


ITC 2/47 Journal of Information Technology and Control Vol. 47 / No. 2 / 2018 pp. 275-294 DOI 10.5755/j01.itc.47.2.16397 © Kaunas University of Technology	Security Enhanced and Cost-effective User Authentication Scheme for Wireless Sensor Networks	
	Received 2016/10/10	Accepted after revision 2018/04/03
	 http://dx.doi.org/10.5755/j01.itc.47.2.16397	

Security Enhanced and Cost-effective User Authentication Scheme for Wireless Sensor Networks

Wenfen Liu

School of Computer Science and Information Security, Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
State Key Laboratory of Integrated Services Networks(Xidian University), Xian 710071, China

Gang Zhou, Jianghong Wei

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China
State Key Laboratory of Integrated Services Networks(Xidian University), Xian 710071, China

Xuexian Hu

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China.

Saru Kumari

Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250 005, India.

Corresponding author: jianghong.wei.xxgc@gmail.com

Due to its significant advantages, wireless sensor networks (WSNs) are now widely deployed in various areas to collect and transmit the required data. To ensure only authorized users can login to WSNs, many user authentication schemes based on password and smart card have been proposed. Most recently, Farash et al. and Kumari et al. subsequently proposed an efficient user authentication and key agreement scheme for WSNs, respectively. Even though the two above schemes are claimed to be secure under reasonable assumptions, we find that they, in fact, cannot resist offline password guessing attack when the secret values stored in the smart card are revealed, and also fail to provide forward secrecy. To overcome these security weaknesses, we propose

a novel user authentication scheme for WSNs by introducing Diffie-Hellman key exchange. The security analysis and performance discussion demonstrate that the proposed scheme is secure against various well known attacks, and also is efficient enough. Thus, it is more desirable for securing communications in WSNs.

KEYWORDS: user authentication, cryptanalysis, password, smart card, wireless sensor networks.

Introduction

A wireless sensor network (WSN) usually consists of a large number of autonomous sensor nodes, which only have limited capacity of computation and storage. Specifically, in a WSN, the sensor nodes are in charge of sensing required data and forwarding them to a nearby gateway node (*GWN*), which is regarded as a computation-efficient node, and a valid user is allowed to access these sensor nodes and obtain the collected data. Nowadays, WSNs are widely deployed in many areas, such as healthcare monitoring, environment monitoring, military sensing and tracking, measurement of seismic activity and so on.

Originally, the data collected by sensor nodes are transmitted over a public channel. This implies that an adversary can maliciously delete, intercept the transmitted data, and further destroy the usability and the reliability of the WSN. Particularly, when the data involve sensitive and valuable information, the above security issues become more serious. Therefore, it is necessary to deploy security mechanisms in WSNs for securing communications. Among available security mechanisms designed for WSNs, the user authentication protocol based on password and smart card receives a substantial attention from researchers [34, 31, 18, 17, 19, 16, 15, 10, 33, 9, 5] since it can provide mutual two-factor authentication and establish a shared session key between protocol participants. In addition, this kind of authentication scheme is convenient to be implemented in WSNs, without mandatory requirement for public key infrastructure as in the setting of certificate based authentication scheme.

Compared with user authentication schemes [2, 25, 26, 13] that are solely based on password, the two-factor authentication scheme based on password and smart card, as its name suggests, provides stronger security guarantee. Concretely, in the setting of this kind of authentication scheme, each user holds a password with low entropy and a smart card storing

some secret values. The password and smart card of each user are bonded together by the gateway node. Consequently, a user intending to validly access a sensor node must provide the correct password and the corresponding smart card simultaneously. In order to capture the security of the two-factor authentication scheme based on password and smart card, Xu et al. [36] suggest that the following two assumptions on the adversary's capabilities should be explicitly made:

- The adversary is allowed to record, insert, delete, or modify any message transmitted over the public channel.
- The adversary can either obtain a user's smart card and then extract secret values in it by the method introduced by Kocher et al. [21] and Messerges et al. [27], or get a user's password, but not the both.

For a two-factor authentication scheme based on password and smart card, it is required that the scheme should remain secure under the above two assumptions. This has been widely approved in the literature of two-factor authentication scheme, and the security analysis of lots of such authentication schemes [14, 35, 23, 28, 32, 3, 11, 12] follows from the above assumptions.

In 2009, Das [6] proposed a two-factor user authentication scheme for WSNs by using one-way hash function and exclusive-OR operation, and demonstrated that the proposed scheme can resist many well known attacks. Unfortunately, several subsequent works [29, 4, 37] show that Das's scheme [6] is vulnerable to offline password guessing attack, sensor node compromising attack, gateway node bypassing attack and privileged insider attack. Subsequently, even there are several protocols [20, 1] proposed to conquer the above security pitfalls, they still suffer from various other attacks. For example, Yuan [38] pointed out that Khan and Algathbar's [20] scheme does not provide non-repudiation and fails to achieve mutual authentication between the user and the gateway node. Most

recently, Farash et al. [7] proposed a user authentication scheme for WSN based on password and smart card to overcome the identified security weaknesses in Turkanovic et al.'s [30] scheme, and Kumari et al. [22] introduced another efficient scheme for user authentication and key agreement for WSN.

In this paper, we find that Farash et al.'s [7] scheme suffers from offline password guessing attack, sensor node spoofing attack, and fails to provide anonymity and forward secrecy. We also point out that Kumari et al.'s [22] scheme is vulnerable to offline password guessing attack when the smart card is lost, and thus fails to provide the security guarantee as a two-factor authentication scheme should do. To conquer the security pitfalls in the above two schemes, we propose a novel user authentication scheme based on password and smart card by introducing Diffie-Hellman key exchange. Security analysis and performance discussion show that not only does the proposed scheme achieve intended security properties, but it also has moderate computation cost and communication overhead, and thus is more desirable for securing communications in WSNs.

The rest of the paper is organized as follows. In Section 2, we introduce Farash et al.'s [7] scheme and present the security pitfalls in this scheme. In Section 3, we briefly review Kumari et al.'s [22] scheme and demonstrate that this scheme suffers from offline password guessing attack. The details of the improved scheme is given in Section 4. In Section 5 and Section 6, we discuss the security and performance of the proposed scheme. Finally, we conclude this paper in Section 7.

Table 1

Notations used in this paper

Symbol	Description	Symbol	Description
U_i	i th user	GWN	gateway node of the network
ID_i	identity of i th user	SK	the shared session key between U_i and S_j
PW_i	password of the i th user	$h(\cdot)$	one-way Hash function
TID_i	the provisional identity of i th user	\oplus	bitwise exclusive-OR operation
X_{GWN}	secret key of the gateway node	\parallel	bitwise concatenation operation
S_j	j th sensor node of the network	T_x	current timestamp, $x = 1, 2, \dots$
SID_j	identity of j th sensor node	ΔT	the expected time interval for the transmission delay

2. Security Analysis of Farash et al.'s Scheme

2.1. Review of Farash et al.'s Scheme

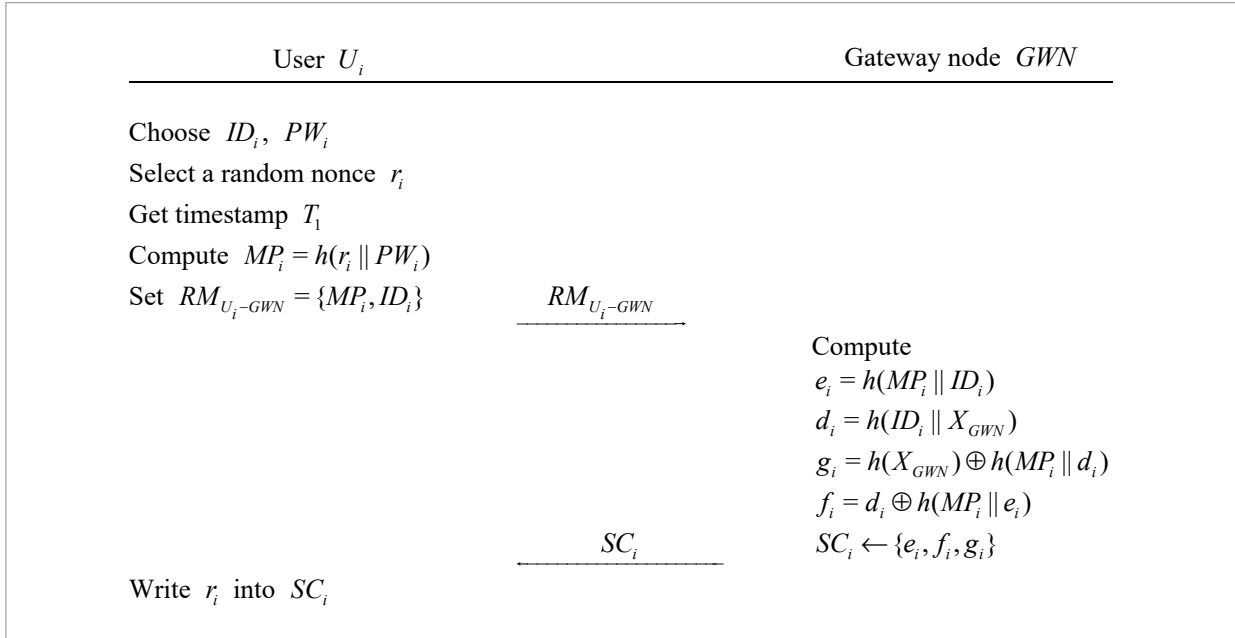
Farash et al.'s [7] authentication scheme involves three participants, i.e., a user U_i , a sensor node S_j and the gateway node GWN . Initially, the gateway node selects a secure one-way hash function $h(\cdot)$, and chooses a random nonce X_{GWN} as its master secret key. In addition, it assigns an identity SID_j and a shared secret value X_{GWN-S_j} for each sensor node S_j . Then, U_i and S_j need to register with the gateway node GWN . During this process, GWN will issue a smart card SC_i containing several secret values to U_i through a *private* channel, and distribute some other secret values to S_j over the *public* channel by using the previously shared secret value X_{GWN-S_j} . After that, whenever the user U_i wants to access the sensor node S_j , they have to authenticate each other by the help of the gateway node GWN , and establish a shared session key for securing subsequent communications.

Specifically, Farash et al.'s scheme consists of three phases, namely, registration phase, authentication phase and password change phase. We now briefly review each phase of this scheme. The notations used throughout this paper are summarized in Table 1.

2.1.1. Registration Phase

The registration phase is comprised of two parts, user registration and sensor node registration. As shown in Figure 1, whenever a user U_i wants to register with

Figure 1

Registration phase for a user U_i in Farash et al.'s scheme

the gateway node GWN , they cooperatively conduct the following steps:

Step 1. U_i chooses an identity ID_i and a password PW_i , as well as a random nonce r_i . Then, U_i computes $MP_i = h(r_i || PW_i)$, and sends the registration message $RM_{U_i-GWN} = \{MP_i, ID_i\}$ to the gateway node GWN in a secure way.

Step 2. Upon receipt of RM_{U_i-GWN} from U_i , the gateway node GWN first checks U_i 's identity, and then successively computes $e_i = h(MP_i || ID_i)$, $d_i = h(ID_i || X_{GWN})$, $g_i = h(X_{GWN}) \oplus h(MP_i || d_i)$ and $f_i = d_i \oplus h(MP_i || e_i)$. At last, GWN issues a smart card SC_i containing $\{e_i, f_i, g_i\}$ to the user U_i .

