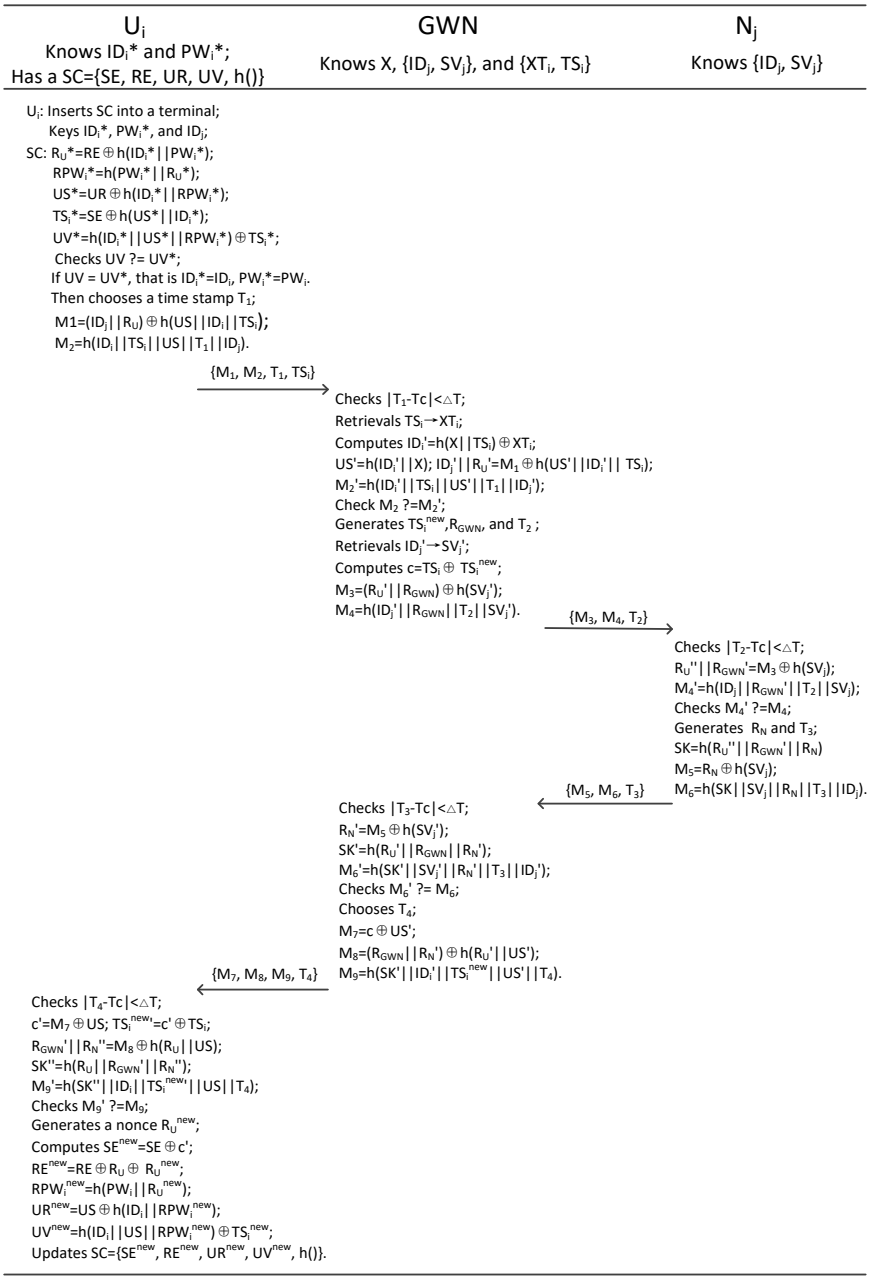

Figure 4: Identity authentication and key agreement phase

3.7. Password Update

Although the user's password is only in his possession, there is still the possibility of password leakage. Therefore, users need to change the password irregularly. The password update phase is only completed at the user terminal, without transmitting any bytes to the gateway, and does not involve the communication between the user and the gateway. Therefore, this phase provides users with a convenient password update operation. The specific process is detailed below and shown in Figure 5.

Step 1. The smart card SC pre-authenticates the user U_i's validity.

The user U_i inserts the smart card SC into the card reader and inputs his identity ID_i^* and password PW_i^*. The smart card SC pre-authenticates the user U_i's validity according to the identity ID_i^* and password PW_i^*. If the pre-authentication fails, SC rejects U_i's password update request; otherwise, U_i's enters a new password PW_i^{new}.

Step 2. The smart card SC updates the parameters.

