Peer-to-Peer Networking and Applications (2019) 12:43–59
https://doi.org/10.1007/s12083-017-0627-8

Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks

AmirHosein Adavoudi-Jolfaei[1,2] · Maede Ashouri-Talouki[1,2] · Seyed Farhad Aghili[1,2]

Received: 14 April 2017 / Accepted: 12 December 2017 / Published online: 21 December 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

Abstract
Wireless sensor networks (WSNs) play an important role and support a variety of real time applications, such as healthcare monitoring, military surveillance, vehicular tracking and, so on. Secure and real time information accessing from the sensor nodes in these applications is very important. Because wireless sensor nodes are limited in computing and communication capabilities and data storage, it is very crucial to design an effective and secure lightweight authentication and key agreement scheme. Recently, Gope et al. proposed a realistic lightweight anonymous authentication scheme in WSNs and claimed that their scheme satisfied all security concerns in these networks. However, we show that in their scheme the adversary can obtain the session key between the user and the sensor node. In order to fix this drawback, we propose an improved three-factor authentication scheme which is more suitable than Gope et al.'s scheme and also provides more desired security properties such as three-factor authentication and access control. Through the informal analysis, we show that our scheme is secure against various known attacks including the attack found in Gope et al.'s scheme. Furthermore, we have demonstrated the validity of our proposed scheme using the BAN logic. As compared with the previous authentication schemes, the proposed scheme is not only more secure but also enough practical and competitive with existing schemes.

Keywords Wireless sensor networks · User anonymity · Three-factor authentication · Key agreement · Access control

1 Introduction

A wireless sensor network consists of low-cost, low-power sensor nodes deployed over a geographical area for controlling physical events like temperature, humidity, vibrations, seismic, and so on [2, 35]. WSNs are powerful in that they are amenable to support a lot of very different real world applications (e.g., disaster relief, environment control and biodiversity mapping, intelligent buildings, medicine and healthcare, machine surveillance and preventive maintenance) [30]. Some of these applications are critical scenarios including battlefield and security applications [18]. It is obvious that there should have great needs to access the real time data in these critical scenarios. The data are to be accessed directly by the external users as and when demanded, so authentication of the user must be ensured before allowing the user to access data [14]. The gateway node (GW) plays a crucial role in the WSNs as all data transmitted to the outside network must pass through it. Generally, registered users regularly log in to and query a WSN via GW. However, it is impractical or inconvenient to access real time data from the sensor nodes through GW only. In these networks, users need to have direct access to the sensor nodes to acquire data from them whenever required. Because of this reason, the user authentication problem is a very important research topic in WSN security which has received considerable research attention in WSN security study in the recent years [8, 9]. In general, there are two approaches in authenticating users in WSN:

– A user is authenticated by the GW before accessing nodes.

✉ Maede Ashouri-Talouki
m.ashouri@eng.ui.ac.ir

AmirHosein Adavoudi-Jolfaei
a_adavoudi@eng.ui.ac.ir

Seyed Farhad Aghili
sf.aghili@eng.ui.ac.ir

[1] Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

[2] Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran

– A user directly contacts a sensor node and performs authentication with it [7].

Since nodes are limited in terms of computation and communication capabilities, lightweight authentication and key agreement protocols are prefered. In this direction, a number of two-factor user authentication schemes have been proposed in the literature [8, 14, 20, 25, 27, 31, 36, 37, 40, 43]. However, most of them have been demonstrated either insecure against different known attacks or lack of some important features. Recently, Gope et al. proposed a realistic lightweight anonymous authentication scheme for WSN and claimed that their scheme is secure against all known attacks [24]. However, in this paper, we analyze this scheme and show that their scheme is vulnerable to session key disclosure. In order to remedy this vulnerability, we improve the Gope et al.'s scheme and enhance its security features using three-factor user authentication along with access control property. In our scheme, we use a user's personal biometric as the third factor. In recent years, the research shows that biometric based user authentication schemes are more secure and reliable. There are the following major advantages of using biometric key over traditional passwords [33]:

– Biometric keys can not be guessed easily;
– Biometric keys are very difficult to copy or share;
– Biometric keys can not be lost or forgotten;
– Biometric keys are extremely hard to forge or distribute;
– Someone's biometrics is not easy to break than others.

