sensors

MDPI

Article

An Energy Efficient Mutual Authentication and Key Agreement Scheme Preserving Anonymity for Wireless Sensor Networks

Yanrong Lu [1,2], Lixiang Li [1,2,*], Haipeng Peng [1,2] and Yixian Yang [1,2]

[1] Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; luyanrong1985@bupt.edu.cn (Y.L.); penghaipeng@bupt.edu.cn (H.P.); yxyang@bupt.edu.cn (Y.Y.)

[2] National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: lixiang@bupt.edu.cn; Tel.: +86-10-6228-2264

Academic Editor: Rongxing Lu

Received: 29 March 2016; Accepted: 1 June 2016; Published: 8 June 2016

Abstract: WSNs (Wireless sensor networks) are nowadays viewed as a vital portion of the IoTs (Internet of Things). Security is a significant issue in WSNs, especially in resource-constrained environments. AKA (Authentication and key agreement) enhances the security of WSNs against adversaries attempting to get sensitive sensor data. Various AKA schemes have been developed for verifying the legitimate users of a WSN. Firstly, we scrutinize Amin-Biswas's currently scheme and demonstrate the major security loopholes in their works. Next, we propose a lightweight AKA scheme, using symmetric key cryptography based on smart card, which is resilient against all well known security attacks. Furthermore, we prove the scheme accomplishes mutual handshake and session key agreement property securely between the participates involved under BAN (Burrows, Abadi and Needham) logic. Moreover, formal security analysis and simulations are also conducted using AVISPA(Automated Validation of Internet Security Protocols and Applications) to show that our scheme is secure against active and passive attacks. Additionally, performance analysis shows that our proposed scheme is secure and efficient to apply for resource-constrained WSNs.

Keywords: anonymity; mutual authentication; wireless sensor networks; smart card

1. Introduction

With the advancement of short range radio communication coupled with advances in miniaturization of computing devices, WSNs (Wireless sensor networks) have drawn continuing attention from both academia and industrial areas due to its deployment scalability, power consumption constraint and wide applications. Within the infrastructure of WSNs, privacy and security are the two major challenges since nodes are generally deployed in hostile environments thus making the nodes vulnerable to attacks. From this context, secure information exchange over an untrusted network is a widely discussed issue in WSNs. In order to allow remote authorized users to access reliable sensor nodes which have been verified as legitimate ones, mutual AKA (Authentication and key agreement) between communicating entities is required in the scheme design. An AKA scheme for WSNs is composed of three classes of entity: users, sensor nodes and a gateway node (GWN), and has registration, login, authentication and key agreement, and password change phases. To date, research in an efficient and robust user authentication and session key agreement mechanism has gained a great deal of attention. A number of AKA schemes are developed in an attempt to enhance the security of the WSNs in the literature [1–3]. Among different kinds of cryptographic primitives (RSA [4], ECC [5,6] Elgamal [7] *etc.*) utilized in AKA for WSNs, lower computational cost

scheme is even more admired owing to stringent constraints on limited computation capability, energy resources, storage and bandwidth of sensor nodes.

Wong *et al.* [8] released a hash function based AKA scheme for WSN, which sharply decreases computational load and makes the scheme adapt into a WSN environment. Nevertheless, as the scheme remains the lookup table of the registered user's private data in the GWN side, it was demonstrated to be defenseless to stolen-verifier attack [9]. Later on, Das [9] developed a better scheme in order to mitigate the security flaws over Wong *et al.* The scheme concentrates on temporal credential and timestamp under defense mechanism aiming at preventing DoS attack efficiently while maintaining lightweight style. Unfortunately, the scheme was analyzed by many researchers and the results illustrated that it had still some drawbacks and flaws [10–14], such as incapability of achieving mutual authentication, notwithstanding node compromise attack, failing to provide the user password update securely. With the hope of amending aforementioned security weaknesses, several authors developed modifications on Das's scheme but at the cost of increasing computational complexity [10,11,14]. Motivated by the thought of achieving better security and efficiency, Das *et al.*'s [15] built an efficient password based user AKA using only the hash function which encompasses the power of smart cards. They justified that compromise of a cluster head is free from node capture attacks. Their scheme allows only updating the password of the user locally without the help of the base station. Further, they evaluated their scheme in support of using no high computation except from the nominal task of assigning identity bits commitments and justified low memory requirement due to small size of identity bits commitment. Nevertheless, Turkanović [16], Wang-Wang [17] and Li [18] came across some additional problems in Das's scheme, like non resistance to insider, stolen-verifier and node capture attacks. After that, Xue *et al.* [19] proposed a temporal-credential-based lightweight and resource user AKA scheme for WSNs using hash and XOR computations. In their scheme, the gateway node issues a temporal credential to each user and sensor node with the help of password-based authentication. Unfortunately, He *et al.* [20] was later remarked that the scheme of Xue *et al.* is imperfection and not applicable for practical implementation, due to some design defects and susceptibility to some attacks. Most recently, Turkanović *et al.* [21] proposed a lightweight user authentication scheme for WSN based only on hash and Xor computations that tend to save both computation and communication resources. Such cryptographic techniques scheme launched with a claim of achieving the basic security attributes as well as thwarting many attacks along with better complexities. The AKA scheme drew considerable attention but was subsequently on determined insecure and susceptible. The authors of [22–24] studied the vulnerability of the scheme [21] that incurs several security drawbacks and not applicable for practical implementation in the presence of an attacker who can mount a smart card theft attack. Motivated by the thought of preventing the security threats of scheme [21], Amin-Biswas [24] developed a modified version of the hash and Xor operations in order to appropriate for resource constrained environments. The authors addressed both security and efficiency, claimed that their designs possess many attractive features in which the system contains multiple gateway nodes. However, problems related to the leakage of the session short-term secrets accidentally are the fatal pitfalls of such scheme. Our contribution is motivated by the above facts.

2. Review of Amin-Biswas's Scheme

This section briefly reviews Amin-Biswas's scheme, which consists of system setup phase, user and sensor node registration phases, login phase, authentication phase (Figure 1), password update phase and dynamic node addition phase. Moreover, their scheme is composed of three entities: user, gateway node, and sensor node. For convenience of description, Table 1 shows the notations used in Amin-Biswas's scheme.

Table 1. Notations.

Symbol	Description
U_i	User
GWN	Gateway node
SN_j	Sensor node
$HGWN$	Home gateway node
ID_i/PW_i	Identity/Password of U_i
TID_i	Random identity of U_i generated by GWN for authentication
ID_{SN_j}	Identity of SN_j
X_k	Secret key of GWN
ΔT	Constant transmission time
T_i	Timestamp
r/r_i	Random numbers of U_i
$h(\cdot)$	One-way hash function
$\oplus$	Xor operation

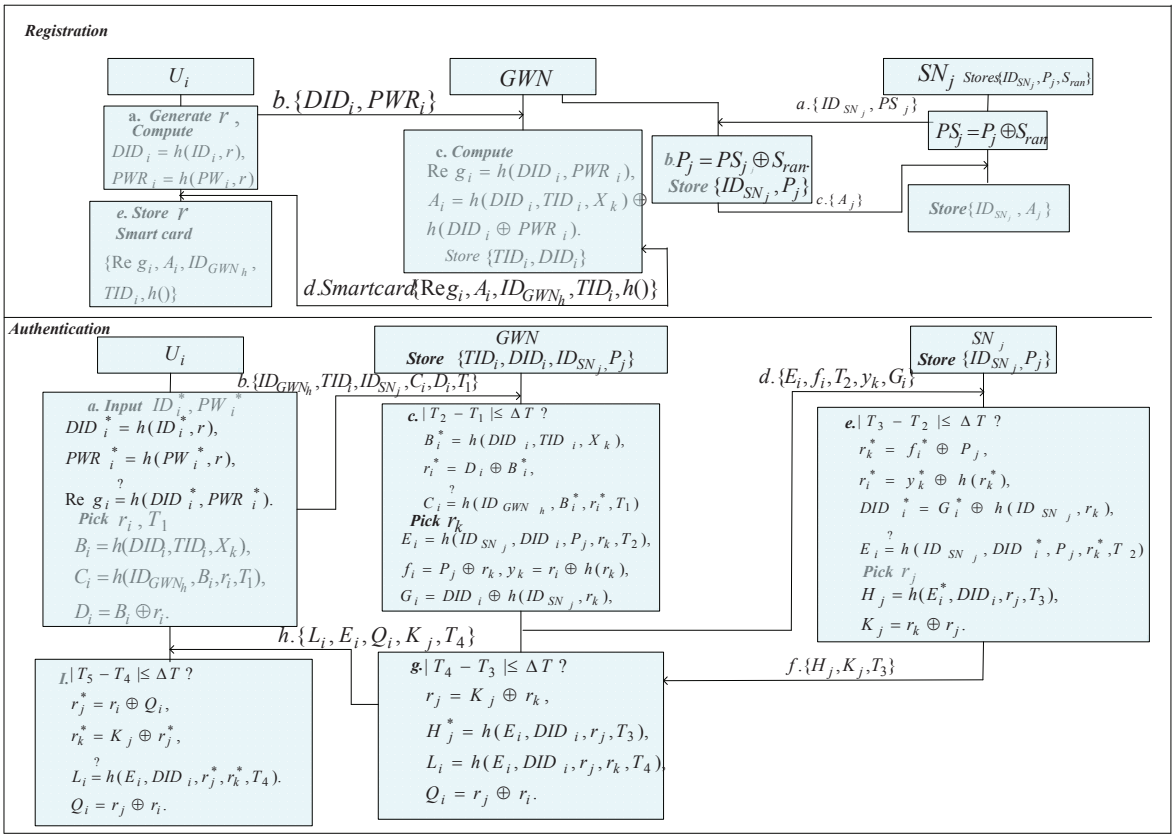

Figure 1. Mutual authentication and key agreement of Amin-Biswas's scheme.