The smart card SC updates its stored parameters as follows based on the new password PW_i^{new} entered by the user. Finally, SC updates the corresponding parameters as $\{SE, RE^{new}, UR^{new}, UV^{new}, h()\}$.

$$RPW_i^{new} = h(PW_i^{new}||R_U^*) \tag{49}$$

$$UR^{new} = US \oplus h(ID_i^*||RPW_i^{new}) \tag{50}$$

$$RE^{new} = R_U^* \oplus h(ID_i^*||PW_i^{new}) \tag{51}$$

$$UV^{new} = h(ID_i^*||US||RPW_i^{new}) \oplus TS_i^* \tag{52}$$

4. Security Proof

In this section, we carry out the formal security analysis of our protocol using the BAN logic. We prove that the AAKA protocol can achieve two-way authentication and negotiate session keys among users, gateways, and sensor

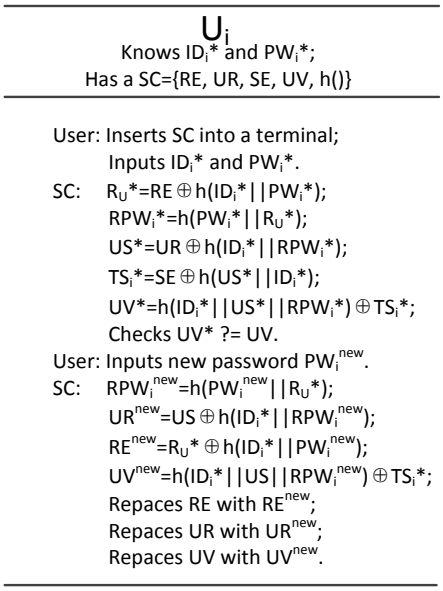

Figure 5: Password update phase

nodes through rigorous security analysis. First, we propose eight security goals that the AAKA protocol needs to meet in a network where users, gateways, and sensor nodes participate in authentication and key negotiation. Then, we use the BAN logic language to describe the initial state of the protocol and establish a set of initial hypotheses. Afterward, the actual messages are transformed into the formula of BAN logical. Finally, we use the inference rules to analyze whether the protocol satisfies the security goals.

4.1. Security Goals

We establish the following goals which the AAKA protocol should be satisfied from the analytic procedures of BAN logic.

$$Goal1 : GWN| \equiv U \overset{SK}{\leftrightarrow} GWN$$

$$Goal2 : GWN| \equiv U| \equiv (U \overset{SK}{\leftrightarrow} GWN)$$

$$Goal3 : U| \equiv GWN \overset{SK}{\leftrightarrow} U$$

$$Goal4: U| \equiv GWN| \equiv (GWN \overset{SK}{\leftrightarrow} U)$$

$$Goal5: N| \equiv GWN \overset{SK}{\leftrightarrow} N$$

$$Goal6: N| \equiv GWN| \equiv (GWN \overset{SK}{\leftrightarrow} N)$$

$$Goal7: GWN| \equiv N \overset{SK}{\leftrightarrow} GWN$$

$$Goal8: GWN| \equiv N| \equiv (N \overset{SK}{\leftrightarrow} GWN)$$

4.2. Canonical Form

This section idealizes the transmission information of the AAKA protocol, i.e., the actual messages of the protocol are represented using the BAN logic language.

(1) The access request $S_1 = \{M_1, M_2, T_1, TS_i\}$ sent from U to GWN is recorded as $(U \rightarrow GWN) : \{M_1, M_2, T_1, TS_i\}$, of which the canonical form is

$$S_1 : GWN \triangleleft \{M_1, M_2, T_1, TS_i\}$$

That is, the gateway GWN has received the message $\{M_1, M_2, T_1, TS_i\}$.

(2) The authentication message $S_2 = \{M_3, M_4, T_2\}$ sent from GWN to N is recorded as $(GWN \rightarrow N) : \{M_3, M_4, T_2\}$, of which the canonical form is

$$S_2 : N \triangleleft \{M_3, M_4, T_2\}$$

That is, the sensor node N has received the message $\{M_3, M_4, T_2\}$.

(3) The authentication message $S_3 = \{M_5, M_6, T_3\}$ sent from N to GWN is recorded as $(N \rightarrow GWN) : \{M_5, M_6, T_3\}$, of which the canonical form is

$$S_3 : GWN \triangleleft \{M_5, M_6, T_3\}$$

That is, the gateway GWN has received the message $\{M_5, M_6, T_3\}$.