Step 3. After receiving SC_i from the gateway node GWN , the user U_i writes the previously selected random nonce r_i into SC_i .

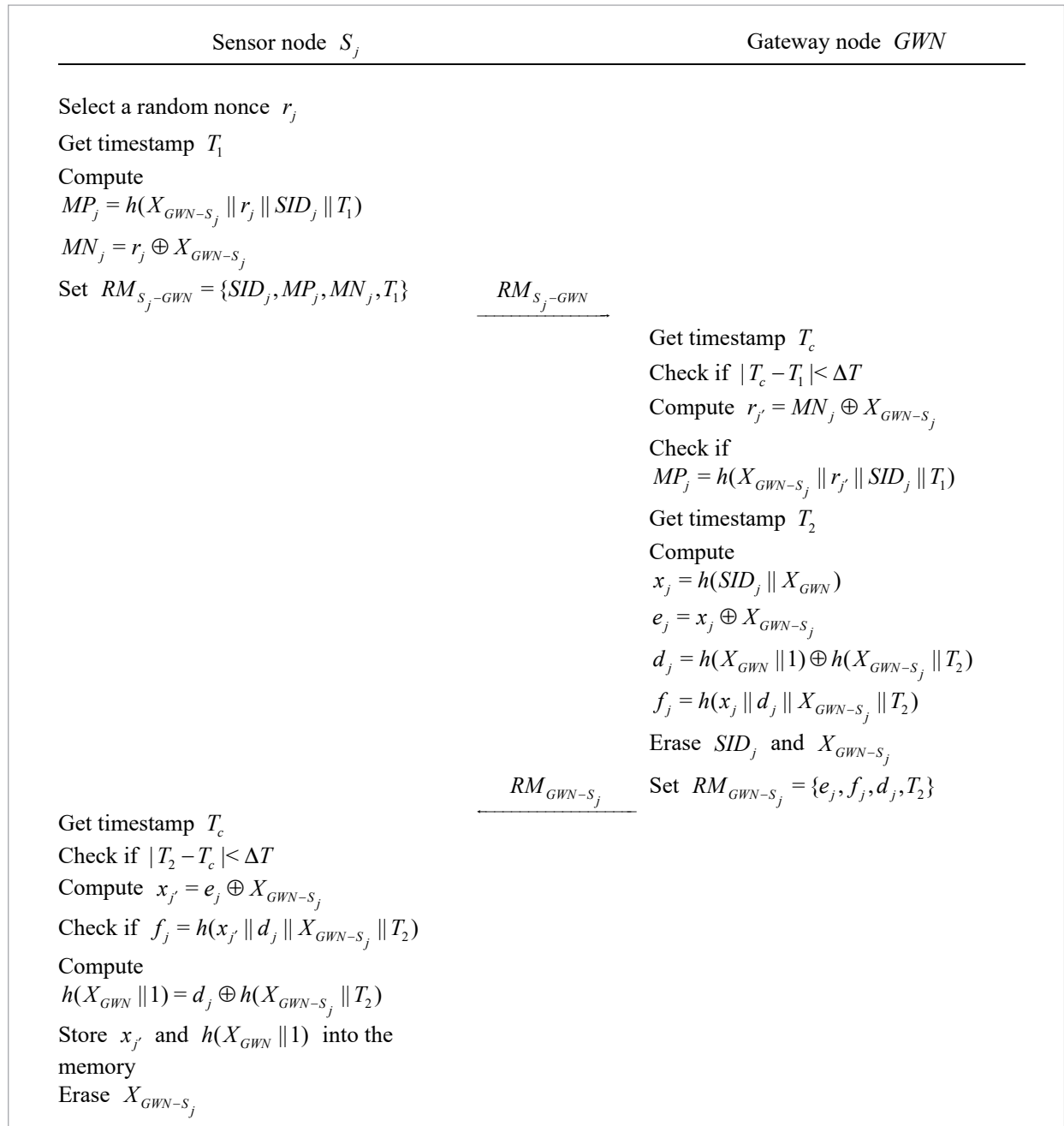
As depicted in Figure 2, for a sensor node S_j holding an identity SID_j and a shared secret value X_{GWN-S_j} it registers with the gateway node GWN by carrying out the following steps:

Step 1. The sensor node S_j first selects a random nonce r_j and gets the current timestamp T_1 . Then, it computes $MP_j = h(X_{GWN-S_j} || r_j || SID_j || T_1)$ and $MN_j = r_j \oplus X_{GWN-S_j}$, and sends the registration mes-

sage $RM_{S_j-GWN} = \{SID_j, MP_j, MN_j, T_1\}$ to the gateway node GWN .

Step 2. After receiving RM_{S_j-GWN} from S_j , the gateway node GWN checks the validity of T_1 by verifying if $|T_c - T_1| < \Delta T$, where T_c is the current timestamp. If T_1 does not pass through the check, GWN rejects S_j 's registration request. Otherwise, it computes $r_{j'} = MN_j \oplus X_{GWN-S_j}$, and further verifies if $MP_j = h(X_{GWN-S_j} || r_{j'} || SID_j || T_1)$. If not, GWN terminates this session. Otherwise, it computes $x_j = h(SID_j || X_{GWN})$, $e_j = x_j \oplus X_{GWN-S_j}$, $d_j = h(X_{GWN} || 1) \oplus h(X_{GWN-S_j} || T_2)$ and $f_j = h(x_j || d_j || X_{GWN-S_j} || T_2)$. Here, T_2 is the current timestamp. Then, the gateway node GWN returns the response message $RM_{GWN-S_j} = \{e_j, f_j, d_j, T_2\}$ to the sensor node S_j . Meanwhile, it deletes SID_j and X_{GWN-S_j} .

Step 3. Upon receipt of RM_{GWN-S_j} from GWN , the sensor node S_j checks the validity of T_2 by verifying if $|T_c - T_2| < \Delta T$, where T_c is the current timestamp. If not, S_j aborts the registration. Otherwise, it computes $x_{j'} = e_j \oplus X_{GWN-S_j}$, and further verifies if $f_j = h(x_{j'} || d_j || X_{GWN-S_j} || T_2)$. If not, S_j also aborts the registration. Otherwise, it stores $x_{j'}$ and $h(X_{GWN} || 1) = d_j \oplus h(X_{GWN} || T_2)$ into its memory.

Figure 2Registration phase for a sensor node S_j in Farash et al.'s scheme

Meanwhile, S_j erases the previously shared secret value X_{GWN-S_j} .

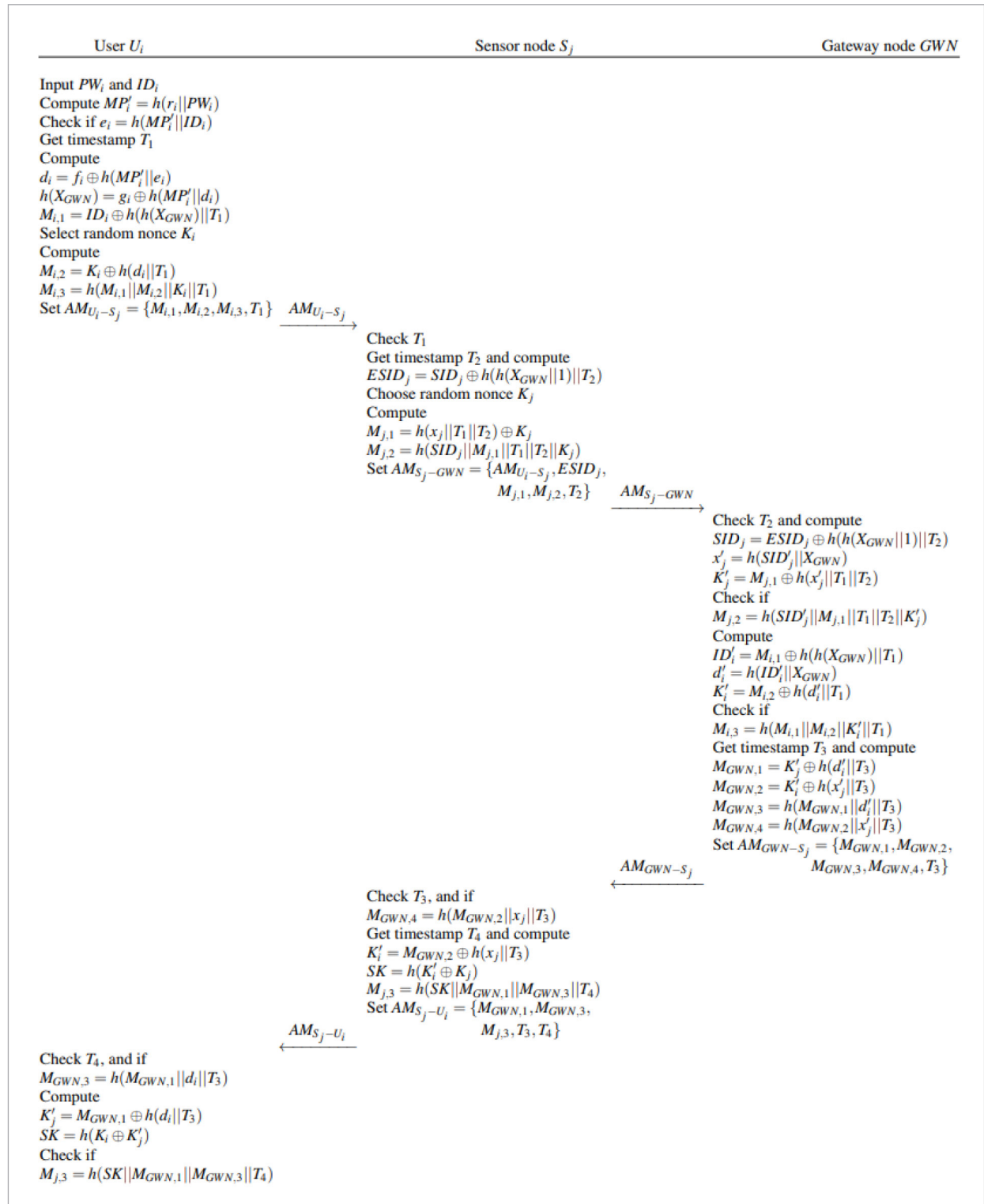
2.1.2. Authentication Phase

Whenever a user U_i wants to access a sensor node S_j ,

he/she has to complete mutual authentication and establish a shared session key for securing subsequent communications with the help of the gateway node GWN . Concretely, as depicted in Figure 3, the authentication procedure is performed as follows:

Figure 3

Authentication phase in Farash et al.'s scheme



Step 1. The user U_i inserts the smart card SC_i into a card reader, and inputs the identity ID_i and the password PW_i . The smart card SC_i computes $MP_i = h(r_i \| PW_i)$, and checks if $e_i = h(MP_i \| ID_i)$. If not, SC_i terminates the authentication procedure. Otherwise, it gets the current timestamp T_1 and computes $d_i = f_i \oplus h(MP_i \| e_i)$, $h(X_{GWN}) = g_i \oplus h(MP_i \| d_i)$ and $M_{i,1} = ID_i \oplus h(h(X_{GWN}) \| T_1)$. Moreover, SC_i chooses a random nonce K_i , and produces $M_{i,2} = K_i \oplus h(d_i \| T_1)$ and $M_{i,3} = h(M_{i,1} \| M_{i,2} \| K_i \| T_1)$. After that, SC_i sends the authentication request message $AM_{U_i-S_j} = \{M_{i,1}, M_{i,2}, M_{i,3}, T_1\}$ to the sensor node S_j .