2.1. System Setup

The system administrator deploys each SN_j which stores $\{ID_{SN_j}, P_j, S_{ran}\}$ into its memory, where $P_j = h(ID_{SN_j}, S_{ran})$, S_{ran} is a random number and is known to all the GWNs and maintains it securely.

2.2. Sensor Node Registration

Step 1: SN_j sends $\{ID_{SN_j}, PS_j\}$ to the nearby GWN, where $PS_j = P_j \oplus S_{ran}$.

Step 2: The GWN stores $\{ID_{SN_j}, P_j\}$, where $P_j = PS_j \oplus S_{ran}$. After that, the GWN sends a confirmation message to each sensor node.

Step 3: Upon receiving the confirmation message from the GWN, each SN_j destroys S_{ran} from the memory.

2.3. User Registration

Step 1: The new user U_i computes $DID_i = h(ID_i, r), PWR_i = h(PW_i, r)$ and sends $\{DID_i, PWR_i\}$ to the $HGWN$ via private channel, where r is a nonce, ID_i is the identity and PW_i is the password of U_i.

Step 2: The $HGWN$ computes $Reg_i = h(DID_i, PWR_i), A_i = h(DID_i, TID_i, X_k) \oplus h(DID_i \oplus PWR_i)$, where TID_i is a random identity and X_k is the $HGWN$'s long term secret key.

Step 3: The $HGWN$ issues a smart card which contains $\{Reg_i, A_i, ID_{GWN_h}, TID_i, h()\}$ and sends it to U_i. Further, the $HGWN$ stores $\{TID_i, DID_i\}$ in its memory.

Step 4: When receiving the smart card, U_i stores $\{r\}$ in the smart card.

2.4. Login and Authentication

Step 1: U_i inserts the smart card and inputs identity ID_i and password PW_i to the card reader. After that, the card reader computes $DID_i = h(ID_i, r), PWR_i = h(PW_i, r)$ and checks whether $h(DID_i, PWR_i) \stackrel{?}{=} Reg_i$.

Step 2: If it matches, the card reader computes $B_i = h(DID_i, TID_i, X_k) = A_i \oplus h(DID_i \oplus PWR_i)$, $C_i = h(ID_{GWN_h}, B_i, r_i, T_1), D_i = B_i \oplus r_i$ and sends a login message $M_1 = \{ID_{GWN_h}, TID_i, ID_{SN_j}, C_i, D_i, T_1\}$ to the $HGWN$ by public channel.

Step 3: When receiving the message M_1, the $HGWN$ first checks whether the received timestamp T_1 is within the valid time period, the $HGWN$ computes $B_i = h(DID_i, TID_i, X_k), r_i = D_i \oplus B_i$, the $HGWN$ extracts DID_i from the database using TID_i. Next, the $HGWN$ checks whether $h(ID_{GWN_h}, B_i, r_i, T_1) \stackrel{?}{=} C_i$. If it holds, the $HGWN$ computes $E_i = h(ID_{SN_j}, DID_i, P_j, r_k, T_2)$, $f_i = P_j \oplus r_k, y_k = r_i \oplus h(r_k), G_i = DID_i \oplus h(ID_{SN_j}, r_k)$ and sends $M_2 = \{E_i, f_i, G_i, y_k, T_2\}$ to the the sensor node SN_j via public channel.

Step 4: After receiving the message M_2, SN_j checks whether $|T_3 - T_2| \leq \Delta T$. If it holds, SN_j computes $r_k = f_i \oplus P_j, r_i = y_k \oplus h(r_k), DID_i = G_i \oplus h(ID_{SN_j}, r_k)$ and checks whether $h(ID_{SN_j}, DID_i, P_j, r_k, T_2) \stackrel{?}{=} E_i$. If it matches, SN_j computes $H_j = h(E_i, DID_i, r_j, T_3), K_j = r_k \oplus r_j$ and sends $M_3 = \{H_j, K_j, T_3\}$ to the $HGWN$ via public channel.

Step 5: Upon receiving the message M_3, the $HGWN$ first checks the timestamp validity, *i.e.*, $|T_4 - T_3| \leq T$, where T_4 is the current timestamp. The $HGWN$ computes $r_j = K_j \oplus r_k$, $H_j = h(E_i, DID_i, r_j, T_3)$. If it is true, the $HGWN$ computes $L_i = h(E_i, DID_i, r_j, r_k, T_4), Q_i = r_j \oplus r_i$ and sends $M_4 = \{L_i, E_i, Q_i, K_j, T_4\}$ to the U_i via public channel.

Step 6: After receiving the message M_4, U_i checks whether the received timestamp is within the valid time intervals. If it holds, U_i extracts $r_j = r_i \oplus Q_i, r_k = K_j \oplus r_j, L_i = h(E_i, DID_i, r_j, r_k, T_4)$. If it is true, U_i confirms the authenticity of SN_j and computes $SK = h(DID_i, r_i, r_j, r_k)$ between the entities involved in the system.

2.5. Dynamic Node Addition

According to the system setup phase, the system administrator deploys the new sensor node over the target region and the deployed sensor node executes sensor node registration phase to the nearby GWN.

2.6. Password Update

Step 1: A user keys his password PW_i, the card reader computes $\alpha_i = h(DID_i, TID_i, X_k) = A_i \oplus h(DID_i \oplus PWR_i)$ and then computes $PWR_i^{new} = h(PW_i, r), Reg_i^{new} = h(DID_i, PWR_i^{new})$, $A_i^{new} = \alpha_i \oplus h(DID_i \oplus PWR_i^{new})$.

Step 2: The card reader stores the new computed values $\{Reg_i^{new}, A_i^{new}\}$ instead of the old values $\{Reg_i, A_i\}$.

3. Security Analysis of Amin-Biswas's Scheme

Although Amin-Biswas claimed that their scheme achieves several security requirements including mutual authentication, user anonymity and resilience against some attacks. Unfortunately, we found that there was still something security vulnerability in Amin-Biswas's scheme.

Known Session-Specific Temporary Information Attack

Cheng *et al.* [25] has demonstrated that the exposure of session temporary information accidentally should not compromise the secrecy of generated session key. However, we will demonstrate that Amin-Biswas's scheme contraries to this security property which is necessary for a good or an ideal authentication scheme [26]. Without loss of generality, we assume that a temporary information r_i is compromised by an adversary unintentionally, which may allow the adversary to frame the session key effortlessly and even more acquire the legitimate user's sensitive data by means of monitoring the transmitted data in the communication. To illustrate the process concretely, you can look at an attack in the next few steps (Figure 2).

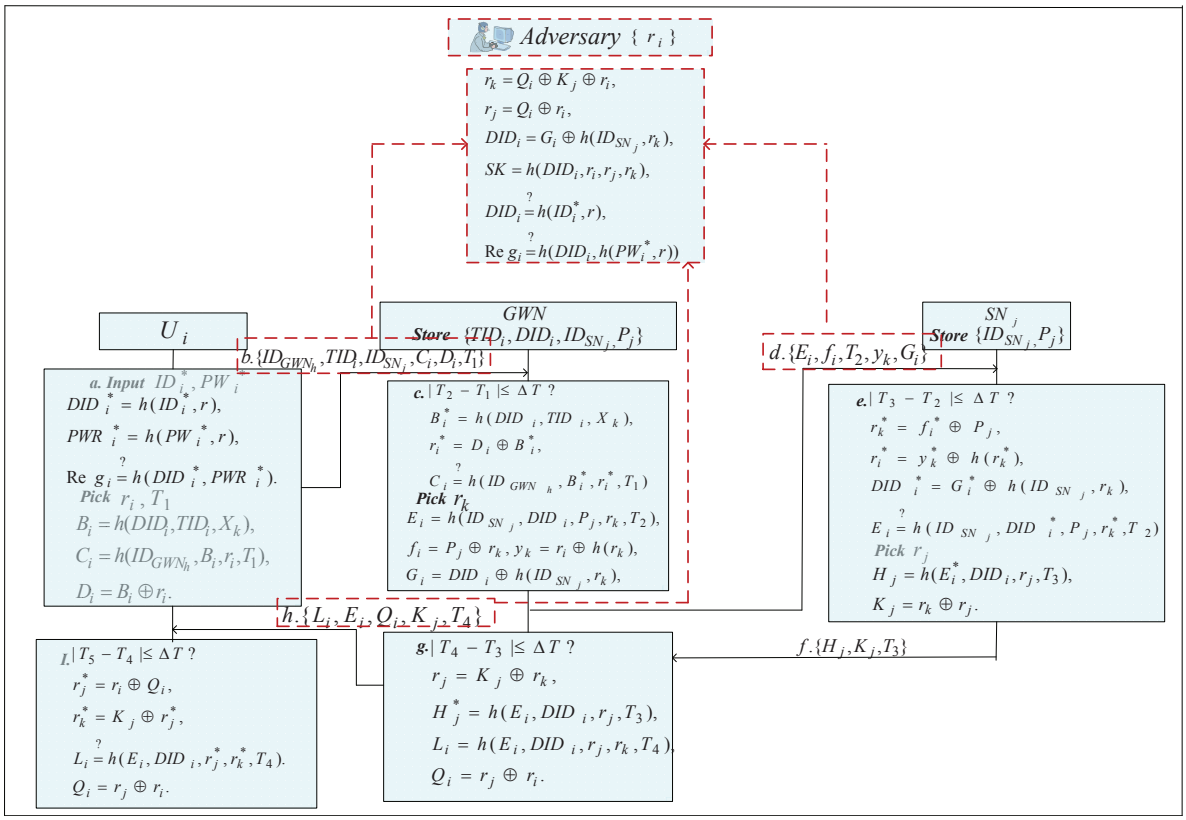

Figure 2. Known session-specific temporary information attack on Amin-Biswas's schem.