(4) The authentication message $S_4 = \{M_7, M_8, M_9, T_4\}$ sent from GWN to U is recorded as $(GWN \rightarrow U) : \{M_7, M_8, M_9, T_4\}$, of which the canonical form is

$$S_4 : U \triangleleft \{M_7, M_8, M_9, T_4\}$$

That is, the user U has received the message $\{M_7, M_8, M_9, T_4\}$.

4.3. Assumptions

The following initial assumptions have been established to prove the security of the proposed protocol.

$$A1 : U| \equiv \sharp(R_U, R_{GWN}, R_N)$$

$$A2 : GWN| \equiv \sharp(R_U, R_{GWN}, R_N)$$

$$A3 : N| \equiv \sharp(R_U, R_{GWN}, R_N)$$

$$A4 : U| \equiv U \overset{US}{\leftrightarrow} GWN$$

$$A5 : GWN| \equiv GWN \overset{SV_j}{\leftrightarrow} N$$

$$A6 : N| \equiv N \overset{SV_j}{\leftrightarrow} GWN$$

$$A7 : GWN| \equiv GWN \overset{US}{\leftrightarrow} U$$

$$A8 : N \triangleleft \{R_{GWN}\}_{SV_j}$$

$$A9 : GWN \triangleleft \{R_N\}_{SV_j}$$

$$A10 : U \triangleleft \{R_{GWN}\}_{US}$$

$$A11 : GWN \triangleleft \{R_U\}_{US}$$

$$A12 : GWN \triangleleft \{U \overset{R_U}{\leftrightarrow} GWN\}_{US}$$

$$A13 : N \triangleleft \{GWN \overset{R_{GWN}}{\leftrightarrow} N\}_{SV_j}$$

$$A14 : GWN \triangleleft \{N \overset{R_N}{\leftrightarrow} GWN\}_{SV_j}$$

$$A15 : U \triangleleft \{GWN \overset{R_{GWN}}{\leftrightarrow} U\}_{US}$$

4.4. Security Proof

Based on the BAN logical inference rules, the initialization assumptions, and the idealized description of the transmitted information, the formal proof for our protocol is as follows.

(1) According to $A7$, $A12$ and $Rule1$, we get:

$$V1 : GWN| \equiv U| \sim (U \overset{R_U}{\leftrightarrow} GWN)$$

(2) According to $A2$, $V1$, and $Rule2$, we get:

$$V2^{'}: GWN| \equiv U| \equiv (U \overset{R_U}{\leftrightarrow} GWN)$$

where R_U is a necessary parameter of SK, so

$$V2: GWN| \equiv U| \equiv (U \overset{SK}{\leftrightarrow} GWN) \quad (Goal2)$$

(3) According to $A7$, S_1, $A11$, and $Rule1$, we get:

$$V3: GWN| \equiv U| \sim R_U$$

(4) According to $V3$, $A2$, $Rule2$, and $Rule5$, we get:

$$V4: GWN| \equiv U| \equiv R_U$$

(5) According to $V4$, $A2$, $A11$, and $Rule6$, we get:

$$V5: GWN| \equiv GWN \overset{SK}{\leftrightarrow} U) \quad (Goal1)$$

where R_U is a necessary parameter of SK.

(6)According to $A15$, $A4$, and $Rule1$, we get:

$$V6: U| \equiv GWN| \sim (GWN \overset{US}{\leftrightarrow} U)$$

(7) According to $A1$, $V6$, and $Rule2$, we get:

$$V7^{'}: U| \equiv GWN| \equiv (GWN \overset{R_{GWN}}{\leftrightarrow} U)$$

where R_{GWN} is a necessary parameter of SK, so

$$V7: U| \equiv GWN| \equiv (GWN \overset{SK}{\leftrightarrow} U) \quad (Goal4)$$

(8) According to $A10$, S_4, $A4$, and $Rule1$, we get:

$$V8: U| \equiv GWN| \sim R_{GWN}$$

(9) According to $A1$, $V8$, $Rule2$, and $Rule5$, we get:

$$V9: U| \equiv GWN| \equiv R_{GWN}$$

(10) According to $A1$, $V9$, $A10$, and $Rule6$, we get:

$$V10 : U| \equiv GWN \overset{SK}{\leftrightarrow} U \quad (Goal3)$$

where R_{GWN} is a necessary parameter of SK.