Step 2. After receiving the message $AM_{U_i-S_j}$ from U_i , the sensor node S_j checks the validity of T_1 by verifying if $|T_c - T_1| < \Delta T$, where T_c is the current timestamp. If not, the sensor node S_j aborts the authentication process. Otherwise, it gets the current timestamp T_2 and computes a temporary identity $ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$. Furthermore, S_j selects a random nonce K_j , and generates $M_{j,1} = h(x_j \| T_1 \| T_2) \oplus K_j$ and $M_{j,2} = h(SID_j \| M_{j,1} \| T_1 \| T_2 \| K_j)$. Then, it sends the authentication request message $AM_{S_j-GWN} = \{AM_{U_i-S_j}, ESID_j, M_{j,1}, M_{j,2}, T_2\}$ to the gateway node GWN .

Step 3. Upon receipt of AM_{S_j-GWN} from S_j , the gateway node GWN checks the validity of T_2 by verifying if $|T_c - T_2| < \Delta T$, where T_c is the current timestamp. If not, GWN aborts the authentication procedure. Otherwise, it computes $SID_j = ESID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$, $x_{j'} = h(SID_{j'} \| X_{GWN})$, $K_{j'} = M_{j,1} \oplus h(x_{j'} \| T_1 \| T_2)$, and further verifies if $M_{j,2} = h(SID_{j'} \| M_{j,1} \| T_1 \| T_2 \| K_{j'})$. If not, GWN also terminates the session. Otherwise, it further computes $ID_{i'} = M_{i,1} \oplus h(h(X_{GWN}) \| T_1)$, $d_{i'} = h(ID_{i'} \| X_{GWN})$ and $K_{i'} = M_{i,2} \oplus h(d_{i'} \| T_1)$. Then, GWN checks if $M_{i,3} = h(M_{i,1} \| M_{i,2} \| K_{i'} \| T_1)$. If not, GWN rejects the authentication request. Otherwise, it gets the current timestamp T_3 and computes $M_{GWN,1} = K_{j'} \oplus h(d_{i'} \| T_3)$, $M_{GWN,2} = K_{i'} \oplus h(x_{j'} \| T_3)$ and $M_{GWN,3} = h(M_{GWN,1} \| d_{i'} \| T_3)$, as well as $M_{GWN,4} = h(M_{GWN,2} \| x_{j'} \| T_3)$. After that, the gateway node GWN sends the authentication message to the sensor node S_j .

Step 4. Once receiving AM_{GWN-S_j} from the gateway node GWN , the sensor node S_j checks if $|T_c - T_3| < \Delta T$ and $M_{GWN,4} = h(M_{GWN,2} \| x_{j'} \| T_3)$, where T_c is the current timestamp. If not, S_j

aborts this procedure. Otherwise, S_j obtains the current timestamp T_4 and further computes $K_{i'} = M_{GWN,2} \oplus h(x_{j'} \| T_3)$, $SK = h(K_{i'} \oplus K_{j'})$ and $M_{j,3} = h(SK \| M_{GWN,1} \| M_{GWN,3} \| T_4)$. After that, the sensor node S_j sends the message $AM_{S_j-U_i} = \{M_{GWN,1}, M_{GWN,3}, M_{j,3}, T_3, T_4\}$ to the user U_i .

Step 5. When receiving the message $AM_{S_j-U_i}$ from S_j , the smart card SC_i verifies the validity of T_4 by checking if $|T_c - T_4| < \Delta T$. If not, SC_i aborts the session. Otherwise, it computes $K_{j'} = M_{GWN,1} \oplus h(d_{i'} \| T_3)$ and $SK = h(K_{i'} \oplus K_{j'})$, and further verifies if $M_{j,3} = h(SK \| M_{GWN,1} \| M_{GWN,3} \| T_4)$. If not, SC_i also terminates the session. At this point, U_i and S_j complete mutual authentication and share a common session key $SK = h(K_{i'} \oplus K_{j'})$.

2.1.3. Password Change Phase

In this phase, a user U_i is allowed to update his/her password offline. To this end, the user U_i and the smart card SC_i interactively perform as follows:

Step 1. The user U_i inserts the smart card SC_i into a card reader and inputs the identity ID_i and the password PW_i .

Step 2. The smart card SC_i computes $MP_i = h(r_i \| PW_i)$ and checks if $e_i = h(MP_i \| ID_i)$. If not, the smart card rejects the user's password update request. Otherwise, it further computes $d_i = f_i \oplus h(MP_i \| e_i)$ and $h(X_{GWN}) = h(MP_i \| d_i) \oplus g_i$. After that, the smart card SC_i requires U_i to input a new password.

Step 3. The user U_i selects and inputs a new password PW_i^{new} .

Step 4. The smart card computes $MP_i^{new} = h(r_i \| PW_i^{new})$, $e_i^{new} = h(MP_i^{new} \| ID_i)$, $f_i^{new} = d_i \oplus h(MP_i^{new} \| e_i^{new})$ and $g_i^{new} = h(X_{GWN}) \oplus h(MP_i^{new} \| d_i^{new})$. Then, SC_i successively replaces e_i , f_i and g_i with e_i^{new} , f_i^{new} and g_i^{new} .

2.2. Security Pitfalls of Farash et al.'s Protocol

In this section, we demonstrate that Farash et al.'s [7] scheme suffers from offline dictionary attack with smart card lost and sensor node spoofing attack with sensor node capture. In addition, we also show that this protocol fails to achieve anonymity and forward

secrecy. Here, we emphasize that we discuss the security of Farash et al.'s protocol under the same threat assumption as adopted in [7].

2.2.1. Offline Dictionary Attack

In this attack, an adversary \mathcal{A} first observes an authentication instance executed among a user U_i , a sensor node S_j and the gateway node GWN , and records these messages $AM_{U_i-S_j}$, AM_{S_j-GWN} , AM_{GWN-S_j} and $AM_{S_j-U_i}$, which are transmitted over a public channel. Then, \mathcal{A} obtains the user U_i 's smart card SC_i and extracts the values r_i, e_i, f_i and g_i stored in SC_i by using technologies introduced in [21, 27]. After that, the adversary \mathcal{A} launches offline dictionary attack by conducting the following steps:

Step 1. Establish a password dictionary space \mathcal{D}_i .

Step 2. Select a candidate password PW_i^* from the dictionary space \mathcal{D}_i , and compute $MP_i^* = h(r_i \| PW_i^*)$, $K_i^* = M_{i,2} \oplus h(d_i^* \| T_1)$ as well as $d_i^* = f_i \oplus h(MP_i^* \| e_i)$.

Step 3. Check the validity of PW_i^* using one of the following manners:

- Compute $M_{i,3}^* = h(M_{i,1} \| M_{i,2} \| K_i^* \| T_1)$ and verify if $M_{i,3}^* = M_{i,3}$.
- Compute $M_{MGN,3}^* = h(M_{GWN,1} \| d_i^* \| T_3)$, and verify if $M_{MGN,3}^* = M_{MGN,3}$.
- Compute $K_j^* = M_{GWN,1} \oplus h(d_i^* \| T_3)$, $SK^* = h(K_i^* \oplus K_j^*)$ and $M_{j,3}^* = h(SK^* \| M_{GWN,1} \| M_{GWN,3} \| T_4)$, and verify if $M_{j,3}^* = M_{j,3}$.
- Compute $h(X_{GWN})^* = g_i \oplus h(MP_i^* \| d_i^*)$, $ID_i^* = M_{i,1} \oplus h(h(X_{GWN})^* \| T_1)$ and $e_i^* = h(MP_i^* \| ID_i^*)$, and verify if $e_i^* = e_i$.

Step 4. If PW_i^* passes through the above check then it must be that $PW_i^* = PW_i$. This completes the attack. Otherwise, choose a new candidate password from \mathcal{D}_i , and repeat the Steps 2 and 3 until the correct password is found.

Denote by T_h the running time of a hash operation and T_{xor} the running time of an XOR operation. If we choose one of the first two equalities (i.e., $M_{i,3}^* = M_{i,3}$ and $M_{MGN,3}^* = M_{MGN,3}$) to check the validity of a candidate password, then the time complexity of the above attack procedure is $\mathcal{O}(4T_h + 2T_{xor})$, which is nearly negligible. On the other hand, since passwords are usually generated in a personal way such that they can be easily memorable by human beings, the size of the

dictionary space \mathcal{D}_i will be very limited. Thus, once a user's smart card is lost, an adversary can recover the correct password within seconds by running the above attack procedure on a PC. After that, as shown in the fourth check manner, with the recovered correct password PW_i , the adversary can further get the user's identity ID_i . As a result, the adversary \mathcal{A} can legitimately access any sensor node on behalf of the user U_i just by obeying the authentication mechanism.

2.2.2. Sensor Node Spoofing Attack

In this attack, an adversary \mathcal{A} first corrupts a sensor node S_c , and obtains the identity SID_c and the secret values $x_c, h(X_{GWN} \| 1)$. Then, the adversary \mathcal{A} impersonates any sensor node S_j a user U_i is trying to access. The details of this attack are as follows:

Step 1. When the user U_i sends the message $AM_{U_i-S_j} = \{M_{i,1}, M_{i,2}, M_{i,3}, T_1\}$ to the sensor node S_j , the adversary \mathcal{A} intercepts this message. Then, it computes $ESID_c = SID_c \oplus h(h(X_{GWN} \| 1) \| T_2)$, $M_{c,1} = h(x_c \| T_1 \| T_2) \oplus K_c$ and $M_{c,2} = h(SID_c \| M_{c,1} \| T_1 \| T_2 \| K_c)$. After that, the adversary \mathcal{A} sends the message $AM_{S_c-GWN} = \{AM_{U_i-S_j}, ESID_c, M_{c,1}, M_{c,2}, T_2\}$ to the gateway node GWN .

Step 2. When receiving the message AM_{S_c-GWN} from the sensor node S_c , the gateway node GWN performs the same as in **Step 3** of the authentication phase. Since the message $AM_{U_i-S_j}$ does not contain any information about the intended sensor node identity SID_j , the gateway node GWN does not know that this message is originally sent to S_j , rather than S_c . On the other hand, the adversary \mathcal{A} has the correct values x_c and $h(X_{GWN} \| 1)$, and thus can pass through the verification of the gateway node GWN . Hence, the gateway node would conclude that the message AM_{S_c-GWN} is correct and return the response message $AM_{GWN-S_c} = \{M_{GWN,1}, M_{GWN,2}, M_{GWN,3}, M_{GWN,4}, T_3\}$ to the sensor node S_c .

Step 3. After receiving the message AM_{GWN-S_c} from the gateway node GWN , with the knowledge of x_c and $h(X_{GWN} \| 1)$, the adversary \mathcal{A} can correctly compute $K_i' = M_{GWN,2} \oplus h(x_c \| T_3)$, $SK = h(K_i' \oplus K_c)$ and $M_{j,3} = h(SK \| M_{GWN,1} \| M_{GWN,3} \| T_4)$. After that, \mathcal{A} sends the message $AM_{S_j-U_i} = \{M_{GWN,1}, M_{GWN,3}, M_{j,3}, T_3, T_4\}$ to the user U_i .

Step 4. Upon receipt of the message $AM_{S_j-U_i}$, the user U_i checks if $M_{GWN,3} = h(M_{GWN,1} || d_i || T_3)$ and $M_{j,3} = h(SK || M_{GWN,1} || M_{GWN,3} || T_4)$. Evidently, the two values would pass through the check since the gateway node GWN correctly produces $M_{GWN,3}$ using d_i and the adversary \mathcal{A} also computes the correct value $M_{j,3}$ with the knowledge of SK .