Step 1: The adversary could extract the session ephemeral secrets r_k and r_j from the results of $Q_i \oplus K_j \oplus r_i$ and $Q_i \oplus r_i$, where Q_i, K_j are the known parameters stemmed from the transferred message $M_4 : \{L_i, E_i, Q_i, K_j, T_4\}$.

Step 2: Based on the derived the session immediate secret r_k, the adversary has ability to retrieve another important parameter DID_i by computing $G_i \oplus h(ID_{SN_j}, r_k)$, where G_i is also obtained through the transmitted messages $M_1 : \{ID_{GWN_h}, TID_i, ID_{SN_j}, C_i, D_i, T_1\}$ and $M_2 : \{E_i, f_i, G_i, y_k, T_2\}$.

Step 3: The adversary could compute the session key $SK = h(DID_i, r_i, r_j, r_k)$ with all those derived data. Not only that, the adversary could easily guess the user's identity ID_i by attempting to check whether $DID_i \stackrel{?}{=} h(ID_i^*, r)$ until making the equation true, where ID_i^* is a candidate identity and r is extracted with a stolen smart card. The adversary is further capable of retrieving the user's password PW_i on the strength of the extracted secrets $\{Reg_i\}$ by checking $Reg_i \stackrel{?}{=} h(DID_i, h(PW_i^*, r))$ from the legal user's smart card. The aforementioned cryptanalysis is based on the concrete fact that identity and passwords are low-entropy keys [27,28]. As a result, the adversary succeeds to get the user's identity ID_i and the user's password PW_i.

Step 4: The above analysis reveals that, all those information leaks allow the adversary to impersonate as a legitimate user to login the GWN and access the real-time information from sensor nodes. In other words, our analysis demonstrates that their scheme can be free from known session-specific temporary information attack, thereby Amin-Biswas's scheme is completely insecure.

4. Proposed Improved Scheme

This section will describe our proposed anonymity-preserving AKA scheme in detail. The proposed AKA scheme conceals the user's real identity in the encryption algorithm along with the hash of random identity and secret key as the symmetric key. The messages, which are transmitted in public channel, are the results of the hash or the encryption, thus avoiding the risk by intercepting the communication channel to acquire the plaintext directly. In order to conquer the known session-specific temporary information attack, each communicate entity only knows the xor results of the others' generated random numbers in computing the session key. The proposed AKA scheme inherits Amin-Biswas's scheme aiming at cope with the loopholes of the aforementioned security drawbacks of their scheme. Based on the previous analysis, the functionality of the proposed scheme has been greatly improved with a slight higher computation cost due to the symmetric cryptographic algorithm. Our proposed AKA scheme has five phases: User registration; Sensor node registration; Login; Authentication and key agreement (Figure 3); Password change. We will introduce them as follows.

4.1. User Registration

Step 1: A new user U_i chooses his identity ID_i and password PW_i, then he sends his registration request message $\{ID_i, h(PW_i, r)\}$ to the gateway node GWN, where r is a random number.

Step 2: Upon receipt of the message, GWN computes $A_i = h(h(ID_i), h(PW_i, r)), B_i = h(TID_i, X_k) \oplus h(PW_i, r), C_i = h(ID_i, X_k) \oplus h(h(ID_i) \oplus h(PW_i, r))$. Next, GWN issues a smart card for each user after storing $\{A_i, B_i, C_i\}$ into the memory of smart card and thus sends back it to U_i. At last, GWN stores $\{TID_i\}$ in its memory.

Step 3: After receiving the smart card, U_i adds r to the smart card.

4.2. Sensor Node Registration

Step 1: The sensor node SN_j transmits its identity ID_{SN_j} to GWN.

Step 2: GWN computes $A_j = h(ID_{SN_j} \oplus S_{ran})$ and returns it to SN_j after storing $\{ID_{SN_j}, A_j\}$ into its memory.

Step 3: When receiving the message from GWN, SN_j also keeps them securely.

4.3. Login

When a registered user U_i desires the WSNs services, he needs to be prepare his personal information along with the smart card. The following procedure are required to be done by U_i:

Step 1: U_i enters his identity ID_i and password PW_i into the smart card after inserting the smart card into the mobile device. The smart card computes $h(h(ID_i), h(PW_i, r))$ and checks whether it is equal to A_i. If it holds, U_i is considered as a legal user.

Step 2: The card reader derives $h(TID_i, X_k)$ and $h(ID_i, X_k)$ by computing $B_i \oplus h(PW_i, r)$ and $C_i \oplus h(h(ID_i) \oplus h(PW_i, r))$, respectively. Based on the two values, the card reader computes D_i by encrypting the information $\{ID_i, T_1, TID_i, r_i\}$ with the derived $h(TID_i, X_k)$ and computes E_i by putting the information $\{h(ID_i, X_k), r_i, T_1\}$ into the hash function, where T_1 is the current timestamp at user side and r_i is a random number. Next, the card reader sends a login message $\{D_i, E_i\}$ to GWN.

Step 3: Upon receiving the login message, GWN decrypts D_i by the symmetric key $h(TID_i, X_k)$ to retrieve $\{ID_i, T_1, r_i\}$. Next, GWN checks whether $|T_2 - T_1| \leq \Delta T$, where T_2 is the current timestamp at GWN side. If it is valid, GWN verifies $h(h(ID_i, X_k), r_i, T_1) \stackrel{?}{=} E_i$. The validation of E_i ensures U_i is a legitimate user. Subsequently, GWN picks a random number r_k and computes $F_i = Enc_{h(ID_{SN_j} \oplus S_{ran})}(r_k \oplus r_i, TID_i, T_1, T_2)$, $G_i = h(TID_i, ID_{SN_j}, h(ID_{SN_j} \oplus S_{ran}), ID_{GWN}, T_2, r_k \oplus r_i)$. Next, GWN sends the message $\{F_i, G_i\}$ to SN_j.

Step 4: When receiving the message from GWN, SN_j decrypts F_i using the symmetric key $h(ID_{SN_j} \oplus S_{ran})$ to derive $\{r_k \oplus r_i, TID_i, T_1, T_2\}$. And then, SN_j checks the timestamp T_2 is within a permissible temporal interval. Next, SN_j computes $h(ID_{SN_j}, TID_i, ID_{GWN}, h(ID_{SN_j} \oplus S_{ran}), T_2, r_k \oplus r_i)$ and checks whether it matches with the received G_i. It it holds, SN_j computes $SK = h(r_k \oplus r_i \oplus r_j, T_1, T_2, T_3)$, $H_i = Enc_{h(ID_{SN_j} \oplus S_{ran})}(r_j, T_3, r_k \oplus r_i)$, $I_i = h(ID_{SN_j}, TID_i, T_3, SK)$. Finally, SN_j transmits the message $\{H_i, I_i\}$ to GWN.

Step 5: After receiving the message from SN_j, GWN also needs to decrypt the received H_i to derive $\{r_j, T_3, r_k \oplus r_i\}$. Upon retrieving T_3, GWN verifies whether T_3 is a valid timestamp. If it is valid, GWN computes $SK = h(r_k \oplus r_i \oplus r_j, T_1, T_2, T_3)$ and checks whether $h(ID_{SN_j}, TID_i, T_3, SK) \stackrel{?}{=} I_i$. If it is correct, GWN computes $J_i = Enc_{h(ID_i, X_k)}(r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4)$ and $K_i = h(SK, T_4, h(TID_i, X_k))$, where T_4 is the current timestamp at GWN side. Next, GWN sends the message $\{J_i, K_i\}$ to U_i.

Step 6: Once receiving the message from GWN, U_i derives $\{r_j \oplus r_k, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4\}$ by decrypting J_i using the symmetric key $h(ID_i, X_k)$. U_i then checks whether T_4 is fresh. The freshness of T_4 is verified, U_i proceeds to compute the session key $SK = h(r_k \oplus r_i \oplus r_j, T_1, T_2, T_3)$ and examine whether $h(SK, T_4, h(TID_i, X_k))$ is equivalent to the received K_i. If the equation is true, the handshake among three-party is successful, and they negotiate the session key SK with each other. The establishment of the session key is considered to be encrypted the following packs in their communication channel.

4.4. Password Change

When a user attempts to update his password into a new one, he needs to execute the following steps:

Step 1: The user initially inserts the smart card into the card reader and inputs his identity ID_i and old password PW_i. Next, the card reader computes $h(h(ID_i), h(PW_i, r))$ and checks whether it is equal to A_i. If it holds, the user is considered as a legal one. And thus, the card reader asks the user to key a new password.

Step 2: After keying the new password, the card reader computes $A_i^* = h(h(ID_i), h(PW_i, r))$, $B_i^* = B_i \oplus h(PW_i, r) \oplus h(PW_i^*, r)$ and $C_i^* = C_i \oplus h(h(ID_i) \oplus h(PW_i, r)) \oplus h(h(ID_i) \oplus h(PW_i^*, r))$. The card reader replaces $\{A_i, B_i, C_i\}$ with $\{A_i^*, B_i^*, C_i^*\}$.