(11) According to $A6$, $A13$, and $Rule1$, we get:

$$V11 : N| \equiv GWN| \sim (GWN \overset{R_{GWN}}{\leftrightarrow} N)$$

(12) According to $A3$, $V11$, and $Rule2$, we get:

$$V12^{'} : N| \equiv GWN| \equiv (GWN \overset{R_{GWN}}{\leftrightarrow} N)$$

where R_{GWN} is a necessary parameter of SK, so

$$V12 : N| \equiv GWN| \equiv (GWN \overset{SK}{\leftrightarrow} N) \quad (Goal6)$$

(13) According to $A6$, S_2, $A8$, and $Rule1$, we get:

$$V13 : N| \equiv GWN| \sim R_{GWN}$$

(14) According to $A3$, $V13$, $Rule2$, and $Rule5$, we get:

$$V14 : N| \equiv GWN| \equiv R_{GWN}$$

(15) According to $A3$, $V14$, $A8$, and $Rule6$, we get:

$$V15 : N| \equiv GWN \overset{SK}{\leftrightarrow} N \quad (Goal5)$$

where R_{GWN} is a necessary parameter of SK.

(16) According to $A5$, $A14$, and $Rule1$, we get:

$$V16 : GWN| \equiv N| \sim (N \overset{R_N}{\leftrightarrow} GWN)$$

(17) According to $A2$, $V16$, and $Rule2$, we get:

$$V17^{'} : GWN| \equiv N| \equiv (N \overset{R_N}{\leftrightarrow} GWN)$$

where R_N is a necessary parameter of SK, so

$$V17 : GWN| \equiv N| \equiv (N \overset{SK}{\leftrightarrow} GWN) \quad (Goal8)$$

(18) According to $A5$, S_3, $A9$, and $Rule1$, we get:

$$V18 : GWN| \equiv N| \sim R_N$$

(19) According to $A2$, $V18$, $Rule2$, and $Rule5$, we get:

$$V19 : GWN| \equiv N| \equiv R_N$$

(20) According to $A2$, $V19$, $A9$, and $Rule6$, we get:

$$V20 : GWN| \equiv N \overset{SK}{\leftrightarrow} GWN \quad (Goal7)$$

where R_N is a necessary parameter of SK.

The formal analysis shows that the proposed protocol can satisfy the above security goals, provide two-way authentication, and secretly negotiate session keys.

5. Performance Analysis

This section discusses the performance of the AAKA protocol from functionality and overhead. We first analyze that our protocol resists most of the known attacks and achieves some ideal functional features. The overhead analysis shows the AAKA protocol has lower computational, storage, and communication overhead under the premise of security.

5.1. Performance Analysis

5.1.1. Users' Anonymity and Untraceability

User anonymity means that the user's real identity is shielded without knowing by any adversary. The AAKA protocol uses a one-way hash function to hide the user's real identity ID_i in the access request $\{M_1, M_2, T_1, TS_i\}$. After receiving the request, GWN calculates $TS_i \oplus h(X||TS_i)$ for obtaining ID_i through searching TS_i in the authentication table UT. Suppose an adversary intercepts the TS_i, the adversary cannot calculate the ID_i because he does not know the master key X. Thus, the user is anonymous in our protocol.

User untraceability means that the adversary cannot trace the users in different sessions through the communication messages transmitted on the common channel. Since each user accesses the gateway irregularly and new users register to the gateway, there is no connection between the dynamic sequence TS_i used in this session and TS_i^{new} used in the next session. Besides, the communication messages $\{M_1, M_2, T_1, TS_i\}$ are different since the user uses different random number R_U in each session. Therefore, the user is untraceability in our protocol.

5.1.2. Anonymity of Sensor Nodes

In our protocol, the real identity ID_j of the sensor node N_j does not explicitly exist in any communication messages, so the adversary cannot directly obtain the sensor's ID_j according to the communication messages on the public channel. Furthermore, the adversary cannot compute $(ID_j||R_U) = M_1 \oplus h(US||ID_i||TS_i)$ without knowing US and the user's real identity ID_i. Thus, the sensor node is anonymous in our protocol.

5.1.3. Forward Security and Backward Security

In the AAKA protocol, the session key is $SK = h(R_U||R_{GWN}||R_N)$, where R_U, R_{GWN}, and R_N are random numbers generated by U_i, GWN, and N_j, respectively. Since the session key does not depend on the GWN's master key X and the secret value SV_j shared between GWN and N_j, the disclosure of their secrets is not beneficial for the attacker to generate the session key. It is impossible to infer the session key of the previous session or the next session even if the adversary gets the current session's key since the random number used in each session is new. Hence, our protocol offers forward security and the backward security [33, 34].