At last, the user U_i and the adversary \mathcal{A} complete mutual authentication and establish a shared session key $SK = h(K_i \oplus K_c)$, which implies that \mathcal{A} has succeeded in masquerading as the sensor node S_j .

2.2.3. Fail to Achieve Anonymity and Forward Secrecy

In Farash et al.'s scheme, to provide user and sensor node anonymity, a user U_i and a sensor node S_j use different temporary identities $M_{i,1} = ID_i \oplus h(h(X_{GWN}) || T_1)$ and $ESID_j = h(h(x_{GWN} || 1) || T_2)$ in each authentication process. It seems that only the gateway node GWN , with the knowledge of the secret key X_{GWN} , can recover the original identities ID_i and SID_j . However, we note that all of users share a common secret value $h(X_{GWN})$ and all of sensor nodes share another common secret value $h(X_{GWN} || 1)$. Consequently, a malicious user U_m , who possesses $h(X_{GWN}) = g_m \oplus h(MP_m || d_m)$, can extract any user U_i 's identity from $M_{i,1}$ by computing $ID_i = M_{i,1} \oplus h(X_{GWN})$, and a corrupted sensor node S_c , who holds $h(X_{GWN} || 1)$, can obtain any sensor node S_j 's identity by computing $SID_j = ESID_j \oplus h(h(X_{GWN} || 1) || T_1 || T_2)$, where T_1 and T_2 are the corresponding timestamps. Therefore, even if the private values of a user U_i and a sensor node S_j are absolutely secure, Farash et al.'s scheme cannot guarantee the anonymity of U_i and S_j .

The session key in Farash et al.'s protocol is computed as $SK = h(K_i \oplus K_j)$, where $K_i = M_{i,2} \oplus h(d_i || T_1)$ and $K_j = M_{j,1} \oplus h(x_j || T_1 || T_2)$ are two random values independently chosen by two protocol participants, a user U_i and a sensor node S_j . Thus, once either U_i 's smart card and password are compromised or S_j 's secret value x_j is revealed, an adversary can recover K_i and K_j from those messages transmitted over public channel, and further obtain the session key $SK = h(K_i \oplus K_j)$. Therefore, Farash et al.'s protocol fails to provide forward secrecy.

3. Security Analysis of Kumari et al.'s Scheme

Kumari et al. [22] proposed a new authentication protocol for WSN to partially conquer the above security pitfalls in Farash et al.'s [7] protocol. Roughly speaking, the two authentication protocols have the similar structure. For the limit of space, we just briefly review the user registration phase and login phase of Kumari et al.'s protocol, and then show that this protocol suffers from offline password guessing attack when the smart card is lost.

3.1. A Brief Review of Kumari et al.' Scheme

In the user registration phase of Kumari et al.'s [22] scheme, a user U_i registers with GWN by carrying out the following steps:

Step 1. U_i chooses an identity ID_i and a password PW_i , as well as a random nonce r_i . Then, U_i computes $MID_i = h(r_i || ID_i)$ and $MP_i = h(r_i || PW_i)$, and sends the registration message $RM_{U_i-GWN} = \{MP_i, MID_i\}$ to the gateway node GWN in a secure way.

Step 2. Upon receipt of RM_{U_i-GWN} from U_i , the gateway node GWN first checks U_i 's identity, and then successively computes $e_i = h(MP_i || MID_i)$, $d_i = h(MID_i || X_{GWN})$, $g_i = h(X_{GWN} || y_i) \oplus h(MP_i || d_i)$ and $f_i = d_i \oplus h(MP_i || e_i)$, where y_i is a random number. At last, GWN issues a smart card SC_i containing $\{e_i, f_i, g_i, y_i, h(\cdot)\}$ to the user U_i .

Step 3. After receiving SC_i from the gateway node GWN the user U_i computes $c_i = r_i \oplus h(ID_i || PW_i)$, and writes c_i into SC_i .

In the login phase, the user U_i performs the following operations:

Step 1. U_i inserts the smart card SC_i into a device reader and inputs his/her identity ID_i and password PW_i . Then, the smart card SC_i computes $r_i' = c_i \oplus h(ID_i || PW_i)$, $MID_i' = h(r_i' || ID_i)$ as well as $MP_i' = h(r_i' || PW_i)$. Moreover, the smart card checks whether $e_i = h(MP_i' || MID_i')$ or not. If not, the smart card terminates the login process.

Step 2. In the case that ID_i and PW_i are both correct, the smart card SC_i further computes $d_i = f_i \oplus h(MP_i' || e_i)$, $h(X_{GWN} || y_i) = g_i \oplus h(MP_i' || d_i)$,

$M_1 = ID_i \oplus h(X_{GWN} \| y_i \| T_1)$. The smart card then picks a random number K_i and continues to calculate $M_2 = K_i \oplus h(d_i \| T_1)$, $M_3 = h(M_1, M_2, SID_j, K_i, T_1)$, where T_1 is the current time stamp on the user side and SID_j is the identity of the sensor node S_j to be accessed. Finally, the smart card sends the login request message $AM_{U_i-S_j} = \{M_1, M_2, M_3, y_i, T_1\}$ to the sensor node S_j via a public channel.

3.2. Security Pitfalls in Kumari et al.'s Protocol

In this section, we demonstrate that Kumari et al.'s protocol is vulnerable to offline password guessing attack when the smart card is lost. In Kumari et al.'s protocol, since a user U_i needs to provide his/her identity ID_i and password PW_i simultaneously in the login phase, they have the same feature. Namely, they are both easy to remember and thus suffer from the threat of offline password guessing attack.

After obtaining a login request message $AM_{U_i-S_j} = \{M_1, M_2, M_3, y_i, T_1\}$ and the corresponding user U_i 's smart card SC_i , an adversary \mathcal{A} first extracts $\{e_i, f_i, g_i, c_i, y_i, h(\cdot)\}$ from SC_i . Then, the adversary \mathcal{A} launches offline dictionary attack by conducting the following steps:

Step 1. Establish a password dictionary space \mathcal{D}_{pw} and an identity dictionary space \mathcal{D}_{id} , respectively.

Step 2. Select a candidate password PW_i^* from the dictionary space \mathcal{D}_{pw} and a candidate identity ID_i^* from the dictionary space \mathcal{D}_{id} , and sequentially compute $r_i^* = c_i \oplus h(ID_i^* \| PW_i^*)$, $MP_i^* = h(r_i^* \| PW_i^*)$ and $MID_i^* = h(r_i^* \| ID_i^*)$. $MP_i^* = h(r_i \| PW_i^*)$, $K_i^* = M_{i,2} \oplus h(d_i^* \| T_1)$ as well as $d_i^* = f_i \oplus h(MP_i^* \| e_i)$.

Step 3. Check the validity of PW_i^* and ID_i^* using one of the following manners:

- Check if $e_i = h(MP_i^* \| MID_i^*)$.
- Compute $M_3^* = h(M_1 \| M_2 \| SID_j \| K_i^* \| T_1)$, and verify if $M_3 = M_3^*$.

Step 4. If PW_i^* and ID_i^* pass through the above check, then it must be that $PW_i^* = PW_i$ and $ID_i^* = ID_i$. This completes the attack. Otherwise, choose a new candidate password and identity from \mathcal{D}_{pw} and \mathcal{D}_{id} , respectively, and repeat Steps 2 and 3 until the correct password is found.

Denote by T_h the running time of a hash operation and T_{xor} the running time of an XOR operation. If

we choose the first equality to check the validity of a candidate password and a candidate identity, then the time complexity of the above attack procedure is $\mathcal{O}(4T_h + T_{xor})$, which is nearly negligible.

We note that the above attack implies that an adversary can directly recover a user's identity and password simultaneously. Thus, the protocol naturally fails to achieve user anonymity. In addition, similar to Farash et al.'s protocol, Kumari et al.'s protocol also cannot provide forward secrecy since the authentication procedure only involves XOR operation, and an adversary can thus utilize the secret key of the gateway node to recover all secret values from those transmitted messages.

4. The Proposed Protocol

In this section, we propose an improved authentication protocol AP that conquers the security pitfalls in Farash et al.'s [7] protocol and Kumari et al.'s [22] protocol. Next, we provide the details of the protocol.

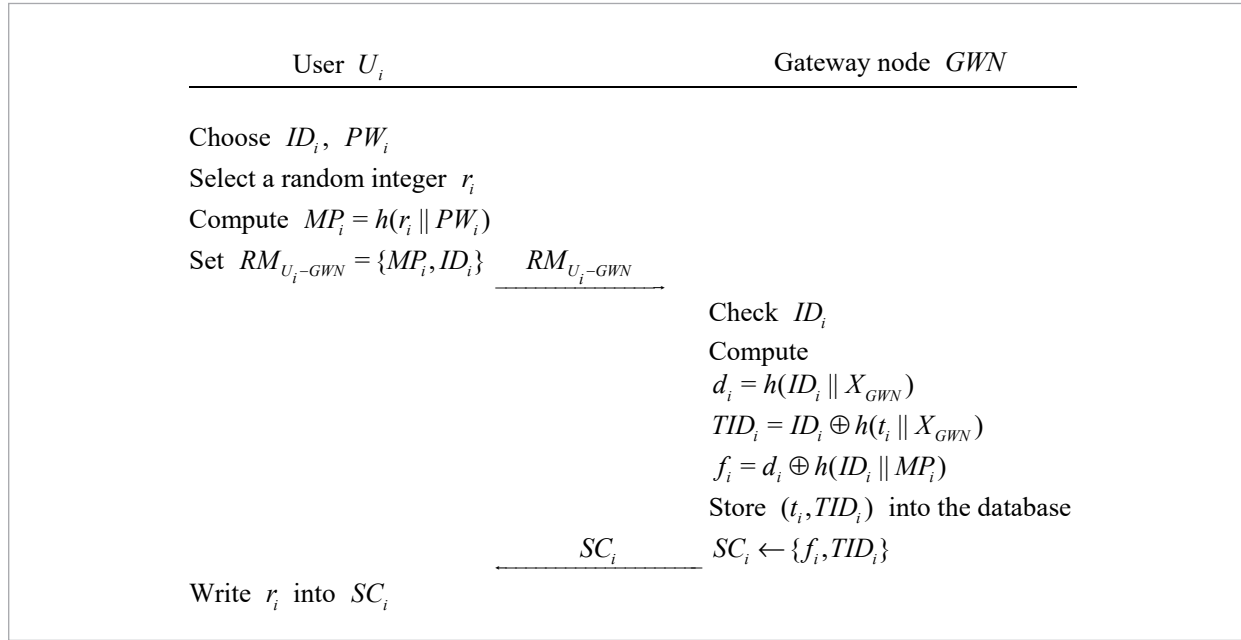
Similarly, there are three kinds of participants in the protocol AP , namely, a user U_i , a sensor node S_j and the gateway node GWN . Initially, the gateway node GWN chooses an elliptic curve group G with prime order p . Let g be a random generator of G . GWN also chooses a secure one-way hash function $h(\cdot): \{0,1\}^* \rightarrow \{0,1\}^\ell$. Then, the gateway node GWN selects a random integer $X_{GWN} \in \mathbb{Z}_p^*$ as its long-term secret key. For each sensor node S_j , the gateway node GWN assigns a unique identity SID_j to identify S_j , and stores a secret value $x_j = h(SID_j \| X_{GWN})$ into S_j 's memory before deploying it into the network. This, in fact, completes S_j 's registration to the gateway node. We now describe the details of the protocol AP .