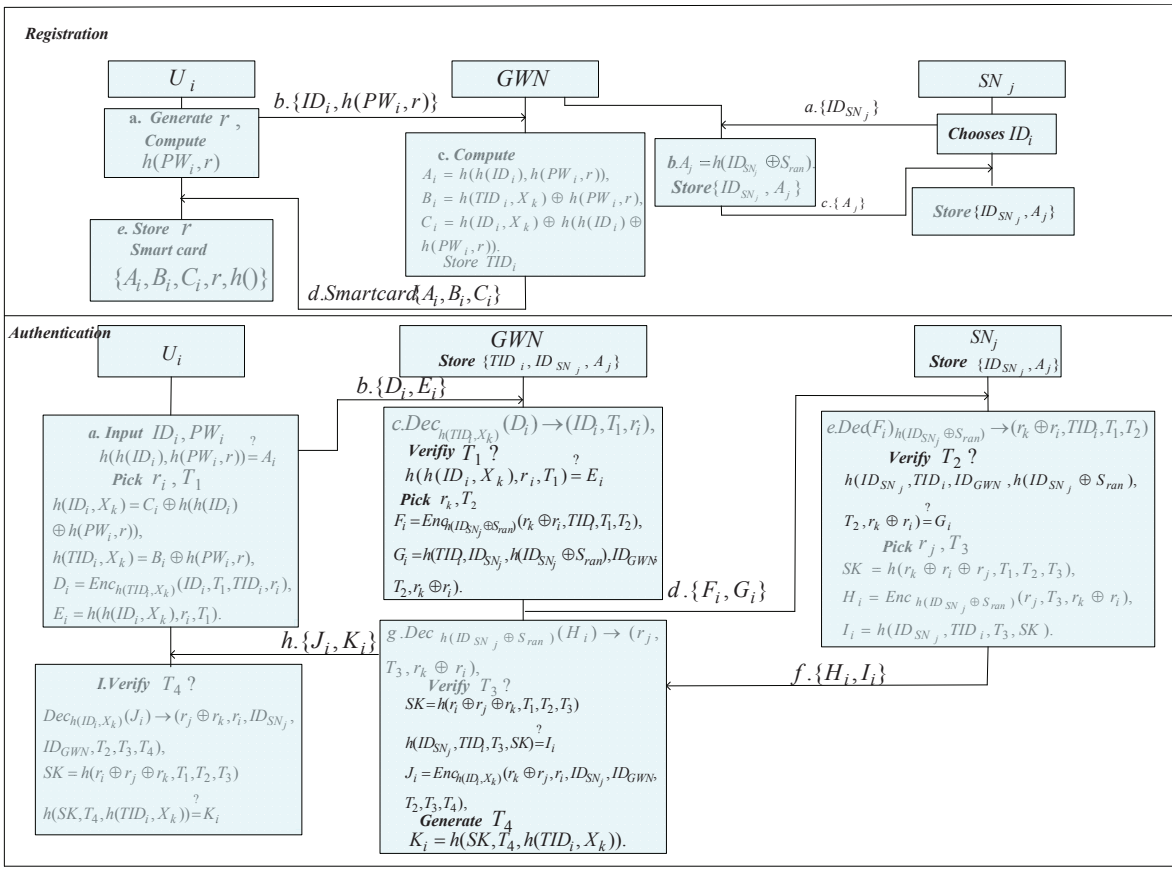

Figure 3. Mutual authentication and key agreement of our scheme.

5. Security Analysis of Our Scheme

In this section, the strength of the proposed AKA scheme by considering the informal and formal analysis has been analyzed. To be specific, our scheme keeps to the system requirements and successfully withstands diverse attacks to enhance the security level. Next, using BAN logic [29] to demonstrate the validity of our AKA scheme. Then, the formal security analysis of our scheme is presented. Besides, the widely-accepted AVISPA tool [28,29] is used to simulated for the security experimental verification of our AKA scheme.

5.1. Informal Security Analysis

This section addresses a detailed security evaluation to indicate that the proposed scheme is secure against various known security attacks. Suppose that an adversary **A** can eavesdrop, intercept, modify, delete or replay the transmission over a public channel.

5.1.1. Session Key Agreement

The session key is established among the user U_i, the sensor node SN_j and the gate-way node GWN. Note that U_i and SN_j has no way to know other participates' random numbers excepts themselves. The established session key is to encrypt the real-time data to ensure the transmission are confidential through an unreliable channel. Therefore, the session key is different in each session due to it is generated by various random numbers, and it is challenging for **A** to extract the current session key from the eavesdropped messages because of the one-way property of the hash function.

5.1.2. Mutual Authentication

The gate-way node GWN first checks whether the received timestamp T_1 is valid as compare to the decrypted one from D_i when receiving the message $\{D_i, E_i, T_1\}$. Next, GWN verifies $h(h(ID_i, X_k), r_i, T_1) \stackrel{?}{=} E_i$. If both the condition are true, the validity of the user U_i is authenticated by GWN. Similarly, U_i checks the validness of the received timestamp with the derived one from J_i after receiving the message $\{J_i, K_i, T_4\}$. He then checks whether $h(SK, T_4, T_1, T_2) \stackrel{?}{=} K_i$. If both the equation hold, the validity of GWN is confirmed by U_i and thus the sensor node SN_j is also verified due to only the valid SN_j would forward the correct random number r_j and thus compute the correct session key. Correspondingly, mutual authentication between SN_j and GWN are performed by checking G_i and I_i. With the same verification mode as GWN and U_i, double authentication is utilized, *i.e.*, to verify the freshness of the received timestamp with the retrieved one, to put the retrieved one to substitute in the awaiting verification value and thus checking the hashed value. In this way, **A** has no ability to modify the hashed value and only modify the timestamp, thus impersonating as any participates. Therefore, mutual authentication among the entities are provided in the proposed scheme.

5.1.3. Resistance to Insider Attack

It is probable that the users use the same identity and password across multiple networks. In our case, the GWN plays the role of a trusted third party, but some curious administrator can have access to the database which stores the user's personal information in order to gain something important. However, during the registration phase, the user U_i transmitted masked password $h(PW_i, r)$ instead of plaintext password. In this way, the insider of system has no ability to derive the privacy of the user because of non-invertible property of one-way hash function. Therefore, the proposed AKA scheme is resilient against the privileged insider attack.

5.1.4. User Anonymity

We adopt two strategies to protect the user's identity from disclosing. One is the masked identity $h(ID_i, X_k)$ with the secret key X_k of GWN. Note that the key is essentially a random number generated by GWN and thus it is computationally infeasible for **A** to extract the user's identity in plaintext. Another is directly the use of dynamic identity selected by GWN, which is hashed in the open channel. In essence, the random identity is no relation with the real one. Consequently, compromise of released one influences nothing on the actual identity of U_i. Therefore, the proposed scheme mechanism is a dynamic identification process and we will verify the point later in simulation.

5.1.5. Resistance to Known Session-Specific Temporary Information Attack

Known session-specific temporary information security means if **A** gets the ephemeral information, such as the random values, $r_a(a = i, k, j)$ and X_k, he still cannot acquire information of the session key. Since **A** has no way to compute the symmetrical key $h(ID_i, X_k)$ without knowing the identity of ID_i and thus decrypting the packs transmitted in communication channel. More seriously, U_i and SN_j only receive the results of xor for the random numbers picked by the rest of participates. As such, attempting to intercept any hashed values in the public communication channel but are unhelpful to compute the session key. Therefore, it is not possible for any attacker to compute the session key on leakage or compromise of session specific temporary information.

5.1.6. Resistance to Denial-of-Service Attack

This attack is to secure against since our proposed scheme works on the principle of request-response communication. Additionally, the sensor node SN_j will check the received packs and chooses refuse or pass the session from the sender. On the other hand, if **A** does the malicious flooding of the authentication requests to SN_j, GWN first knows about malicious dropping of such control messages as a referee. And **A** needs to know the symmetric key between the legal user and

the legitimate sensor node unless he can solve the one-way hash functions. Furthermore, we have introduced timestamps into the scheme, which mitigate any consequential request. As such, we say that our scheme has also the ability to withstand the denial-of-service attack.

5.1.7. Resistance to Sensor Node Impersonation Attack

Suppose **A** gets all transmitted information such as $\{E_i, F_i, G_i\}$ and $\{H_i, I_i\}$ and plans to impersonate as a legitimate sensor node. However, it has no feasible way to decrypt the cryptographic packs like F_i without knowing the symmetry key with the GWN, thus failing to compute the correct session key and thus excluding by GWN. Therefore, **A** can not impersonate as a valid sensor node.

5.1.8. Resistance to Off-Line Password Guessing with Smart Card Breach Attack

The system is secure even if the stored information $\{A_i, B_i, C_i, r, h()\}$ and the login message $\{D_i, E_i\}$ are revealed. Since the user's identity and password are hashed by GWN's long-term private X_k. The adversary **A** has no information about these private keys. Therefore, the proposed scheme is secure against off-line password guessing attack.

5.2. Authentication Proof Based on the BAN Logic

The BAN logic, which is the first suggestion to formalize the description and analysis of authentication schemes, is used to analyze existing schemes to bring out their flaws. We analyze the proposed scheme by establishing some required goals, making some assumptions about the initial state of the scheme and transforming the proposed AKA scheme to the idealized form. Some descriptions about its notations and formulas are shown as follows.