5.1.4. Resisting Replay Attack

Replay attack means that an adversary retransmits the intercepted message to receiver to impersonate the legitimate user. In this protocol, users, gateway and sensor nodes first check the validity of the timestamp after receiving the message. The authentication will be terminated if the timestamp is invalid.

Suppose that an adversary replaces T_1 from $\{M_1, M_2, T_1, TS_i\}$ as T_A, where T_A is the current timestamp. Evidently, T_A would pass the freshness test, but the replay would fail since T_1 was used to compute $M_2 = h(ID_i||TS_i||US||T_1||ID_j)$. Therefore, M_2^* would be different from M_2 as T_A was used to compute $M_2^* = h(ID_i||TS_i||US||T_A||ID_j)$. Because of same reasons, the adversary could not replay the message $\{M_3, M_4, T_2\}$, $\{M_5, M_6, T_3\}$, and $\{M_7, M_8, M_9, T_4\}$. So, the proposed protocol withstands replay attack.

5.1.5. Resisting Stolen Smart Card Attack

Smart card is a tamper-resistant and counterfeit-resistant hardware. Legitimate users with smart cards confront the risk of smart card theft. When an adversary gets a legitimate user's smart card, he acquires the parameters stored in it. In our protocol, the smart card includes $\{RE, UR, SE, UV, h()\}$, where $RE = R_U \oplus h(ID_i||PW_i)$, $UR = US \oplus h(ID_i||RPW_i)$, $RPW_i = h(PW_i||R_U)$, $SE = TS_i \oplus h(US||ID_i)$, $UV = h(ID_i||US||RPW_i) \oplus TS_i$, $US = h(ID_i||X)$. For RE, the adversary has no idea of user's ID_i and PW_i, so he cannot retrieve R_U. For UR, it is difficult for the adversary to compute RPW_i without knowing PW_i and R_U. Therefore, the adversary cannot restore US by $UR \oplus h(ID_i||RPW_i)$. For SE and UV, as the results of one-way hash function, the adversary gets no information from $h(US||ID_i)$ and $h(ID_i||US||RPW_i)$ although TS_i is exposed in the channel. Furthermore, the adversary cannot compute US since X is only secretly known by GWN. Thus, the stolen smart card attack does not exist in our protocol.

5.1.6. Resisting Impersonation Attack

Impersonation attack means that an adversary acts as a legitimate user, gateway or sensor node by using the information in the smart card or the communication message intercepted on the public channel. In our protocol, US, ID_i, ID_j, and R_U are the necessary parameters to generate the access request M_1, M_2, T_1, TS_i. According to Section 5.1.1, the adversary cannot get ID_i and US through the access request without knowing the master key X.

Therefore, the adversary cannot get R_U and ID_j by computing $(ID_j||R_U) = M_1 \oplus h(US||ID_i||TS_i)$ without knowing ID_i and US. According to Section 5.1.5, the adversary cannot obtain ID_i and PW_i even if he has the legitimate user's smart card. Therefore, our protocol can prevent adversaries from impersonating users.

Besides, the master key X of GWN and the secret value SV_j shared between GWN and N_j are essential information to generate communication messages. Any adversary cannot impersonate as the gateway and sensor nodes since he does not know X and SV_j.

5.1.7. Resisting Off-line Password Guessing Attack

Off-line password guessing attack means that an adversary can access gateway by guessing the password of a legitimate user. The adversary can guess the user's password with the help of the access request sent by the user or the data stored in the smart card. In our protocol, the access request $\{M_1, M_2, T_1, TS_i\}$ does not contain any information about the password, so the adversary cannot guess the user's password through this way. Assuming that the adversary gets a smart card of a legitimate user U_i and guesses that U_i's identity is $ID^* \in \{0,1\}^n$ and password is $PW^* \in \{0,1\}^n$, the probability of $ID^* = ID_i$ and $PW^* = PW_i$ is $\frac{1}{2^{2n}}$, which is negligible. Therefore, the protocol can prevent off-line password guessing attack.

5.1.8. Resisting Insider Attack

Insider attack means that the internal attacker obtains legitimate user's identity or password by using the obtained registration information. In the registration phase, the user U_i just sends $\{ID_{SC}, RPW_i = h(PW_i||R_U), REG_i = ID_i \oplus h(R_{SC}||RPW_i)\}$ to GWN instead of directly transmitting ID_i and PW_i. The insiders of GWN cannot obtain U_i's ID_i and PW_i because of the irreversibility of $h()$ and the randomness of R_U, so the protocol can resist insider attacks.

5.2. Overhead Analysis

In this section, we compare the performance of the AAKA protocol with other related works in terms of computational, communication, and storage overhead. We only compare the overheads of identity authentication and key agreement phase since other phases are executed less frequently.