Moreover, in this paper, we propose a new user access control to allow authorized users with the relevant groups to access the real-time information from the WSN for which they are permitted. For example in the battlefield scenario, a commander should be able to access all types of data for the purpose of overall coordinating, but a soldier may only need to access the type of data relevant to his/her mission. Considering this point, the importance of user access control in WSNs for any applications becomes an important research field.

1.1 Our contribution

- We first analyze the security of the recently proposed Gope et al.'s scheme [24], and find that their scheme has a vulnerability.
- In order to remedy the vulnerability found in Gope et al.'s scheme, we improve the scheme and enhance its security features as follow:
  - Our scheme makes use of three factors, namely user's password and biometric along with the non-tamper resistant smart card.
  - Our scheme makes use of access control which lets users from different groups access permitted sensor's information.
- We analyze the efficiency of our proposed scheme, and show our scheme is also efficient as compared to other schemes.
- Our scheme is shown to be secure against relevant known attacks through both informal and formal security analysis.

2 Related work

Watro et al. [46] proposed a user authentication in WSNs which is known as TinyPK. Their schemes employed RSA [39] and Diffie-Hellman [15] algorithms to calculate an encrypted public key. TinyPK has a security flow as pointed out in [14]. Wong et al. [47] proposed a hash-based user authentication scheme, but some researchers found that is vulnerable to stolen-verifier, replay and forgery attacks [14]. Authors in [14] introduced a two-factor method of user authentication, which uses password and smart card of a user. However, that scheme cannot resist denial of service attack and node compromise attack [13]. Many researchers proposed several improvements [8, 32, 36, 38, 49] which inspired from [14].

Fan et al. [19] observed that two-factor authentication schemes [14, 49] have various defeats overlooked for real time data access and they proposed an efficient and denial of service resistant user authentication scheme which only employs lightweight cryptographic operations, such as hash functions and exclusive-OR. More recently, Vaidya et al. [42] identified some security weakness in [8, 14, 31]. These weaknesses are under the assumption that an adversary can extract parameters from the smart card or a smart card is lost or stolen. He et al. [25] proposed an improvement on Das's scheme [14]. Chen and shih [8] showed that Das's scheme [14] fails to mutual authentication so they proposed a lightweight authentication scheme, but their protocol is vulnerable to attacks such as forgery attack and replay attack [36]. Additionally, Amin et al. [1] proposed a three-factor user authentication protocol for WSNs using password, smart card and biometric. Later, in [3] the authors proved that Amin et al. protocol is vulnerable to replay and Denial-of-Service (DoS) attacks.

Xue et al.in [48] proposed a temporal credential-based mutual authentication which is based on hash function. However, their scheme is vulnerable to privileged-insider attack, tracking attack, stolen smart card attack and so on [29, 45]. Jiang et al. [29] and K Das et al. [11] used a temporary identity which helps the GW to comprehend

exactly who is the user, but their schemes are vulnerable to DoS attack [22, 44]. Das et al. [10] proposed an efficient user anonymity authentication scheme, however in this scheme the GW needs to perform an exhaustive search operation for finding the user's identity. This scheme also is vulnerable to DoS attack [24]. Gope et al. demonstrated the schemes proposed in [10, 11, 29, 48, 49] support session key agreement between the user and the sensor node, however none of the scheme can ensure perfect forward secrecy (PFC) [24].

Gope et al. [24] proposed a realistic lightweight anonymous authentication protocol. This scheme provides kinds of important properties such as user anonymity, PFC, etc.The author claimed that their scheme is secure against known attacks. However in this paper we analyze the Gope's scheme and demonstrate that unfortunately their scheme is vulnerable to session key disclosure which is the goal of this user authentication scheme. The details are discussed in Section 5.

Paper organization The remainder of this paper is organized as follows. The next section introduces the preliminary of fuzzy extractor briefly. Section 4 reviews Gope et al.'s scheme. Section 5 elaborates the drawback of their scheme. Section 6 presents the improved authentication and key agreement scheme. In Sections 7 and 8, the informal and formal security and efficiency of the proposed scheme are discussed. Finally, in Section 9, we conclude the paper.