4.1. Registration Phase

In this phase, a user U_i wanting to access any sensor node registers with the gateway node GWN . As shown in Figure 4, the user U_i and the gateway node GWN interactively complete the registration process by carrying out the following steps:

Step 1. U_i selects an identity ID_i and a password PW_i , as well as a random nonce $r_i \in \mathbb{Z}_p^*$. Then, U_i computes $MP_i = h(r_i \| PW_i)$, and sends the registration message $RM_{U_i-GWN} = \{MP_i, ID_i\}$ to the gateway node

Figure 4

Registration phase in the protocol *AP*

GWN via a secure channel.

Step 2. After receiving RM_{U_i-GWN} from U_i , the gateway node GWN first checks the uniqueness of ID_i , namely, whether ID_i is occupied by the other registered users. If yes, the gateway node GWN prompts U_i to choose a new identity. Otherwise, GWN selects a random nonce $t_i \in \mathbb{Z}_p^*$, and then successively computes $d_i = h(ID_i || X_{GWN})$, $TID_i = ID_i \oplus h(t_i || X_{GWN})$ and $f_i = d_i \oplus h(ID_i || MP_i)$. At last, GWN issues a smart card SC_i containing $\{f_i, TID_i\}$ to the user U_i , and stores the tuple (t_i, TID_i) into the user database.

Step 3. Upon receipt of SC_i from GWN , the user U_i writes the previously selected random nonce r_i into SC_i .

4.2 Authentication Phase

In this phase, a user U_i intending to access a sensor node S_j authenticates against S_j to ensure that S_j is a valid sensor node deployed by the gateway node GWN . Meanwhile, the sensor node S_j verifies U_i 's validity to avoid unauthorized access. When they successfully complete mutual authentication, a shared session key is established for securing subsequent communications between U_i and S_j . Concretely, as depicted in Figure 5, the authentication procedure is executed in the following manner:

Step 1. U_i inserts the smart card SC_i into a terminal and inputs the identity ID_i and the password PW_i . The smart card SC_i computes $MP_i = h(r_i || PW_i)$ and $d_i = f_i \oplus h(ID_i || TID_i || MP_i)$. Moreover, SC_i selects a random integer $x \in \mathbb{Z}_p^*$ and sets $K_i = g^x$. Then, it gets the current timestamp T_1 , and further computes $d_i^* = d_i \oplus K_i$ and $M_{i,1} = h(d_i^* || TID_i || SID_j || T_1)$. After that, SC_i sends the authentication request message $AM_{U_i-S_j} = \{d_i^*, TID_i, M_{i,1}, T_1\}$ to the sensor node S_j .

Step 2. Upon receipt of $AM_{U_i-S_j}$ from U_i , the sensor node S_j captures the current timestamp T_c and checks whether $|T_c - T_1| < \Delta T$ and $M_{i,1} = h(d_i^* || TID_i || SID_j || T_1)$, where ΔT is the allowed maximum transmission delay. If not, S_j terminates this session. Otherwise, it chooses a random integer $y \in \mathbb{Z}_p^*$ and sets $K_j = g^y$. Then, it gets the current timestamp T_2 , and computes the two values $M_{j,1} = h(x_j || T_2) \oplus K_j$ and $M_{j,2} = h(M_{j,1} || AM_{U_i-S_j} || K_j)$. Subsequently, it sends the authentication request message $AM_{S_j-GWN} = \{AM_{U_i-S_j}, SID_j, M_{j,1}, M_{j,2}, T_2\}$ to the gateway node GWN .

Step 3. After receiving AM_{S_j-GWN} from S_j , the gateway node GWN checks the validity of T_2 and $M_{j,2}$

in a similar way. If they are not acceptable, GWN terminates this session. Otherwise, GWN computes $x_{j'} = h(SID_j \| X_{GWN})$ and $c_{j'} = M_{j,1} \oplus h(x_{j'} \| T_2)$. Then, the gateway node GWN examines whether it holds that $M_{j,2} = h(M_{j,1} \| AM_{U_i-S_j} \| SID_j \| K_{j'} \| T_2)$. If not, GWN also terminates this session. Otherwise, it retrieves the tuple (t_i, TID_i) from the database and recovers $ID_{i'} = TID_i \oplus h(t_i \| X_{GWN})$. Moreover, GWN computes $d_{i'} = h(ID_{i'} \| X_{GWN})$ and $K_{i'} = d_{i'} \oplus d_i^*$. After that, the gateway node GWN selects a random integer $z \in \mathbb{Z}_p^*$ and captures the current timestamp T_3 , and further computes the following values:

$$M_{GWN,1} = h(K_{j'} \| T_3),$$

$$M_{GWN,2} = h(x_{j'} \| TID_i \| T_3) \oplus (K_{j'})^z,$$

$$M_{GWN,3} = h(d_{i'} \| SID_j \| T_3) \oplus (K_{j'})^z,$$

$$M_{GWN,4} = h(M_{GWN,1} \| M_{GWN,2} \| M_{GWN,3} \| T_3).$$

At the end, the gateway node GWN sends the response message $AM_{GWN-S_j} = \{M_{GWN,1}, M_{GWN,2}, M_{GWN,3}, M_{GWN,4}, T_3\}$ to the sensor node S_j .

Step 4. Once receiving AM_{GWN-S_j} from GWN , the sensor node S_j gets the current timestamp T_c and verifies whether $|T_c - T_3| < \Delta T$ and $M_{GWN,4} = h(M_{GWN,1} \| M_{GWN,2} \| M_{GWN,3} \| T_3)$. If not, S_j aborts this session. Otherwise, S_j further checks whether it holds that $M_{GWN,1} = h(K_{j'} \| T_3)$. If not, S_j also terminates this session. Otherwise, S_j authenticates against the gateway node GWN . Moreover, S_j recovers $(K_{j'})^z = M_{GWN,2} \oplus h(x_{j'} \| TID_i \| T_3)$. Then, it obtains the current timestamp T_4 and computes $M_{j,3} = h((K_{j'})^{z'} \| T_3 \| T_4)$ and $M_{j,4} = h(M_{j,3} \| M_{GWN,3} \| T_3 \| T_4)$. After that, the sensor node S_j sends the message $AM_{S_j-U_i} = \{M_{GWN,3}, M_{j,3}, M_{j,4}, T_3, T_4\}$ to the user U_i .

Step 5. When receiving $M_{S_j-U_i}$ from S_j , the smart card SC_i first checks the validity of T_4 and $M_{j,4}$. If they are not acceptable, SC_i aborts this session. Otherwise, it computes $(K_{j'})^z = M_{GWN,3} \oplus h(d_{i'} \| SID_j \| T_3)$ and verifies if $M_{j,3} = h((K_{j'})^{z'} \| T_3 \| T_4)$. If not, SC_i also terminates this session. Otherwise, SC_i authenticates against the sensor node S_j and produc-

es the session key as $SK = h((K_{j'})^{z'} \| TID_i \| SID_j)$. Moreover, it gets the current timestamp T_5 and computes $M_{i,2} = h((K_{j'})^{z'} \| T_5)$, and sends the message $AM_{U_i-S_j} = \{M_{i,2}, T_5\}$ to the sensor node S_j .

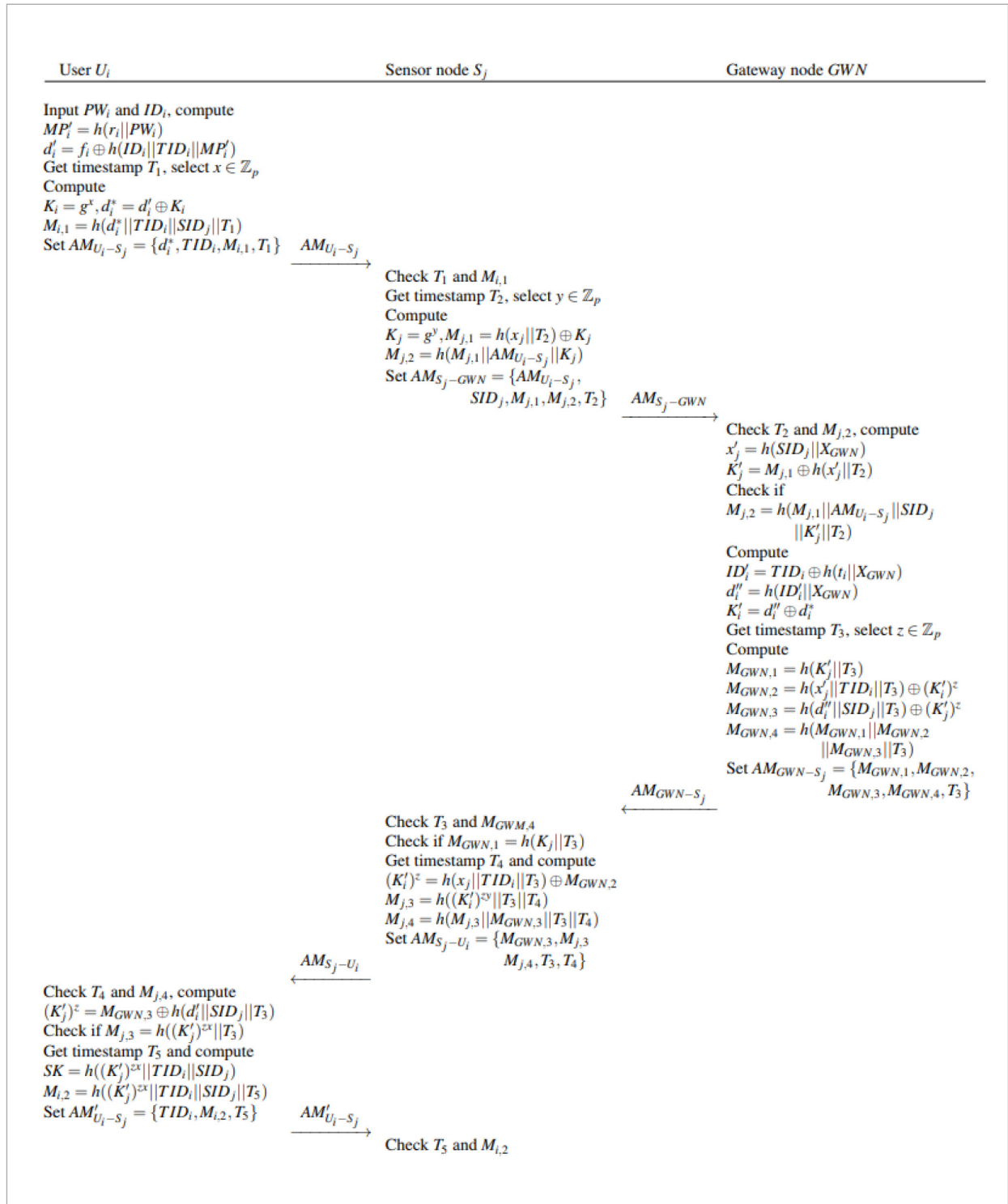
Step 6. After receipt of $AM_{U_i-S_j}$ from U_i , the sensor node S_j gets the current timestamp T_c and checks whether $|T_c - T_5| < \Delta T$ and $M_{i,2} = h((K_{j'})^{z'} \| T_5)$. If not, S_j terminates this session. Otherwise, the sensor node S_j authenticates against the user U_i and generates the session key as $SK' = h((K_{j'})^{z'} \| TID_i \| SID_j)$.