Notations* & *Formulas

·: PX: P has received message X

·: $P| \equiv X$: P believes X

·: $P| \sim X$: P once said X

·: $P \Rightarrow X$: P has jurisdiction over X

·: $P \overset{K}{\rightarrow} Q$: P and Q shared key K

·: $\#(X)$: X is fresh

·: $< X >_K$: the formula X encrypted under the formula K

·: (X, Y): X or Y is one part of (X, Y)

·: $P \overset{K}{\Leftrightarrow} Q$: P and Q share secret K

·: Message meaning rule: $\frac{P| \equiv P \overset{K}{\leftrightarrow} Q,\ P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$

·: Nonce-verification rule: $\frac{P| \equiv \#(X),\ P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$

·: Jurisdiction rule: $\frac{P| \equiv Q \Rightarrow X,\ P| \equiv Q| \equiv X}{P| \equiv X}$

·: Belief rule: $\frac{P| \equiv Q| \equiv (X,Y)}{P| \equiv Q| \equiv X}$

·: Freshness distribution rule: $\frac{P| \equiv \#X}{P| \equiv \#(X,Y)}$

Aims

Aim_1. $GWN| \equiv ID_i$

Aim_2. $SN_j| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$

Aim_3. $SN_j| \equiv GWN| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$

Aim_4. $GWN| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN, GWN| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$

Aim_5. $GWN| \equiv U_i| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$

Aim_6. $GWN| \equiv SN_j| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$

Aim_7. $U_i| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$

Aim_8. $U_i| \equiv GWN| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$

Aim_9. $SN_j| \equiv U_i| \equiv U_i \overset{SK}{\rightleftharpoons} S_j$

Aim_{10}. $U_i| \equiv SN_j| \equiv U_i \overset{SK}{\rightleftharpoons} S_j$

Idealization

$U_i \rightarrow GWN$: $\{D_i, E_i\}$

D_i: $< ID_i, T_1, TID_i, r_i >_{U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} GWN}$, E_i: $< h(ID_i, X_k), r_i, T_1 >$

$GWN \rightarrow SN_j$: $\{TID_i, F_i, G_i, T_2\}$

F_i: $< r_k \oplus r_i, ID_{SN_j}, T_1, T_2 >_{GWN \overset{h(ID_{SN_j} \oplus X_k)}{\rightleftharpoons} SN_j}$, G_i: $< ID_{SN_j}, TID_i, ID_{GWN}, T_2, r_k \oplus r_i >_{GWN \overset{h(ID_{SN_j} \oplus X_k)}{\rightleftharpoons} SN_j}$

$SN_j \rightarrow GWN$: $\{H_i, I_i, T_3\}$

H_i: $< r_j, T_3, r_i \oplus r_k >_{GWN \overset{h(ID_{SN_j} \oplus X_k)}{\rightleftharpoons} SN_j}$, I_i: $< ID_{SN_j}, TID_i, T_3, T_2, SK >_{GWN \overset{SK}{\rightleftharpoons} SN_j}$

$GWN \rightarrow U_i$: $\{J_i, K_i, T_4\}$

J_i: $< r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4 >_{U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} GWN}$

K_i: $< SK, T_4, h(TID_i, X_k) >_{GWN \overset{SK}{\rightleftharpoons} U_i}$

Assumptions

A_1: $U_i| \equiv \#r_i$

A_2: $GWN| \equiv U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} HGWN$

A_3: $U_i| \equiv U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} HGWN$

A_4: $GWN| \equiv GWN \overset{h(ID_{SN_j}, X_k)}{\rightleftharpoons} SN_j$

A_5: $SN_j| \equiv GWN \overset{h(ID_{SN_j}, X_k)}{\rightleftharpoons} SN_j$

A_6: $GWN| \equiv \#TID_i$

A_7: $GWN| \equiv U_i \Rightarrow ID_i$

A_8: $GWN| \equiv X_k$

A_9: $SN_j| \equiv ID_{SN_j}$

A_{10}: $GWN| \equiv U_i \Rightarrow r_i$

A_{11}: $SN_j| \equiv GWN \Rightarrow r_k$

A_{12}: $SN_j| \equiv \#(r_i, r_k, r_j)$

A_{13}: $GWN| \equiv \#(r_i, r_k, r_j)$

A_{14}: $GWN| \equiv SN_j \Rightarrow r_j$

A_{15}: $U_i| \equiv U_i \overset{h(ID_i, X_k)}{\rightleftharpoons} GWN$

Derivation process

According to D_i, we get:

D_1. $GWN \triangleleft < ID_i, T_1, TID_i, r_i >_{U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} GWN}$

According to D_1, A_2 and message rule, we derive:

D_2. $GWN| \equiv U_i \sim (ID_i, T_1, TID_i, r_i)$

According to A_6, D_2 and freshness distribution rule, we gain:

D_3. $GWN| \equiv \#(ID_i, T_1, TID_i, r_i)$

According to D_2-D_3 and nonce-verification rule, we achieve:

D_4. $GWN| \equiv U_i| \equiv (ID_i, T_1, TID_i, r_i)$

According to D_4 and belief rule, we acquire:

D_5. $GWN| \equiv U_i| \equiv ID_i, GWN| \equiv U_i| \equiv r_1, GWN| \equiv U_i| \equiv T_1$

According to D_5, A_7 and jurisdiction rule, we attain:
D_6. $GWN| \equiv ID_i(Aim_1), GWN| \equiv r_i, GWN| \equiv T_1$
According to Aim_1, A_8 and jurisdiction rule, we get:
D_7. $GWN| \equiv h(ID_i, X_k)$
According to F_i, we collect:
D_8. $SN_j \triangleleft < r_k \oplus r_i, ID_{SN_j}, T_1, T_2 >_{GWN \overset{h(ID_{SN_j} \oplus X_k)}{\rightleftharpoons} SN_j}$
According to D_8, A_5 and message rule, we seek:
D_9. $SN_j| \equiv GWN \sim (r_k \oplus r_i, ID_{SN_j}, T_1, T_2)$
According to A_9 and freshness distribution rule, we receive:
D_{10}. $SN_j| \equiv \#(r_k \oplus r_i, ID_{SN_j}, T_1, T_2)$
According to D_9-D_{10} and nonce-verification rule, we extract:
D_{11}. $SN_j| \equiv GWN| \equiv (r_k \oplus r_i, ID_{SN_j}, T_1, T_2)$
According to A_{10}-A_{11}, D_5 and jurisdiction rule, we derive:
D_{12}. $SN_j| \equiv GWN \Rightarrow r_k \oplus r_i$
According to D_{11}-D_{12}, and jurisdiction rule, we regain:
D_{13}. $SN_j| \equiv r_k \oplus r_i$
According to D_{13}, A_{12} and $SK = h(r_k \oplus r_i \oplus r_k)$
Aim_2. $SN_j| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$
According to Aim_2, A_{12} and nonce verification rule, we earn:
Aim_3. $SN_j| \equiv GWN| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$
According to H_i, we get:
D_{14}. $GWN \triangleleft < r_j, T_3, r_i \oplus r_k >_{GWN \overset{h(ID_{SN_j} \oplus X_k)}{\rightleftharpoons} SN_j}$
According to D_{14}, A_4 and message rule, we seek:
D_{15}. $GWN| \equiv SN_j| \sim (r_j, T_3, r_i \oplus r_k)$
According to D_{15}, A_{13}, D_6 and freshness distribution rule, we gain:
D_{16}. $GWN| \equiv SN_j| \equiv \#(r_j, T_3, r_i \oplus r_k)$
According to D_{15}-D_{15} and nonce-verification rule, we derive:
D_{17}. $GWN| \equiv SN_j| \equiv (r_j, T_3, r_i \oplus r_k)$
According to D_{17} and belief rule, we get:
D_{18}. $GWN| \equiv SN_j| \equiv r_j$
According to D_{18}, A_{14} and jurisdiction rule, we regain:
D_{19}. $GWN| \equiv r_j$
According to D_{19}, A_{13}, D_6 and $SK = h(r_j \oplus r_i \oplus r_k)$
Aim_4. $GWN| \equiv U_i \overset{SK}{\rightleftharpoons} GWN, GWN| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$
According to Aim_4, A_{13} and nonce-verification rule, we collect:
Aim_5. $GWN| \equiv U_i| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$
According to I_i, we obtain:
D_{20}. $GWN \triangleleft < ID_{SN_j}, TID_i, T_3, T_2, SK >_{GWN \overset{SK}{\rightleftharpoons} SN_j}$
According to Aim_2, Aim_4, D_{20} and message meaning rule, we get:
D_{21}. $GWN| \equiv SN_j| \sim (ID_{SN_j}, TID_i, T_3, T_2, SK)$
According to D_{21}, Aim_4 and nonce-verification rule, we regain:
Aim_6. $GWN| \equiv SN_j| \equiv SN_j \overset{SK}{\rightleftharpoons} GWN$
According to J_i, we attain:
D_{22}. $U_i \triangleleft < r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4 >_{U_i \overset{h(TID_i, X_k)}{\rightleftharpoons} GWN}$
According to A_{15}, D_{22} and message meaning rule, we reach:
D_{23}. $U_i| \equiv GWN| \sim (r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4)$