3 Preliminaries and notations

3.1 Notations

In this section we describe the notations used in this paper (Table 1).

3.2 Preliminaries

Given biometric input B, a fuzzy extractor capables of extracting an almost random string RS from the biometric template B in an error-tolerant way, which means the fuzzy extractor could output the same random string when the input changes with the help of an auxiliary string P. The fuzzy extractor is composed of the following two algorithms, called Gen and Rep:

1. Gen is a probabilistic algorithm. Upon receiving biometric input B, the algorithm will output a secret data key RS and a random auxiliary string P as follows: $Gen(B) = (RS, P)$.

Table 1 Notations

Notation	Description
U	User
GW	Gateway
SC	Smart card
SN	Sensor
ID_u	Identity of the user
AID_u	One-time-alias identity of the user
SID	Shadow identity of user
ID_G	Secret identity of the gateway
w	Secret key of the gateway
SN_{id}	Identity of the sensor node
PSW_u	Password of the user
B_u	Biometric of the user
N_u	Random number generated by the user
SK	Session key between SN and U
APM	A set of user U's access privilege masks
G	A set of user U's group ids
K_{ug}	Shared key between U and GW
KEM_{ug}	Shared emergency key between U and GW
K_{gs}	Secret key shared between the GW and SN
Ts_{ug}	Transaction sequence number
$h(.)$	One-way hash function
$\oplus$	XOR operation
$\|$	Concatenation operation

2. Rep is a deterministic algorithm which takes a noisy biometric B^* and the corresponding random auxiliary P, and then recovers the biometric secret data key RS as follows: $Rep(B^*, P) = RS$.

4 Review of Gope's scheme

In this section, we review Gope's authentication protocol [24] based on XOR and hash functions, which is composed of four phases, i.e., registration, anonymous authentication and key exchange, password renewal, and dynamic node addition phase.

4.1 Registration phase

U performs the following steps with GW through a secure channel, as is shown in Fig. 1.

Step 1. U submits his identity ID_u to the GW.

Step 2. GW generates the random number (n_g) and computes $K_{ug} = h(ID_u \| n_g) \oplus ID_G$ and also generates a set of unlinkable shadow-IDs and a set of emergency keys $SID = \{sid_1, sid_2, \ldots\}$ and $KEM_{ug} =$

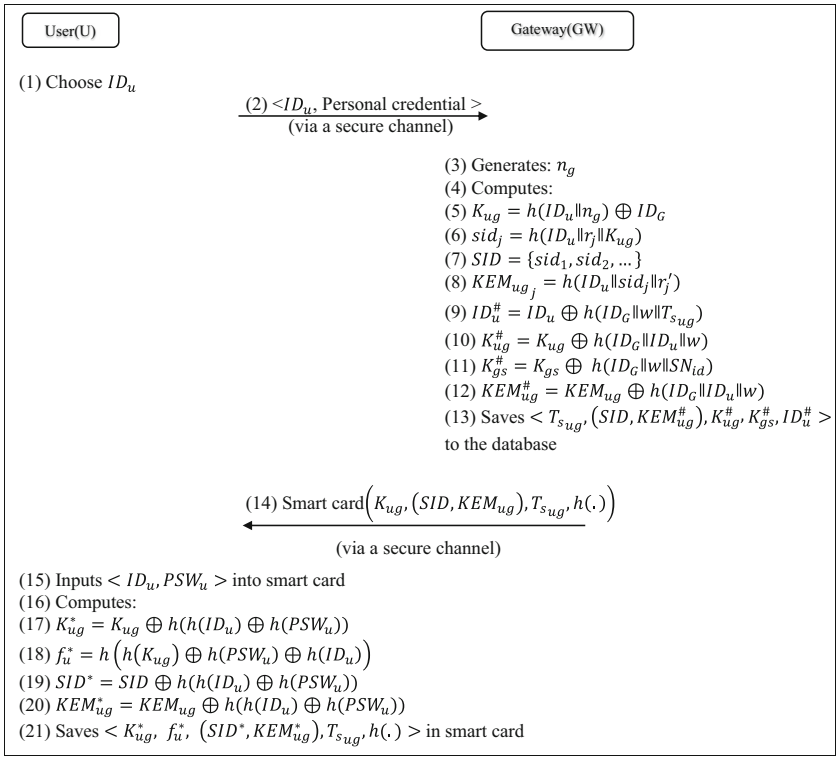

Fig. 1 User registration phase of Gope et al.'s scheme [24]