Here we briefly describe the intuition behind the above authentication mechanism. First, U_i sends a hidden challenge value K_i in the form of $d_i^* = K_i \oplus d_{i'}$ to S_j , where $d_{i'} = h(ID_{i'} \| X_{GWN})$, such that only the gateway node GWN can recover it from d_i^* with the knowledge of long-term secret key X_{GWN} . We emphasize that the computation of hash value $M_{i,1}$ does not involve any private value (e.g., K_i). Thus, it naturally cannot be used to check the validity of a candidate password. After receiving the authentication message, S_j itself produces challenge values $M_{j,1} = h(x_j \| T_2) \oplus K_j$ and $M_{j,2} = h(M_{j,1} \| AM_{U_i-S_j} \| K_j \| T_2)$, and sends them to GWN . With these two values and the long-term secret value X_{GWN} , the gateway node can ensure that S_j is a registered sensor node since it holds the initially issued secret value x_j . Moreover GWN recovers U_i 's identity ID_i and the challenge value K_i , but it cannot directly check their validity. Therefore, GWN also chooses a challenge value z , and computes $M_{GWN,1}$ and $M_{GWN,2}$ for S_j , and $M_{GWN,3}$ for U_i . By checking the two values, S_j can be convinced that GWN also knows the secret value x_j and thus is valid. Meanwhile, S_j recomputes the hash value $M_{j,3}$ for U_i . Given $M_{j,3}$ and $M_{GWN,3}$, the user U_i can make sure that S_j knows the value K_i^z and thus is valid. Finally, U_i generates a hash value $M_{i,2}$ for S_j to prove that he also knows the value K_j^z , which implies that U_i is a valid user.

4.3. Password Change Phase

In this phase, a user U_i updates the original password PW_i under the supervision of the gateway node GWN . To this end, the user U_i should be authenticated by a sensor node and the gateway node in advance, which guarantees that the original password

Figure 5

Authentication phase in the protocol *AP*

PW_i and the identity ID_i input by the user U_i are correct. After that, U_i is required to select and input a new password PW_i^{new} . Then, the smart card SC_i successively computes $MP_i^{new} = h(r_i || PW_i^{new})$ and $f_i^{new} = f_i \oplus h(ID_i || MP_i) \oplus h(ID_i || MP_i^{new})$, and replaces f_i with f_i^{new} . This completes U_i 's password update. In the protocol AP , password change is done in an online way, rather than offline update as in Farash et al.'s protocol. Essentially, the difference between the two methods comes from the fact that who is in charge of checking the validity of the password PW_i and the identity ID_i input by the user U_i . Note that it is the gateway node GWN in the protocol AP , and the smart card SC_i in Farash et al.'s protocol. However, as demonstrated in Subsection 2.2, once the corresponding verification information stored in the smart card SC_i is revealed, an adversary can utilize it to launch offline password guessing attack. This is also why we adopt an online manner of updating password. Namely, a smart card in the protocol AP does not contain any information that can be directly used to check the validity of the corresponding password.

5. Security Analysis

In this section, we evaluate the security of the protocol AP . Specifically, we demonstrate that AP can withstand various well-known attacks, including offline password guessing attack, user/sensor node impersonation attack, parallel and reflection attack, reply attack and privileged insider attack. We also show that AP features desired security properties, such as mutual authentication, user anonymity and key agreement.

5.1. Offline Password Guessing Attack

Assuming an adversary \mathcal{A} has obtained a legal user U_i 's smart card SC_i , from which \mathcal{A} extracted $\{f_i, r_i, TID_i\}$, where $f_i = h(ID_i || X_{GWN}) \oplus h(ID_i || h(r_i || PW_i))$. Moreover, we suppose that \mathcal{A} also has recorded these authentication messages $AM_{U_i-S_j}$, AM_{S_j-GWN} , AM_{GWN-S_j} , $AM_{S_j-U_i}$ and $AM_{U_i-S_j}'$ that were transmitted publicly among U_i , S_j and GWN . Now we show that \mathcal{A} cannot use the above values to verify the validity of a candidate password. Given a candidate password PW_i^* ,

the adversary \mathcal{A} would compute $MP_i^* = h(r_i || PW_i^*)$, $d_i^* = f_i \oplus h(ID_i || TID_i || MP_i^*)$ and $K_i^* = d_i^* \oplus d_i^*$. If one of the above values is correctly computed, then PW_i^* is the correct one (i.e., $PW_i^* = PW_i$). Since MP_i^* is just used to compute d_i^* , thus the only way for \mathcal{A} to launch offline password guessing attack is to check the correctness of d_i^* or K_i^* .

First, if the long-term secret key X_{GWN} gets compromised, then \mathcal{A} can compute $d_i = h(ID_i || X_{GWN})$ and further check the validity of PW_i^* by comparing d_i^* with d_i . Of course, the offline password guessing attack in this case is *trivial*. Second, note that the computation of $M_{GWN,3}$ involves d_i , which means that \mathcal{A} can recover $(K_j)^z = M_{GWN,3} \oplus h(d_i^* || SID_j || T_3)$. Observe that $(K_j)^z$ is never transmitted among the protocol participants and thus is not available to \mathcal{A} . Consequently, \mathcal{A} cannot utilize the recovered value $(K_j)^z$ to check the validity of d_i^* . Moreover, even if S_j 's secret key x_j is revealed, which implies that \mathcal{A} can recover $K_j = M_{j,1} \oplus h(x_j || T_2)$, the adversary \mathcal{A} also cannot check the validity of the recovered value $(K_j)^z$ without the knowledge of z , which is randomly sampled from \mathbb{Z}_p^* by the gateway node. To get the value z , the adversary \mathcal{A} has to solve the discrete logarithm problem, which is believed to be hard. Thus, \mathcal{A} cannot verify the validity of d_i^* . Finally, note that \mathcal{A} can utilize x_j to recover $(K_i)^z = M_{GWN,2} \oplus h(x_j || TID_i || T_3)$. However, without the knowledge of z , the adversary \mathcal{A} also cannot check the validity of K_i^* . Therefore, we conclude that the proposed protocol is secure against offline password guessing attack, even if the private information stored in the smart card gets compromised and the sensor node a user trying to access is corrupted.

5.2. User Impersonation Attack

In this attack, an adversary \mathcal{A} intends to access a sensor node S_j by impersonating an honest user U_i . To this end, from the protocol flow we know that \mathcal{A} is initially required to produce an authentication message $AM_{U_i-S_j} = \{d_i^*, TID_i, M_{i,1}, T_1\}$ and finally has to generate a response message $AM_{U_i-S_j}' = \{TID_i, M_{i,2}, T_5\}$, where $d_i^* = d_i \oplus K_i$ and $K_i = g^x$. By the protocol criteria, if \mathcal{A} can pass S_j 's check, then it must hold that $M_{i,2} = h((K_j)^{zx} || TID_i || SID_j || T_5) = M_{i,2}' = h((K_i)^{zy} ||$

$TID_i \parallel SID_j \parallel T_3$). Moreover, due to the property of the hash function $h(\cdot)$ withstanding collision attack, the above equality indicates that $(K_{j'})^{zx} = (K_r)^{zy}$. If $K_{r'} = h(ID_{r'} \parallel X_{GWN}) \oplus d_i^* = K_i$, then the previous equality requires that \mathcal{A} must correctly recover $(K_{j'})^z = M_{GWN,3} \oplus h(d_{r'} \parallel SID_j \parallel T_3)$, which also means that \mathcal{A} has to get the value $d_{r'} = h(ID_i \parallel X_{GWN}) = f_i \oplus h(ID_i \parallel TID_i \parallel h(r_i \parallel PW_i))$. There are two ways for \mathcal{A} to compute this value, namely, getting the long-term secret key X_{GWN} or U_i 's password PW_i and the value f_i stored in U_i 's smart card SC_i . In the first case that \mathcal{A} obtains X_{GWN} , it can impersonate any legal user. In the second case that \mathcal{A} gets PW_i and f_i , it in fact has corrupted the user U_i . Thus, despite in which case \mathcal{A} 's impersonation attack is trivial. If $K_{r'} \triangleq g^{x'} \neq K_i$, then \mathcal{A} possessing K_i and $(K_{j'})^z = g^{yz}$ must correctly compute $(K_{r'})^{zy} = g^{x'yz}$. From the discrete logarithm assumption we know that this is impossible for \mathcal{A} without the knowledge of x' . In conclusion, the proposed protocol can withstand user impersonation attack.

5.3. Sensor Node Spoofing Attack

In this attack, a malicious sensor node S_c tries to impersonate an honest sensor node S_j that a user U_i intends to access. Recall that the reason why Farash et al's protocol suffers from sensor node spoofing attack is that the message $AM_{U_i-S_j}$ in their protocol does not contain any information about S_j 's identity SID_j , and the gateway node does not care which sensor node that U_i is trying to access. To fix this security pitfall, we let U_i compute $M_{i,1} = h(d_i^* \parallel TID_i \parallel redSID_j \parallel T_1)$ and GWN produce $M_{GWN,3} = h(d_{r'} \parallel redSID_j \parallel T_3) \oplus (K_{j'})^z$, which guarantees that the sensor node U_i is trying to access is consistent with the one that authenticates against the gateway node GWN . In other words, from S_c 's perspective, if it wants to pass through GWN 's authentication on behalf of S_j , then it must know the value x_j , which implies that S_j is corrupted and this attack is trivial. On the other hand, S_c can successfully authenticate to GWN by using its own secure value x_c . However, this will result in that GWN would compute $M_{GWN,3} = h(d_{r'} \parallel redSID_c \parallel T_3) \oplus (K_{j'})^z$ and U_i would recover $(K_{j'})^z = M_{GWN,3} \oplus h(d_{r'} \parallel redSID_j \parallel T_3)$. Clearly, we have that $(K_{j'})^z \neq (K_{r'})^z$ under the assumption that $h(\cdot)$ can withstand collision attack.

Consequently, S_c cannot pass through U_i 's authentication because $M_{j,3} \neq h((K_{j'})^{zx} \parallel T_3)$. Hence, the proposed protocol can resist sensor node spoofing attack.

5.4. Reflection Attack

In a reflection attack, when an honest protocol participant sends to an intended communication partner for the later to perform a cryptographic process, an adversary \mathcal{A} intercepts the message and simply sends it back to the message originator. In such an attack, \mathcal{A} tries to deceive the message originator into believing that the reflected message is expected by the originator from the intended communication partner, either as a response to, or as a challenge for, the originator. If \mathcal{A} is successful, the message originator would either accept an "answer" to a question which was, in fact, asked and answered by the originator itself, or would provide \mathcal{A} with an oracle service which \mathcal{A} needs but cannot provide to itself.

In the proposed protocol, a user U_i sends the message $AM_{U_i-S_j}$ to a sensor node S_j , from which U_i expects to receive $AM_{S_j-U_i}$. Obviously, an adversary \mathcal{A} cannot pass through U_i 's authentication by simply sending $AM_{U_i-S_j}$ back to U_i , since $AM_{S_j-U_i}$ and $AM_{U_i-S_j}$ are different in terms of structure and associated timestamp. Moreover, S_j successively sends AM_{S_j-GWN} and $AM_{S_j-U_i}$ to the gateway node GWN , and expects to receive AM_{GWN-S_j} from GWN and $AM_{U_i-S_j}$ from U_i , respectively. For the same reason, the adversary \mathcal{A} also cannot utilize these messages to launch reflection attack. Therefore, the proposed protocol is secure against reflection attack.