5.2.1. Computational Overhead

To facilitate the evaluation of computational costs, T_h and T_e are defined as the time cost of a hash function operation and an ECC point multiplication, respectively, where $T_e \approx 0.442$ and $T_h \approx 0.0004ms$ [26]. Table 5 provides the performance comparison of the AAKA protocol along with other related protocols. As shown in Table 5, Li's protocol has higher computational overhead because of using elliptic curve point multiplication; while the remaining three protocols have lower computational overhead since they only use the hash functions and XOR operations. Compared with the elliptic curve encryption algorithm, hash function and XOR encryption algorithm can achieve similar functions in less execution time. Therefore, lightweight encryption algorithms should be used to reduce the computational overhead of sensor nodes in resource-constrained wireless sensor networks.

Table 5: Computational overhead

Protocols	U_i	GWN	N_j	Toal overhead (ms)
Li et al.[17]	$8T_h + 2T_e$	$9T_h + 1T_e$	$4T_h$	1.3344
Devender et al.[19]	$8T_h$	$8T_h$	$4T_h$	0.0080
Zhang et al.[21]	$8T_h$	$7T_h$	$4T_h$	0.0076
AAKA protocol	$13T_h$	$8T_h$	$3T_h$	0.0096

5.2.2. Communication Overhead

The communication overhead refers to the sum of bits transmitted by users, gateway, and sensor nodes in the identity authentication and key agreement

phase. For convenience, we assume that the lengths of identity, password, timestamp, secret value, random numbers, and the output of hash functions are 128 bits, and the elliptic curve point's length is 160 bits. Table 6 lists the communication overhead of AAKA protocol along with other related protocols. As shown in Table 6, the total communication overhead in [17], [19], [21], and AAKA protocol are 1856 bits, 1536 bits, 2560 bits, and 1792 bits, respectively. Evidently, the AAKA's communication overhead is lower than that of Zhang et al. and Li et al. Compared with the protocol in [19], although the communication overhead of AAKA protocol is slightly higher, the sensor node of AAKA protocol has a lower computational overhead.

Table 6: Communication overhead

Protocols	U_i		GWN		N_j		Toal overhead (*bits*)
	Send	Receive	Send	Receive	Send	Receive	
Li et al.[17]	704	384	896	256	256	512	1856
Devender et al.[19]	512	384	768	768	256	384	1536
Zhang et al.[21]	640	896	1152	1024	768	640	2560
AAKA protocol	512	512	896	896	384	384	1792

5.2.3. Storage Overhead

Table 7 shows the storage overhead of the AAKA protocol along with other related protocols. As shown in Table 7, the total storage overhead in [17], [19], [21], and AAKA protocol are 1920 bits, 1408 bits, 1280 bits, and 1408 bits, respectively. Obviously, the AAKA and Devender's protocol have the same storage overhead, and lower than that of Li's. Compared with the literature [21], although the storage overhead of the AAKA protocol is slightly higher, it has a lower communication overhead.

5.2.4. Performance Comparisons

Figure 6 shows the comprehensive comparison of the four protocols in terms of computational, communication, and storage overhead. The closer the coordinate of the protocol is to the origin $(0, 0, 0)$, the smaller the protocol overhead

Table 7: Storage overhead

Protocols	SC	GWN	N_j	Toal overhead ($bits$)
Li et al.[17]	960	704	256	1920
Devender et al.[19]	640	512	256	1408
Zhang et al.[21]	512	384	384	1280
AAKA protocol	512	640	256	1408

and the better the overall performance. As shown in Figure 6, the computational and storage overhead of the AAKA protocol is similar to that proposed by Devender et al., but the communication overhead is slightly higher than that of Devender. The protocols proposed by Li et al. have significantly higher costs in computation, storage, and communication. Although the storage and computation overhead of the protocol proposed by Zhang et al. is low, the communication overhead is the highest. Generally speaking, the AAKA protocol has relatively good performance in balancing computation, storage, and communication overhead, and meets the low-cost requirements of WSNs. Therefore, the AAKA protocol is more suitable for WSNs.