According to A_1, D_{23} and freshness distribution rule, we attain:
D_{24}. $U_i| \equiv GWN| \equiv \#(r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4)$
According to D_{23}-D_{24} and nonce-verification rule, we seek:
D_{25}. $U_i| \equiv GWN| \equiv (r_k \oplus r_j, r_i, ID_{SN_j}, ID_{GWN}, T_2, T_3, T_4)$
According to D_{25} and belief rule, we extract:
D_{26}. $U_i| \equiv GWN| \equiv (r_k \oplus r_j)$
According to D_{19}, A_{13}, we get:
D_{27}. $U_i| \equiv GWN \Rightarrow (r_k \oplus r_j)$
According to D_{26}-D_{27}, A_1 and jurisdiction rule, we obtain:
D_{28}. $U_i| \equiv r_k \oplus r_j$
According to D_{28}, A_1 and $SK = h(r_j \oplus r_k \oplus r_j)$, we gain:
Aim_7. $U_i| \equiv U_i \overset{SK}{\rightleftharpoons} GWN$
According to K_i, we seek:
D_{29}. $U_i \triangleleft < SK, T_4, h(TID_i, X_k) >_{GWN \overset{SK}{\rightleftharpoons} U_i}$
According to D_{29}, Aim_4, Aim_7 and message meaning rule, we obtain:
D_{30}. $U_i| \equiv GWN| \sim (SK, T_4, h(TID_i, X_k))$
According to D_{30}, Aim_7 and nonce-verification rule, we reach:
Aim_8: $U_i| \equiv GWN| \equiv SK$
According to Aim_3 and Aim_5, we ge
Aim_9: $SN_j| \equiv U_i| \equiv U_i \overset{SK}{\rightleftharpoons} SN_j$
According to Aim_6 and Aim_8, we get:
Aim_{10}: $U_i| \equiv SN_j| \equiv U_i \overset{SK}{\rightleftharpoons} SN_j$

5.3. Formal Security Proof

In order to show that our scheme is secure, we first define the following assumption:

The encryption algorithm* Ω *assumption: Ω is secure if $Adv_{\mathbf{A}}^{\Omega} \leq \epsilon$ for any sufficiently small $\varepsilon > 0$, any probabilistic, polynomial time adversary $\mathbf{A}$, where $Adv_{\mathbf{A}}^{\Omega}$ denotes the Ω-advantage.

Theorem 1. *Let Ω be secure. Under the assumption that the one-way hash function $h(\cdot)$ closely behaves as an oracle, the proposed scheme is provably secure against an adversary for protecting user anonymity and session key.*

We consider the following two random oracles to construct an adversary $\mathbf{A}$:

Reveal 1: This oracle will unconditionally output the value x from the given hashed result $y = h(x)$.

Reveal 2: This oracle will unconditionally output the plaintext x from the given ciphertext $C = Enc_k(x)$.

Proof of Theorem 1. We assume that $\mathbf{A}$ has the ability to derive the identity ID_i of the user U_i and the session key SK among U_i, the gateway node GWN and the sensor node SN_j. Then he needs to execute the following experimental algorithm, say $EXP1_{\mathbf{A}}^{\Omega}$ (Algorithm 1), $EXP2_{\mathbf{A}}^{Hash}$ (Algorithm 2) for our proposed scheme. Define the success for $EXP1_{\mathbf{A}}^{\Omega}$ as $Succ1_{\mathbf{A}}^{\Omega} = Pr[EXP2_{\mathbf{A}}^{\Omega} = 1] - 1$, $EXP2_{\mathbf{A}}^{Hash}$ as $Succ2_{\mathbf{A}}^{Hash} = Pr[EXP2_{\mathbf{A}}^{Hash} = 1] - 1$, and the advantage for $EXP1_{\mathbf{A}}^{\Omega}$ becomes $Adv1_{\mathbf{A}}^{\Omega}(t_1, q_1) = max_{\mathbf{A}} Succ1_{\mathbf{A}}^{\Omega}$, the advantage for $EXP1_{\mathbf{A}}^{\Omega}$ becomes $Adv2_{\mathbf{A}}^{Hash}(t_2, q_2) = max_{\mathbf{A}} Succ2_{\mathbf{A}}^{Hash}$, where t_i denotes the maximum time interval, q_i denotes the number of queries to the $Reveali(i = 1, 2)$ oracle. However, according to Ω assumption and the one-way property of hash function, both they are hard problems within polynomial time, *i.e.*, $Adv1_{\mathbf{A}}^{\Omega}(t_1, q_1) \leq \epsilon$, $Adv2_{\mathbf{A}}^{Hash}(t_2, q_2) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. As a result, there is no way for the adversary $\mathbf{A}$ to retrieve the user identity ID_i and the session key SK. □

Algorithm 1 $EXP1_{\mathbf{A}}^{\Omega}$.

1: Eavesdrop the login message $\{D_i, E_i\}$, $D_i = Enc_{h(TID_i, X_k)}(ID_i, T_1, TID_i, r_i)$, $E_i = h(h(ID_i, X_k), r_i, T_1)$
2: Call Reveal1 oracle. Let $(ID_i^{'}, T_1^{'}, TID_i^{'}, r_i^{'}) \leftarrow Reveal1(D_i)$
3: Intercept the authenticated message $\{F_i, G_i\}$, where $F_i = E_{h(ID_{SN_j} \oplus S_{ran})}(r_k \oplus r_i, TID_i, T_1, T_2)$, $G_i = h(TID_i, ID_{SN_j}, h(ID_{SN_j} \oplus S_{ran}), ID_{GWN}, T_2, r_k, r_i)$.
4: Call Reveal1 oracle. Let $(r_k^*, r_i^*, TID_i^*, T_1^*, T_2^*) \leftarrow Reveal(F_i)$
5: **If** $(T_1' = T_1^*)$ **then**
6: Accept ID_i' as the true identity of the user U_i
7: **return 1**
8: **else**
9: **return 0**
10: **end if**

Algorithm 2 $EXP2_{\mathbf{A}}^{Hash}$.

1: Eavesdrop the authenticated message $\{G_i, F_i\}$, where $G_i = h(TID_i, ID_{SN_j}, h(ID_{SN_j} \oplus S_{ran}), ID_{GWN}, T_2, r_k, r_i)$, $F_i = E_{h(ID_{SN_j} \oplus S_{ran})}(r_k \oplus r_i, TID_i, T_1, T_2)$
2: Call Reveal2 oracle. Let $(TID_i^{'}, ID_{SN_j}^{'}, h(ID_{SN_j} \oplus$
3: Eavesdrop the communicated message $\{I_i, H_i\}$, $I_i = h(ID_{SN_j}, TID_i, T_3, SK)$, $H_i = Enc_{h(ID_{SN_j} \oplus S_{ran})}(r_j, T_3, r_k, r_i)$
4: Call Reveal2 oracle. Let $(ID_{SN_j}^{''}, TID_i^{''}, T_3^{''}, SK^{''}) \leftarrow Reveal2(D_i)$
5: **If** $(TID_i^{'} = TID_i^{''})$ **then**
6: Accept SK' as the session key among U_i, GWN and SN_j
7: **return 1**
8: **else**
9: **return 0**
10: **end if**

5.4. Simulation Results Using AVISPA Tool

AVISPA is one of the publicly accepted Internet schemes verification techniques among many developed semi-automated formal security analysis tools and several schemes [30,31] have been analyzed using it. It is a push-button tool for error detection based on the Dolev and Yao model [32] and provides a modular role-based expressive formal language called the HLPSL (High level protocol specification language) for targeting the design of the schemes. The HLPSL presentation of the protocol is translated into the lower level description language called IF (Intermediate Format) by the translator called HLPSL2IF, which is the entrance of architecture of AVISPA. IF presentation of the scheme is used as the start point to the four various back-ends: OFMC (On the-fly Model-Checker), CL-AtSe (CL-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree-Automata based Protocol Analyzer). These back-ends are utilized to analyze different security properties such as secrecy of the shared session key, authentication, the privacy of user and robustness against replay attacks. The OF (output format) is generated by using one of the four back-ends which measures whether the security scheme is SAFE or UNSAFE and under what conditions it has been obtained.

In order to evaluate the security of the proposed AKA scheme by the AVISPA tools, we have implemented the specifications for the user U_i (Appendix A, Figure A1), the sensor node SN_j (Appendix A, Figure A2), the gate-way node GWN (Appendix A, Figure A3), the session (Appendix A, Figure A4), goal and the environment (Appendix A, Figure A5) in HLPSL. The desired goals, mutual authentication between U_i and GWN by checking E_i and K_i, between GWN and SN_j by checking G_i

and I_i, the secrecy of session key, user's identity and password are all achieved. We have chosen the widely-accepted OFMC and CL-AtSe back-ends for the execution tests and a bounded number of sessions model checking. In OFMC backend (Figure 4), the depth for the search is 12, the total number of nodes searched in this case is 9143, which takes 44.93 s. In CL-AtSe backend (Figure 5), 7067 states were analyzed and 1360 states were reachable. Further, CL-AtSe backend took 0.46 s for translation and 0.8 s for computation. After simulation of the code through OFMC and CL-AtSe back-ends, the results show the proposed AKA scheme is guard against both the active and passive adversaries.

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /opt/avispa-1.1/testsuite/results/LuS.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 44.93s
 visitedNodes: 9143 nodes
 depth: 12 plies
```

Figure 4. Simulation result for the OFMC.

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /opt/avispa-1.1/testsuite/results/LuS.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed  : 7067 states
 Reachable  : 1360 states
 Translation: 0.46 seconds
 Computation: 0.80 seconds
```

Figure 5. Simulation result for the CL-AtSe.