$\{KEM_{ug_1}, KEM_{ug_2}, \ldots\}$, respectively. Hereafter, the GW generates a sequence number of 64-bit Ts_{ug}. This sequence number will be computed based on the number of requests (m) handled by the GW, including the present request of the current user, then GW sets $Ts_{ug} = m$. This parameter is used by GW to speed up the authentication process, where by comparing it with the stored value of its database, the GW can define exactly who is the user. In the case of the user and the GW have been asynchronized, the user will send a pair of shadow-ID sid_j and the emergency key KEM_{ug_j} to the GW. The pair of (sid_j, KEM_{ug_j}) must be deleted from the list by both the U and GW.

Step 3. The GW issues a smart card with $\{K_{ug}, (SID, KEM_{ug}), Ts_{ug}, h(.)\}$ to U. The GW uses its secret id ID_G, the secret key w(stored in secure ROM-BIOS of the GW) and other parameters to encode ID_u, K_{gs}, K_{ug} and KEM_{ug} as depicted in lines 9 to 12 of Fig. 1 and stores $ID_u^{\sharp}$, $K_{ug}^{\sharp}$, $K_{gs}^{\sharp}$, $(SID, KEM_{ug}^{\sharp})$, and Ts_{ug} in its database.

Step 4. U chooses a password PSW_u and computes K_{ug}^*, f_u^*, SID^* and KEM_{ug}^* (lines 17-20). Finally the smart card contains the tuple shown in line 21.

4.2 Anonymous authentication and key exchange phase

In this phase, U and SN are mutually authenticated. At the end of this phase, a session key is established between the user U and the sensor node SN_{id}. The following steps are performed as follows shown in Fig. 2.

Step 1. U inserts his smart card to a terminal, and enters his identity ID_u and password PSW_u. The smart card computes K_{ug}, f_u as depicted in lines 2 to 3 of Fig. 2 and checks whether the condition $f_u = f_u^{\star}$ holds or not. If it holds, the smart card generates a random number N_u and computes one-time alias identity AID_u, N_x, and V_1 (lines 4-6). Finally, the user forms a request message $M_{A_1} : U \rightarrow GW : \{AID_u, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$. Note: In case of loss of synchronization between user and GW, the user need not to send any transaction sequence number Ts_{ug} in M_{A_1}. In that case, the user needs to choose one of the unused pair of $(sid_j^*, KEM_{ug_j}^*)$ from (SID^*, KEM_{ug}^*) and then submits his ID_u and PSW_u and computes sid_j and KEM_{ug_j} as depicted in lines 8 to 10, then user U assigns the sid_j as AID_u and KEM_{ug_j} as K_{ug}.

Step 2. Upon receiving the message M_{A_1} from user, the GW checks whether Ts_{ug} is valid or not. Since the GW maintains the most recent transaction sequence number for each user, when the GW finds Ts_{ug} in its database then it selects that tuple and uses its secret id ID_G and the secret key w to decode the identity ID_u and K_{ug} of the user. Hence, the GW can identify exactly who is the user. Then the GW checks the validity of V_1. If so, the GW first computes $N_u = K_{ug} \oplus N_x$, and then checks AID_u, if the verification of AID_u is not successful the