6. Conclusion

To address the problems of identity counterfeiting and the security risks of transmitting data over public channels, we designed an Anonymous Authentication and Key Agreement protocol (AAKA). The AAKA protocol consists of five phases, namely network setup, registration, pre-authentication, anonymous authentication and key agreement, and password update. Based on dynamic sequences, shared secret values, and dynamic random numbers, the AAKA protocol enables bidirectional authentication and session key negotiation between users, gateways, and sensor nodes. Furthermore, we proved the AAKA's security using BAN logic. Performance analysis shows that our protocol can resist most current known attacks and obtains multiple security attributes. Compared with other authentication protocols, the AAKA protocol has lower computation,

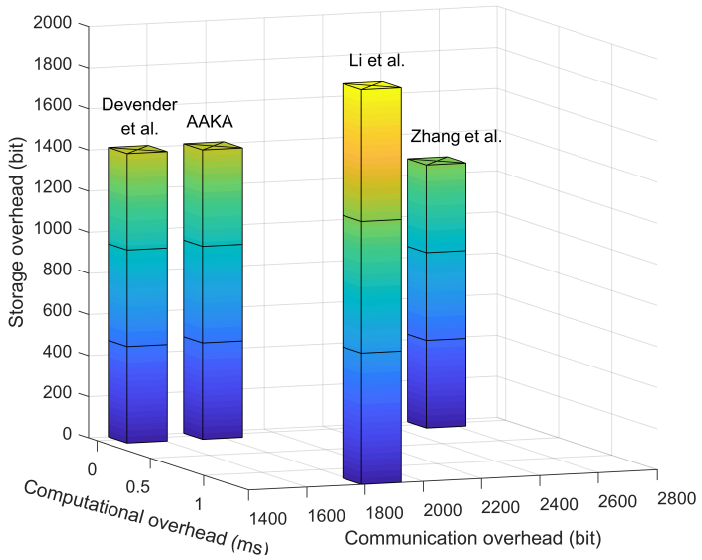

Figure 6: Performance comparison of four protocols

storage, and communication overhead besides the premise of security. Therefore, the AAKA protocol is suitable for secure access control and confidential data transmission in WSNs.

7. Acknowledgements

This work was supported by Natural Science Foundation of Shandong Province (Grant Number: ZR2019MF062)

References

[1] J. Liu, Y. Lai, S. Yang, L. Xu, Bilateral authentication protocol for wsn and certification by strand space model, Computing Science 46 (9) (2019) 169–175.

[2] X. Yu, F. Li, T. Li, N. Wu, H. Wang, H. Zhou, Trust-based secure directed diffusion routing protocol in wsn, Journal of Ambient Intelligence and Humanized Computing (2020) 1–13doi:10.1007/s12652-020-02638-z.

[3] C. Esposito, M. Ficco, B. B. Gupta, Blockchain-based authentication and authorization for smart city applications, Information Processing & Management 58 (2) (2021) 102468. doi:10.1016/j.ipm.2020.102468.

[4] F. Li, R. Ge, H. Zhou, Y. Wang, Z. Liu, X. Yu, Tesia: A trusted efficient service evaluation model in internet of things based on improved aggregation signature, Concurrency and Computation: Practice and Experience (2020) e5739doi:10.1002/cpe.5739.

[5] Y. Wang, G. Yang, T. Li, F. Li, Y. Tian, X. Yu, Belief and fairness: A secure two-party protocol toward the view of entropy for iot devices, Journal of Network and Computer Applications 161 (2020) 102641. doi:10.1016/j.jnca.2020.102641.

[6] A. Tewari, B. Gupta, Security, privacy and trust of different layers in internet-of-things (iots) framework, Future generation computer systems 108 (2020) 909–920. doi:10.1016/j.future.2018.04.027.

[7] C. L. Stergiou, K. E. Psannis, B. B. Gupta, Iot-based big data secure management in the fog over a 6g wireless network, IEEE Internet of Things Journaldoi:10.1109/JIOT.2020.3033131.

[8] F. Li, D. Wang, Y. Wang, X. Yu, N. Wu, J. Yu, H. Zhou, Wireless communications and mobile computing blockchain-based trust management in distributed internet of things, Wirel. Commun. Mob. Comput. 2020 (2020) 8864533:1–8864533:12. doi:10.1155/2020/8864533.

[9] A. Zhao, J. Li, J. Dong, L. Qi, Q. Zhang, N. Li, X. Wang, H. Zhou, Multimodal gait recognition for neurodegenerative diseases, IEEE Transactions on Cybernetics.

[10] Z. Chen, S. Qian, A security authentication scheme suitable for wireless sensor network, Computer Engineering 39 (7) (2013) 173–176. doi:10.3969/j.issn.1000-3428.2013.07.039.

[11] F. Li, Z. Liu, Y. Wang, N. Wu, J. Yu, C. zhi Gao, H. Zhou, Aitac: an identity-based traceable anonymous communication model, Journal of Ambient Intelligence and Humanized Computing (2020) 1–10.