6. Performance Analysis

This section summarily presents the performance of the proposed AKA scheme and compares in terms of security analysis and computation overheads with existing hash-function based schemes. While computing the cost of the scheme, we assume the length of the identity is 128 bits, the AES encryption/decryption [33] require each 128 bits, the timestamp is 24 bits and the message digest of SHA-3 [34] is 256 bits. Let T_h be the time for one hashing operation, and T_s be the time for one symmetric cryptography operation, we omit xor operation due to its negligible computational cost.

Table 2 shows the computational complexity and communication overhead analysis along the main security attributes with schemes Aim-Biswas [24], Farash *et al.* [23], Turkanović *et al.* [21] and Xue *et al.* [19] It is noted that the communication parameters of the proposed scheme are $\{ID_i, h(PW_i, r), ID_{SN_j}), A_i, B_i, C_i, A_j, D_i, E_i, F_i, G_i, H_i, I_i, J_i, K_i\} = 128 \times 2 + 256 \times 13 = 3680$ bits, the cost of registration is $9T_h$, during the authentication process, the computation cost of the GWN is $5T_h + 3T_s$, the computation cost of the simple resource constrained sensor node is $4T_h + 2T_s$, the total

time spent by the proposed scheme is $22T_h + 7T_s$. According to our experiment results using the jPBC library (2.0.0, [35]) (CPU: 3.2 GHz, RAM: 4.0 GB), the arithmetic mean for executing T_h is 0.0359 ms, T_s is 0.1755 ms after running them 1000 times. Thus, the execution time of the user side is 0.6023 ms, the resource constrained sensor node is 0.4946 ms, the GWN is 0.9214 ms and the total execution time of the proposed AKA scheme is 2.0183 ms. The results shows that the computational cost of the user and the gateway node are considered to be taken on more than sensor node part due to its resource constrained environment. From Table 2, we can see that Farash *et al.*'s scheme [23] achieves more security, that is, resistance to stolen smart card attack and protection of sensor node's identity, although Farash *et al.*'s scheme consumes more computations than Turkanović *et al.* [21]. Even though the efficiency of Aim-Biswas's scheme [24] is higher than Turkanović *et al.* [21]'s scheme, Aim-Biswas's scheme is still vulnerable to known session-specific temporary information attack and no protection of sensor node anonymity. Xue *et al.* [19] is insecure against sensor node impersonation attack and denial-of-service attack excepts vulnerability to known session-specific temporary information attack even though its computational overheads is lower than Farash *et al.*'s scheme. Compared with other four schemes which cannot ensure known session-specific temporary information attack resistance, the proposed AKA scheme consumes a slight higher computation cost lies in using symmetric cryptographic operations. In the face of the perspective of practical application, we consider the security of a cryptographic protocol is the most important. It is acceptable with such high level of security at the expense of increasing computational cost moderately. Therefore, the proposed AKA scheme is very efficient and practical for the resource constrained WSNs environment.

Table 2. Performance analysis.

	Ours	**Aim-Biswas [24]**	**Farash *et al.* [23]**	**Turkanović *et al.* [21]**	**Xue *et al.* [19]**
Communication cost (bits)	3680	3808	3808	2816	3212
Computation cost (user)	$7T_h + 2T_s$	$9T_h$	$13T_h$	$9T_h$	$8T_h$
Computation cost (sensor)	$4T_h + 2T_s$	$5T_h$	$11T_h$	$6T_h$	$8T_h$
Computation cost (GWN)	$11T_h + 3T_s$	$11T_h$	$23T_h$	$12T_h$	$18T_h$
Total (ms)	2.0183	0.8975	1.6873	0.9693	1.2206
*R*1	Yes	*No*	*No*	*No*	*No*
*R*2	Yes	Yes	Yes	Yes	*No*
*R*3	Yes	Yes	Yes	Yes	Yes
*R*4	Yes	Yes	Yes	Yes	*No*
*R*5	Yes	Yes	Yes	Yes	Yes
*R*6	Yes	Yes	Yes	*No*	Yes
*R*7	Yes	*No*	Yes	*No*	Yes

*R*1: Resiliency of known session-specific temporary information attack; *R*2: Resiliency of denial-of-service attack; *R*3: Resiliency of insider attack; *R*4: Resiliency of sensor node impersonation attack; *R*5: User identity protection; *R*6: Resiliency of stolen smart card attack; *R*7: Sensor node anonymity.

7. Conclusions

In this paper, we review and show that Amin-Biswas's scheme is susceptible to known session-specific temporary information attack, thus suffering from various kinds of attacks, such as user impersonation, off-line password guessing attacks and leakage of user identity. In order to erase the drawbacks of Amin-Biswas's scheme, we propose an anonymous AKA scheme for WSNs by using the lightweight operations, such as one-way hash functions, xor and symmetric cryptography. The proposed anonymous AKA scheme is characterized to provide relatively more security features and high security level, simulation results confirmed the efficiency of our proposal in terms of the computation and communication overheads. We are interested in extending the integration of biometrics to design a relatively more efficiency AKA scheme without compromising several security aspects in future.

Acknowledgments: The authors would like to thank all the anonymous reviewers for their helpful advice. This paper is supported by the National Natural Science Foundation of China (Grant Nos. 61472045, 61573067), the Beijing Natural Science Foundation (Grant No. 4142016), the BUPT Excellent Ph.D. Students Foundation (Grant No. CX2015310), and the Asia Foresight Program under NSFC Grant (Grant No. 61411146001).

Author Contributions: Conceived and designed the experiments: Yanrong Lu, Lixiang Li, Haipeng Peng, Yixian Yang. Performed the experiments: Yanrong Lu, Lixiang Li, Haipeng Peng and Yixian Yang. Analyzed the data: Yanrong Lu, Lixiang Li, Haipeng Peng and Yixian Yang. Contributed reagents/materials/analysis tools: Yanrong Lu, Lixiang Li, Haipeng Peng and Yixian Yang. Wrote the paper: Yanrong Lu, Lixiang Li, Haipeng Peng and Yixian Yang.

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

Appendix A HLPSL Implementation of the Proposed Scheme

This section shows the proposed AKA for the roles of the user U_i (Figure A1), the gateway node GWN (Figure A2), the sensor node SN_j (Figure A3), the session (Figure A4) and the environment (Figure A5).

```
role ui     ( UI, GWN, SN     : agent,
              H               : hash_func,
              K               : symmetric_key,
              Snd, Rcv        : channel(dy) )
played_by UI
def=
      local
      State : nat,
            IDi,PWi,R                  :message,
            IDsn,IDgwn,Req,Res,SK      :text,
            Ai,Bi,Ci,Di,Ei,Fi,Gi,Hi,Ii,Ji :text,
            TIDi,Xk,Sran,Aj,Ki         :text,
            Ri,T1,Rk,T2,Rj,T3,T4       :text
      const subs1,subs2,subs3 : protocol_id
      init   State := 0
      transition
     1. State = 0
     /\ Rcv(start)
     =|>
     State'   := 1
     /\ R' := new()
     /\ Snd({IDi.Req}_K)
     /\ Snd({IDi.H(PWi.R')}_K)
     /\ secret(IDi,subs1,{UI,GWN,SN})
     /\ secret(PWi,subs2,{UI,GWN,SN})
     1. State = 1
     /\ Rcv({IDsn.Res}_K)
     /\ Rcv({Ai'.xor(H(TIDi'.Xk'),H(PWi.R')).Ci'}_K)
     =|>
     State'   := 2
     /\ T1' := new()
     /\ Ri' := new()
     /\ Di' := {IDi.T1'.TIDi'.Ri'}_xor(xor(H(TIDi'.Xk'),H(PWi.R')),H(PWi.R'))
     /\ Ei' := H(H(IDi.Xk').Ri'.T1')
     /\ Snd(Di'.Ei'.T1')
     /\ request(UI,GWN,gwn_ui_Ai,Ai')
     /\ witness(UI,GWN,ui_gwn_Ei,Ei')
     2. State = 2
     /\ Rcv({xor(Rk',Rj').Ri'.IDsn.IDgwn}_H(IDi.Xk').H(SK'.T4'.T2'.T1').T4')
     =|>
     State' := 3
     /\ SK' := H(xor(xor(Ri',Rk'),Rj').T1'.T2'.T3')
     /\ request(UI,GWN,gwn_ui_Ki,H(SK'.T4'.H(TIDi'.Xk')))
end role
```

Figure A1. Role specification for the user U_i.

```
role sn      ( UI, GWN, SN     : agent,
               H               : hash_func,
               K               : symmetric_key,
               Snd, Rcv        : channel(dy)  )
played_by SN
def=
      local
      State : nat,
            IDi,PWi,R                  :message,
            IDsn,IDgwn,Req,Res,SK      :text,
            Ai,Bi,Ci,Di,Ei,Fi,Gi,Hi,Ii,Ji :text,
            TIDi,Xk,Sran,Aj,Ki         :text,
            Ri,T1,Rk,T2,Rj,T3,T4       :text
      const subs1,subs2,subs3 : protocol_id
      init   State := 0
      transition
     1. State = 0
     /\ Rcv({IDi.Req}_K)
     =|>
     State'   := 1
     /\ Snd({IDsn}_K)
     2. State = 1
     /\ Rcv({H(xor(IDsn,Sran))}_K)
     =|>
     State'   := 2
     /\ Snd({IDsn.Res}_K)
     3. State = 2
     /\ Rcv(TIDi'.{xor(Ri',Rk').IDsn.T2'}_H(xor(IDsn,Xk')).Gi'.T2'.T1')
     =|>
     State'   := 3
     /\ T3' := new()
     /\ Rj' := new()
     /\ SK' := H(xor(xor(Rk',Ri'),Rj'))
     /\ Hi' := {Rj'.T3'.xor(Rk',Ri')}_H(xor(IDsn,Xk'))
     /\ Ii' := H(IDsn.TIDi'.T3'.T2'.SK')
     /\ Snd(Hi'.Ii'.T3'.T1')
     /\ request(SN,GWN,gwn_sn_Gi,Gi')
     /\ witness(SN,GWN,sn_gwn_Ii,Ii')
     /\ secret(SK',subs3,{UI,GWN,SN})
end role
```