5.5. Replay Attack

In a message replay attack, an adversary \mathcal{A} has recorded a old message from a preceding instance of a protocol and now replays the recorded message in a new instance of this protocol. To eliminate this attack against the proposed protocol, we use timestamp and random nonce to guarantee the freshness of exchanged messages among communication partners. Specifically, note that each message in the proposed protocol is associated with the corresponding timestamp, which implies that if \mathcal{A} wants to replay these messages, then it has to modify the previous timestamps. For the recorded old mes-

sage $AM_{U_i-S_j} = \{d_i^*, TID_i, M_{i,1}, T_1\}$, an adversary \mathcal{A} can get the current timestamp T_1^{new} and compute $M_{i,1}^{new} = h(d_i^* || TID_i || SID_j || T_1^{new})$, and further replay $AM_{U_i-S_j}^{new} = \{d_i^*, TID_i, M_{i,1}^{new}, T_1^{new}\}$ to S_j . However, \mathcal{A} cannot pass through S_j 's authentication because does not know the previous random nonce $K_i = g^x$ and thus cannot produce $M_{i,2}^{new} = h((K_i')^{zx} || TID_i || SID_j || T_5^{new})$ correctly. For the recorded old messages AM_{S_j-GWN} , AM_{GWN-S_j} , $AM_{S_j-U_i}$ and $AM_{U_i-S_j}'$, if the adversary \mathcal{A} replaces those old timestamps T_2, T_3, T_4, T_5 with the current ones $T_2^{new}, T_3^{new}, T_4^{new}, T_5^{new}$, then it has to recompute $M_{j,1}^{new} = h(x_j || T_2^{new})$, $M_{GWN,1}^{new} = h(K_j || T_3^{new})$, $M_{j,3}^{new} = h((K_i')^{zy} || T_3^{new} || T_4^{new})$ and $M_{i,2}^{new} = h((K_j')^{xz} || TID_i || SID_j || T_5^{new})$. Obviously, this is impossible for \mathcal{A} since it does not know the secret value x_j and random nonces K_i and K_j . In conclusion, the proposed protocol can withstand message replay attack.

5.6. Privileged Insider Attack

In a privileged attack, a malicious insider \mathcal{M} can get any data stored in the memory of the gateway node GWN except the long-term secret key X_{GWN} . Below we argue that \mathcal{M} cannot obtain any information about a registered user U_i 's password PW_i and identity ID_i . First, note that when U_i registers with GWN , he/she sends $MP_i = h(r_i || PW_i)$, rather than PW_i or $h(PW_i)$, to the gateway node GWN , where r_i is a random nonce. Moreover, GWN itself does not store any information that can be used to verify the validity of PW_i . As a consequence, \mathcal{M} cannot launch offline password guessing attack without the knowledge of r_i . Second, to provide user anonymity, the gateway node GWN stores a tuple (t_i, TID_i) for each registered user U_i , where t_i is a random nonce and $TID_i = ID_i \oplus h(t_i || X_{GWN})$. Even if this tuple gets compromised, \mathcal{M} cannot utilize it to recover U_i 's real identity ID_i , without the knowledge of the long-term secret key X_{GWN} . Consequently, the proposed protocol can be free from privileged attack.

5.7. Perfect Forward Secrecy

The idea of perfect forward secrecy is that when a long-term secret key is revealed, session keys that were previously established using that long-term secret key should not be compromised. In the proposed protocol, the session key, in fact, is computed as

$SK = h(g^{xyz} || TID_i || SID_j)$, where x, y, z are random nonces selected by U_i , SID_j and GWN , respectively. Particularly, these random nonces are erased at the end of each authentication procedure. When the long-term secret key (e.g., U_i 's smart card and password, SID_j 's secret key x_j and GWN 's secret key X_{GWN}) gets compromised, an adversary \mathcal{A} can recover $K_i = g^x$, $K_j = g^y$, $K_i^z = g^{xz}$ and $K_j^z = g^{yz}$ from those publicly transmitted messages. To recompute the previously established session key SK , the adversary \mathcal{A} has to recompute g^{xyz} with the above values. By the computational Diffie-Hellman assumption, we know that this is impossible for \mathcal{A} . Hence, the proposed protocol enjoys perfect forward secrecy.

5.8. Mutual Authentication and Key Agreement

Mutual authentication guarantees that both protocol participants are authenticated to each other in the same protocol instance. That is, each one has a fresh assurance of the identity of the peer one. The proposed protocol achieves mutual authentication between a user U_i and a sensor node S_j , which implicitly includes mutual authentication between S_j and the gateway node GWN . Specifically, throughout the authentication procedure, U_i and S_j independently generate their fresh challenge values $K_i = g^x$ and $K_j = g^y$, which are both transmitted to the gateway node GWN in a hidden way, i.e., $d_i^* = d_i \oplus K_i$ and $M_{j,1} = h(x_j || T_2) \oplus K_j$. With the knowledge of X_{GWN} , GWN can correctly recover K_i and K_j , and further verify the validity of S_j by checking $M_{j,2}$. At this moment, S_j is authenticated by GWN . Moreover, GWN computes response value $M_{GWN,1}$ for S_j , and returns the modified challenge values K_j^z and K_i^z to U_i and S_j in a private way. If $M_{GWN,1}$ passes through S_j 's check, then GWN is authenticated by S_j . This completes mutual authentication between S_j and GWN . Then, S_j itself computes the response value $M_{j,3}$ to certify that it indeed has the knowledge of K_i^z and y , which also implies that S_j is an authorized sensor node with the identity SID_j . If $M_{j,3}$ is checked to be valid, then S_j is authenticated by U_i . Finally, U_i generates a response value $M_{i,2}$ to prove that it has the knowledge of K_j^z and x . If $M_{i,2}$ is verified to be correct, then U_i is authenticated by S_j . Now U_i and S_j complete mutual authentication, under the help of the gateway node GWN .

When U_i and S_j accomplish mutual authentication, a shared session key $SK = h(g^{xyz} \parallel TID_i \parallel SID_j)$ is immediately established between them for subsequent cryptographic use. Note that SK is separately generated by each participant using its own contributed information and received information. For example, U_i computes $g^{xyz} = (K_j^z)^x$, where x is random nonce chosen by U_i and K_j^z is recovered from the received message. Therefore, U_i and S_j have the same influence on the value of the shared session key, namely, neither principal can control the shared secret value. This realizes the security goal of key agreement.

5.9. Weak User Anonymity

In the context of the proposed protocol, user anonymity requires that the real identity ID_i of a registered user U_i keeps hidden from anyone, except the gateway node GWN . An intuitive strategy of achieving this goal is to encrypt all transmitted messages using a symmetric encryption algorithm. However, this forces each user to share a high-entropy key with the gateway node, which will bring heavy workload of managing these keys for the gateway node GWN . On the other hand, since the shared symmetric key is with high-entropy, the user U_i has to store it into the smart card. As a result, this mechanism would fail once the smart card is lost. In Farash et al.'s protocol, the authors adopt a similar approach. That is, all

users share the same key $h(X_{GWN})$ with the gateway node GWN , and all sensor nodes also share the same key $h(X_{GWN} \parallel 1)$ with the gateway node. This may be even worse since any malicious user can get other user's real identity from those publicly transmitted messages.

In our protocol, we employ a simple method to provide user anonymity. Specifically, the gateway node stores a tuple (t_i, TID_i) for each user U_i and assigns $TID_i = ID_i \oplus h(t_i \parallel X_{GWN})$ to U_i as its provisional identity. We note that each user's provisional identity is the same in all authentication procedures. This implies that although an adversary cannot get the real identity of a user, it can identify the user in different sessions. Therefore, our scheme provides weak user anonymity.

6. Performance Discussions

In this section, we evaluate the performance of the proposed protocol in terms of security property and computation cost by comparing it with other related works.

In Table 2, we summarize the security properties of the listed schemes. We can see that early schemes [6, 20, 38] are designed to only achieve user authentication, without considering the functionality of key agreement. In addition, as a special attack against

Table 2

Security comparisons with previous related works

Security properties	Das [6]	Khan et al. [20]	Yuan [38]	Farash et al. [7]	Kumari et al. [22]	Ours
Offline password guessing attack	X	X	X	X	✓	✓
Sensor node spoofing attack	X	X	X	X	✓	✓
User impersonation attack	X	X	X	X	✓	✓
Privileged insider attack	X	X	X	✓	✓	✓
Message replay attack	✓	✓	✓	✓	✓	✓
Mutual authentication	X	✓	✓	✓	✓	✓
Session key agreement	X	X	X	✓	✓	✓
Perfect forward secrecy	X	X	X	X	X	✓
Friendly password change	X	✓	✓	✓	✓	✓

[*] The symbol X indicates a scheme cannot resist the corresponding attack or cannot provide the corresponding security property. The symbol ✓ represents the contrary case.

Table 3

Performance comparisons with previous related works (unit: s)

Schemes	Use side	Sensor node side	Gateway node side	Total
Das [6]	$4 \cdot T_h \approx 0.00128$	$T_h \approx 0.00032$	$4 \cdot T_h \approx 0.00128$	$9 \cdot T_h \approx 0.00288$
Yuan [38]	$6 \cdot T_h + 2 \cdot T_e \approx 0.04032$	$2 \cdot T_h \approx 0.00064$	$6 \cdot T_h + T_e \approx 0.02112$	$14 \cdot T_h + 3 \cdot T_e \approx 0.06208$
Khan et al. [20]	$4 \cdot T_h \approx 0.00128$	$2 \cdot T_h \approx 0.00064$	$5 \cdot T_h \approx 0.0016$	$11 \cdot T_h \approx 0.00352$
Farash et al. [7]	$11 \cdot T_h \approx 0.00352$	$7 \cdot T_h \cdot 0.00224$	$14 \cdot T_h \approx 0.00448$	$32 \cdot T_h \approx 0.01024$
Kumari et al. [22]	$17 \cdot T_h \approx 0.00544$	$9 \cdot T_h \approx 0.00288$	$18 \cdot T_h \approx 0.00576$	$44 \cdot T_h \approx 0.01408$
Ours	$7 \cdot T_h + 2 \cdot T_e \approx 0.03872$	$7 \cdot T_h + 2T_e \approx 0.03872$	$10 \cdot T_h + 2 \cdot T_e \approx 0.0416$	$24 \cdot T_h + 6 \cdot T_e \approx 0.12288$

[*] T_e = the running time of one exponentiation operation. T_h = the running time of one hash operation.

WSNs, except Kumari et al.'s [22] and our scheme, all listed schemes suffer from sensor node spoofing attack. Note that all listed schemes are two-factor authentication protocols based on smart card and password. This implies that these schemes should remain secure even if the secret values stored in the smart card are revealed. However, all these schemes, except our scheme, are vulnerable to offline password guessing attack when the smart card is lost. That is, they fail to achieve the required security guarantee of the two-factor authentication scheme. In addition, by introducing Diffi-Hellman key exchange, only our scheme can provide perfect forward secrecy, which ensures the security of previously used session keys when the gateway node is corrupted.

Table 3 presents the computation cost of each protocol participant in each listed scheme. These schemes mainly involve two kinds of cryptographic operations, namely, exponentiation operation and hash operation¹. To be precise, the running time of a hash operation and an exponentiation operation is roughly 0.00032 s and 0.0192 s [24, 8], respectively. Das et al.'s [6] scheme is the most efficient one. However, the development process of this kind of two-factor authentication scheme demonstrates that security is the first goal and major motivation of designing such an authentication. Even though our scheme consumes more computation resource, it overcomes security

¹ Relatively, since the running time of XOR operation is nearly negligible, we thus ignore it.

weaknesses in previous works and provides the required security properties. On the other hand, with the rapid development of information technology, the computation capacity of smart card and sensor node has being enhanced, which enables the computation cost of our scheme to be acceptable for practical applications.