[12] C. Cui, F. Li, T. Li, J. Yu, R. Ge, H. Liu, Research on direct anonymous attestation mechanism in enterprise information management, Enterprise Information Systems 15 (2021) 513–529.

[13] F. Xue, D. Wang, P. Cao, Y. Li, Cryptanalysis of two anonymous user authentication schemes for wireless sensor networks, Computer Applications 35 (12) (2015) 3424–3428.

[14] W. Li, , D. Wang, P. Wang, Insider attacks against multi-factor authentication protocols for wireless sensor networks, Journal of Software Vol.30Issue (8) (2019) 2375–2391.

[15] G. Qiu, X. Wang, Y. Zhang, Research on wsn identity authentication protocol based on hecc, Netinfo Security 26 (12) (2015) 54.

[16] M. Wang, B. Kang, D. Jing, An identity authentication and key agreement scheme for wireless sensor network, Computer Technology and Development (12) (2017) 104–108.

[17] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, K.-K. R. Choo, A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments, Journal of Network and Computer Applications 103 (2018) 194–204. doi:10.1016/j.jnca.2017.07.001.

[18] M. Chen, T.-F. Lee, J.-I. Pan, An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring, Sensors 19 (5) (2019) 1146.

[19] D. Kumar, S. Chand, B. Kumar, Cryptanalysis and improvement of an authentication protocol for wireless sensor networks applications like safety

monitoring in coal mines, Journal of Ambient Intelligence and Humanized Computing 10 (1) (2018) 1–20. doi:10.1007/s12652-018-0712-8.

[20] S. Kumari, H. Om, Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines, Computer Networks 104 (2016) 137–154. doi:10.1016/j.comnet.2016.05.007.

[21] X. Zhang, F. Wen, An novel anonymous user wsn authentication for internet of things, Soft Computing 23 (14) (2019) 5683–5691. doi:10.1007/s00500-018-3226-6.

[22] J. Lee, S. Yu, M. Kim, Y. Park, A. K. Das, On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks, IEEE Access 8 (2020) 107046–107062.

[23] H. A. N. Far, M. Bayat, A. K. Das, M. Fotouhi, S. M. Pournaghi, M. A. Doostari, Laptas: lightweight anonymous privacy-preserving three-factor authentication scheme for wsn-based iiot, Wireless Networks 27 (2021) 1389–1412.

[24] S. Nashwan, Aaa-wsn: Anonymous access authentication scheme for wireless sensor networks in big data environment, Egyptian Informatics Journal 22 (1) (2021) 15–26.

[25] F. Zhou, J. Xu, Lattices and Cryptography, Science Press, 2013.

[26] X. Liu, Research on authentication scheme for wireless sensor networks, Ph.D. thesis, Lanzhou University (2019).

[27] H. Ren, C. Zhao, C. Grebogi, One-way hash function based on delay-induced hyperchaos, International Journal of Bifurcation and Chaos 30 (02) (2020) 2050020. doi:10.1142/S0218127420500200.

[28] C. Yang, D. Wang, Property analysis of xor-based visual cryptography, IEEE transactions on circuits and systems for video technology 24 (2) (2014) 189–197. doi:10.1109/TCSVT.2013.2276708.

[29] S. Yang, Research on security protocol and ban logic analysis, Ph.D. thesis, Guizhou University (2007).

[30] Y. Lu, G. Xu, L. Li, Y. Yang, Anonymous three-factor authenticated key agreement for wireless sensor networks, Wireless Networks 25 (4) (2019) 1461–1475. doi:10.1007/s11276-017-1604-0.

[31] B. Yu, H. Li, Anonymous authentication key agreement scheme with pairing-based cryptography for home-based multi-sensor internet of things, International Journal of Distributed Sensor Networks 15 (9). doi:10.1177/1550147719879379.

[32] Z. Ren, G. Zhang, D. Lin, Z. Zhang, X. Zhao, Review on application of wsns, Transducer and Microsystem Technologies 037 (003) (2018) 1–2,10.

[33] F. Li, Z. Liu, T. Li, H. Ju, H. Wang, H. Zhou, Privacy-aware pki model with strong forward security, International Journal of Intelligent Systemsdoi:10.1002/int.22283.

[34] F. Li, Y. Wang, H. Ju, Y. Wang, Z. Wang, H. Zhou, Farpscusn: Fully anonymous routing protocol with self-healing capability in unstable sensor networks, Sensors (Basel, Switzerland) 20. doi:10.3390/s20226683.