Figure A2. Role specification for the sensor node SN_j.

```
role gwn     ( UI, GWN, SN     : agent,
               H               : hash_func,
               K               : symmetric_key,
               Snd, Rcv        : channel(dy) )
played_by GWN
def=
      local
      State : nat,
            IDi,PWi,R                  :message,
            IDsn,IDgwn,Req,Res,SK      :text,
            Ai,Bi,Ci,Di,Ei,Fi,Gi,Hi,Ii,Ji :text,
            TIDi,Xk,Sran,Aj,Ki         :text,
            Ri,T1,Rk,T2,Rj,T3,T4       :text
      const subs1,subs2,subs3 : protocol_id
      init   State := 0
      transition
     1. State = 0
     /\ Rcv({IDi.H(PWi.R')}_K)
     /\ Rcv({IDsn}_K)
     =|>
     State'   := 1
     /\ TIDi' := new()
     /\ Xk'   := new()
     /\ Sran' := new()
     /\ Ai'   := H(H(IDi).H(PWi.R'))
     /\ Bi'   := xor(H(TIDi'.Xk'),H(PWi.R'))
     /\ Ci'   := xor(H(IDi.Xk'),H(xor(H(IDi),H(PWi.R'))))
     /\ Aj'   := H(xor(IDsn,Sran))
     /\ Snd({Aj'}_K)
     /\ Snd({Ai'.Bi'.Ci'}_K)
     /\ witness(GWN,UI,gwn_ui_Ai,Ai')
     2. State = 1
     /\ Rcv({IDi.T1'.TIDi'.Ri'}_xor(xor(H(TIDi'.Xk'),H(PWi.R')),H(PWi.R')).Ei'.T1')
     =|>
     State'   := 2
     /\ T2' := new()
     /\ Rk' := new()
     /\ Fi' := {xor(Ri',Rk').IDsn.T2'}_H(xor(IDsn,Xk'))
     /\ Gi' := H(TIDi'.IDgwn.T2'.xor(Rk',Ri'))
     /\ Snd(TIDi'.Fi'.Gi'.T2'.T1')
     /\ request(GWN,UI,ui_gwn_Ei,Ei')
     /\ witness(GWN,SN,gwn_sn_Gi,Gi')
     3. State = 2
     /\ Rcv({Rj'.T3'.xor(Rk',Ri')}_H(xor(IDsn,Xk')).H(IDsn.TIDi'.T3'.T2'.SK').T3'.T1')
     =|>
     State' := 3
     /\ T4' := new()
     /\ Ji' := {xor(Rk',Rj').Ri'.IDsn.IDgwn}_H(IDi.Xk')
     /\ Ki' := H(SK'.T4'.T2'.T1')
     /\ Snd(Ji'.Ki'.T4')
     /\ request(GWN,SN,sn_gwn_Ii,H(IDsn.TIDi'.T3'.T2'.SK'))
     /\ witness(GWN,UI,gwn_ui_Ki,Ki')
end role
```

Figure A3. Role specification for the gateway node *GWN*.

```
role session ( UI, GWN, SN     : agent,
               H               : hash_func,
               K               : symmetric_key  )
def=
  local
     S1,S2,S3,R1,R2,R3  : channel(dy)

  composition

        ui(UI, GWN, SN, H, K, S1, R1)

      /\ gwn(UI, GWN, SN, H, K, S2, R2)

      /\ sn(UI, GWN, SN, H, K, S3, R3)

end role
```

Figure A4. Role specification for the session.

```
role environment() def=
 const
      gwn_ui_Ai,ui_gwn_Ei,gwn_sn_Gi,sn_gwn_Ii,gwn_ui_Ki : protocol_id,
      ui, gwn, sn     : agent,
      h               : hash_func,
      k               : symmetric_key
 intruder_knowledge = {ui, gwn, sn, h}
 composition
   session(ui, gwn, sn, h, k)
    /\session(i , gwn, sn, h, k)
    /\session(ui,   i, sn, h, k)
    /\session(ui, gwn,  i, h, k)
end role
goal
 secrecy_of subs1
 secrecy_of subs2
 secrecy_of subs3
authentication_on gwn_ui_Ai
authentication_on ui_gwn_Ei
authentication_on gwn_sn_Gi
authentication_on sn_gwn_Ii
authentication_on gwn_ui_Ki
end goal
environment()
```

Figure A5. Role specification for the environment.

References

1. Jiang, Q.; Ma, J.F.; Lu, X.; Tian, Y.L. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2015**, *8*, 1070–1081.
2. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Comput. Netw.* **2014**, *73*, 41–57.
3. He, D.B.; Zeadally, S.; Xu, B.W.; Huang, X.Y. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2681–2691.
4. Giri, D.; Maitra, T.; Amin, R; Srivastava, P.D. An efficient and robust rsa-based remote user authentication for telecare medical information systems. *J. Med. Syst.* **2015**, *39*, 1–9.
5. Yeh, H.-L.; Chen, T.H.; Liu, P.C.; Kim, T.-H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779.
6. Choi, Y.; Lee, D.; Kim, J.; Jung, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors*. **2014**, *14*, 10081–10106.
7. Watro, R.; Kong, D.; Cuti, S.F.; Gardiner, C.; Lynn, C.; Kruus, P. Tinypk: Securing sensor networks with public key technology. In Proceedings of the 2nd ACM Workshopon Security of Ad Hoc and Sensor Networks (SASN'04), Washington, DC, USA, 25 October 2004; pp. 59–64.

8. Wong, K.H.M.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan, 5–7 June 2006.
9. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
10. Nyang, D.; Lee, M.K. Improvement of Das's two-factor authentication protocol in wireless sensor networks. *IACR Cryptol. ePrint Arch.* **2009**, *2009*, 631.
11. Huang, H.-F.; Chang, Y.F.; Liu, C.H. Enhancement of two-factor user authentication in wireless sensor networks. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Germany, 15–17 October 2010; pp. 27–30.
12. Vaidya, B.; Makrakis, D.; Mouftah, H.T. Improved two-factor user authentication in wireless sensor networks. In Proceedings of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Niagara Falls, ON, Canada, 11–13 October 2010; pp. 600–606.
13. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of two-factor userauthentication in wireless sensor networks. *Sensors* **2010**, *10*, 2450–2459.
14. He, D.J.; Gao, Y.; Chan, S.; Chen, C.; Bu, J.J. An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2010**, *10*, 361–371.
15. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656.
16. Turkanović, M.; Hölbl, M. Notes on "a temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks". *Wirel. Pers. Commun.* **2013**, *77*, 907–922.
17. Wang, D.; Wang, P. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Netw.* **2014**, *20*, 1–15.
18. Li, C.-T.; Weng, C.-Y.; Lee, C.C. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. *Sensors* **2013**, *13*, 9589–9603.
19. Xue, K.P.; Ma, C.S.; Hong, P.L.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323.
20. He, D.B.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inform. Sci.* **2015**, *321*, 263–277.
21. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* **2014**, *20*, 96–112.
22. Chang, C.C.; Le, H.D. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 357–366.
23. Farash, M.S.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176.
24. Amin, R.; Biswas, G.P. A secure lightweight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2015**, *20*, 1–23.
25. Cheng, Z.; Nistazakis, M.; Comley, R.; Vasiu, L. On the in distinguishability-based security model of key agreement protocols-simple cases. *IACR Cryptology ePrint Arch.* **2005**, *2005*, 129.
26. Blake-Wilson, S.; Johnson, D.; Menezes, A. Key agreement protocols and their security analysis. In Proceedings of the Sixth IMA International Conference on Cryptography and Coding, Cirencester, UK, 17–19 December 1997.
27. Bonneau, J. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In Proceedings of the 33th IEEE Symposium on Security and Privacy (S&P 2012), San Francisco, CA, USA, 20–23 May 2012; pp. 538–552.
28. Dell'Amico, M.; Michiardi, P.; Roudier, Y. Password strength: An empirical analysis. In Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM 2010), San Diego, CA , USA, 14–19 March 2010; pp. 1–9.
29. Burrow, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36.
30. AVISPA, Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 6 June 2016).

31. AVISPA, AVISPA Web Tool. Available online: http://www.avispa-project.org/web-interface/expert.php/ (accessed on 6 June 2016).
32. Dolev, D.; Yao, A.C. On the Security of Public Key Protocols. *IEEE Trans. Inform. Theory* **1983**, *29*, 198–208.
33. Advanced Encryption Standard, FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce. Available online: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed on 6 June 2016).
34. SHA-3 Standardization. NIST. Available online: http://csrc.nist.gov/groups/ST/hash/sha-3/sha-3-standardization.html (accessed on 6 June 2016).
35. Java Pairing Based Cryptography Library (JPBC). Available online: http://gas.dia.unisa.it/projects/jpbc (accessed on 6 June 2016).