7. Conclusion

In this study, we first briefly review user authentication schemes proposed by Farash et al. [7] and Kumari et al. [22], respectively, and further demonstrate that their schemes fail to achieve intended security properties. To remedy the security loopholes in the above two schemes, we have proposed a novel user authentication and key agreement scheme for WSNs. Security analysis shows that our proposal can resist various well known attacks and provide perfect forward secrecy. Furthermore, in order to examine the performance of our scheme, we compared it with other related works. The comparison results indicate that our scheme is efficient enough, while providing more security guarantees. Thus, it is more feasible for practical applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgements

Wenfen Liu, Jianghong Wei and Xuexian Hu were supported in part by the National Nature Science Foundation of China under Grants 61702549 and 61502527, and in part by the Open Foundation of State Key Laboratory of Integrated Services Networks (Xid-

ian University) under Grant ISN19-12. Saru Kumari was supported by the University Grants Commission, India, through UGC-BSR Research Start-up grant under Grant no. 3(A)(60)31, and in part by Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-14-004.

References

- Althobaiti, O., Alrodhaan, M., Aldhelaan, A. An Efficient Biometric Authentication Protocol for Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 2013, Article ID: 407971. <https://doi.org/10.1155/2013/407971>
- Bellare, M., Pointcheval, D., Rogaway, P. Authenticated Key Exchange Secure Against Dictionary Attacks. *Advances in Cryptology—EUROCRYPT 2000*, Belgium, May 14-18, 2000, 139-155. https://doi.org/10.1007/3-540-45539-6_11
- Chen, B., Kuo, W., Wu, L. Robust Smart-Card-Based Remote User Password Authentication Scheme. *International Journal of Communication Systems*, 2014, 27(2), 377-389. <https://doi.org/10.1002/dac.2368>
- Das, A. K., Sharma, P., Chatterjee, S., Sing, J. K. A Dynamic Password-Based User Authentication Scheme for Hierarchical Wireless Sensor Networks. *Journal of Network and Computer Applications*, 2012, 35(5), 1646-1656. <https://doi.org/10.1016/j.jnca.2012.03.011>
- Das, A. K., Sutrala, A. K., Kumari, S., Odelu, V., Wazid, M., Li, X. An Efficient Multi-Gateway-Based Three-Factor User Authentication and Key Agreement Scheme in Hierarchical Wireless Sensor Networks. *Security and Communication Networks*, 2016, 9(13), 2070-2092. <https://doi.org/10.1002/sec.1464>
- Das, M. L. Two-Factor User Authentication in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 2009, 8(3), 1086-1090. <https://doi.org/10.1109/TWC.2008.080128>
- Farash, M. S., Turkanovic, M., Kumari, S., Hölbl, M. An Efficient User Authentication and Key Agreement Scheme for Heterogeneous Wireless Sensor Network Tailored for the Internet of Things Environment. *Ad Hoc Networks*, 2016, 36, 152-176. <https://doi.org/10.1016/j.adhoc.2015.05.014>
- He, D., Kumar, N., Lee, J. H., Sherratt, R. S. Enhanced Three-Factor Security Protocol for Consumer USB Mass Storage Devices. *IEEE Transactions on Consumer Electronics*, 2014, 60(1), 30-37. <https://doi.org/10.1109/TCE.2014.6780922>
- He, D., Kumar, N., Shen, H., Lee, J. One-to-Many Authentication for Access Control in Mobile Pay-TV Systems. *SCIENCE CHINA Information Sciences*, 2016, 59(5), 1-14. <https://doi.org/10.1007/s11432-015-5469-5>
- He, D., Wang, D. Robust Biometrics-Based Authentication Scheme for Multiserver Environment. *IEEE Systems Journal*, 2015, 9(3), 816-823. <https://doi.org/10.1109/JSYST.2014.2301517>
- He, D., Zeadally, S., Kumar, N., Lee, J. H. Anonymous Authentication for Wireless Body Area Networks With Provable Security. *IEEE Systems Journal*, 2017, 11(4), 2590-2601. <https://doi.org/10.1109/JSYST.2016.2544805>
- He, D., Zeadally, S., Kumar, N., Wu, W. Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-server Architectures. *IEEE Transactions on Information Forensics and Security*, 2016, 11(9), 2052-2064. <https://doi.org/10.1109/TIFS.2016.2573746>
- Hu, X., Zhang, Z., Zhang, Q. Universally Composable Three-party Password-authenticated Key Exchange With Contributiveness. *International Journal of Communication Systems*, 2015, 28(6), 1100-1111. <https://doi.org/10.1002/dac.2746>
- Hwang, M., Chong, S., Chen, T. DoS-resistant ID-based Password Authentication Scheme Using Smart Cards. *Journal of Systems and Software*, 2010, 83(1), 163-172. <https://doi.org/10.1016/j.jss.2009.07.050>
- Jiang, Q., Khan, M. K., Lu, X., Ma, J., He, D. A Privacy Preserving Three-factor Authentication Protocol for E-health Clouds. *The Journal of Supercomputing*, 2016, 72(10), 3826-3849. <https://doi.org/10.1007/s11227-015-1610-x>
- Jiang, Q., Kumar, N., Ma, J., Shen, J., He, D., Chilamkurti, N. A Privacy-aware Two-factor Authentication Protocol Based on Elliptic Curve Cryptography for Wireless Sensor Networks. *International Journal of Network Management*, 2016, 27(3), e1937. <https://doi.org/10.1002/nem.1937>

17. Jiang, Q., Ma, J., Li, G., Li, X. Improvement of Robust Smart-Card-Based Password Authentication Scheme. *International Journal of Communication Systems*, 2015, 28(2), 383-393. <https://doi.org/10.1002/dac.2644>
18. Jiang, Q., Ma, J., Lu, X., Tian, Y. An Efficient Two-factor User Authentication Scheme With Unlinkability for Wireless Sensor Networks. *Peer-to-Peer Networking and Applications*, 2015, 8(6), 1070-1081. <https://doi.org/10.1007/s12083-014-0285-z>
19. Jiang, Q., Ma, J., Wei, F., Tian, Y., Shen, J., Yang, Y. An Untraceable Temporal-Credential-Based Two-Factor Authentication Scheme Using ECC for Wireless Sensor Networks. *Journal of Network and Computer Applications*, 2016, 76, 37-48. <https://doi.org/10.1016/j.jnca.2016.10.001>
20. Khan, M. K., Alghathbar, K. Cryptanalysis and Security Improvements of 'Two-Factor User Authentication in Wireless Sensor Networks'. *Sensors*, 2010, 10(3), 2450-2459. <https://doi.org/10.3390/s100302450>
21. Kocher, P. C., Jaffe, J., Jun, B. Differential Power Analysis. *Advances in Cryptology-CRYPTO 1999*, California, USA, August 15-19, 1999, 388-397. https://doi.org/10.1007/3-540-48405-1_25
22. Kumari, S., Das, A. K., Wazid, M., Li, X., Wu, F., Choo, K. R., Khan, M. K. On the Design of a Secure User Authentication and Key Agreement Scheme for Wireless Sensor Networks. *Concurrency and Computation: Practice and Experience*, 2017, 29(23), e3930. <https://doi.org/10.1002/cpe.3930>
23. Kumari, S., Khan, M. K. Cryptanalysis and Improvement of 'A Robust Smart-Card-Based Remote User Password Authentication Scheme'. *International Journal of Communication Systems*, 2014, 27(12), 3939-3955. <https://doi.org/10.1002/dac.2590>
24. Lee, C. C., Chen, C. T., Chen, P. Three-Factor Control Protocol Based on Elliptic Curve Cryptosystem for Universal Serial Bus Mass Storage Devices. *IET Computers & Digital Techniques*, 2013, 7(1), 48-55. <https://doi.org/10.1049/iet-cdt.2012.0073>
25. Liao, I., Lee, C., Hwang, M. A Password Authentication Scheme over Insecure Networks. *Journal of Computer System Sciences*, 2006, 72(4), 727-740. <https://doi.org/10.1016/j.jcss.2005.10.001>
26. Lu, Y., Li, L., Yang, Y. Robust and Efficient Authentication Scheme for Session Initiation Protocol. *Mathematical Problems in Engineering*, 2015, Article ID: 894549.
27. Messerges, T. S., Dabbish, E. A., Sloan, R. H. Examining Smart-Card Security Under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, 2002, 51(5), 541-552. <https://doi.org/10.1109/TC.2002.1004593>
28. Shi, W., Gong, P. A New User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *International Journal of Distributed Sensor Networks*, 2013, Article ID: 730831. <https://doi.org/10.1155/2013/730831>
29. Song, R. Advanced Smart Card Based Password Authentication Protocol. *Computer Standards & Interfaces*, 2010, 32(5), 321-325. <https://doi.org/10.1016/j.csi.2010.03.008>
30. Turkanovic, M., Brumen, B., Holbl, M. A Novel User Authentication and Key Agreement Scheme for Heterogeneous Ad Hoc Wireless Sensor Networks Based on the Internet of Things Notions. *Ad Hoc Networks*, 2014, 20, 96-112. <https://doi.org/10.1016/j.adhoc.2014.03.009>
31. Wang, D., He, D., Wang, P., Chu, C. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals are Beyond Attainment. *IEEE Transactions on Dependable and Secure Computing*, 2015, 12(4), 428-442. <https://doi.org/10.1109/TDSC.2014.2355850>
32. Wang, D., Ma, C. Cryptanalysis of a Remote User Authentication Scheme for Mobile Client-Server Environment Based on ECC. *Information Fusion*, 2013, 14(4), 498-503. <https://doi.org/10.1016/j.inffus.2012.12.002>
33. Wang, D., Wang, N., Wang, P., Qing, S. Preserving Privacy for Free: Efficient and Provably Secure Two-factor Authentication Scheme With User Anonymity. *Information Sciences*, 2015, 321, 162-178. <https://doi.org/10.1016/j.ins.2015.03.070>
34. Wang, D., Wang, P. Understanding Security Failures of Two-factor Authentication Schemes for Real-time Applications in Hierarchical Wireless Sensor Networks. *Ad Hoc Networks*, 2014, 20, 1-15. <https://doi.org/10.1016/j.adhoc.2014.03.003>
35. Wei, J., Hu, X., Liu, W. An Improved Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 2012, 36(6), 3597-3604. <https://doi.org/10.1007/s10916-012-9835-1>
36. Xu, J., Zhu, W. T., Feng, D. An Improved Smart Card Based Password Authentication Scheme with Provable Security. *Computer Standards & Interfaces*, 2009, 31(4), 723-728. <https://doi.org/10.1016/j.csi.2008.09.006>
37. Xue, K., Ma, C., Hong, P., Ding, R. A Temporal-Credential-Based Mutual Authentication and Key Agreement Scheme for Wireless Sensor Networks. *Journal of Network and Computer Applications*, 2013, 36(1), 316-323. <https://doi.org/10.1016/j.jnca.2012.05.010>
38. Yuan, J. J. An Enhanced Two-Factor User Authentication in Wireless Sensor Networks. *Telecommunication Systems*, 2014, 55(1), 105-113. <https://doi.org/10.1007/s11235-013-9755-5>