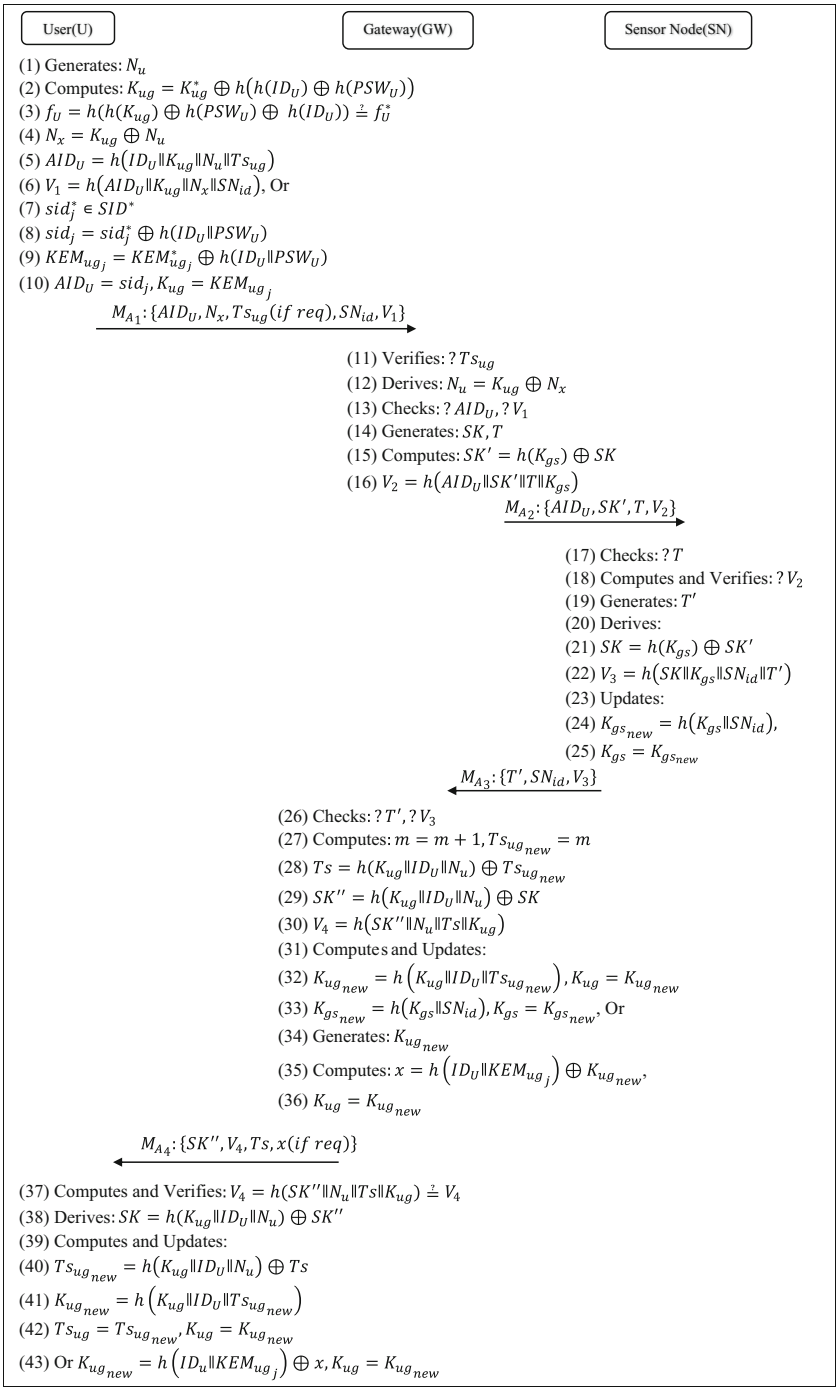

Fig. 2 Authentication and key agreement phase of Gope et al.'s scheme [24]

GW terminates the connection, otherwise GW generates a session key SK and a timestamp T randomly. Then, the GW computes SK', V_2 as represented in lines 15 to 16 and sends message $M_{A_2} : GW \rightarrow SN : \{AID_u, SK', T, V_2\}$ to the sensor node SN_{id}. Note: If the GW cannot find the Ts_{ug} provided by the user in M_{A_1} in its database, it terminates the connection.

Step 3. After receiving the message M_{A_2}, SN_{id} checks the T and V_2 (lines 17-18). If both of them are valid, SN_{id} computes $SK = h(K_{gs}) \oplus SK'$, generates a timestamp T' and computes V_3 (lines 19-22). Hereafter, SN_{id} forms message $M_{A_3} : SN \rightarrow GW : \{T', SN_{id}, V_3\}$ and sends it to the GW. Finally the sensor node computes $K_{gs_{new}}$ and updates its shared secret key as $K_{gs} = K_{gs_{new}}$ (lines 24-25). Note: In case of loss of synchronization between SN_{id} and GW, the sensor node needs to ask GW for the new secret shared key, i.e, $K_{gs_{new}}$, which will be securely sent to the sensor node.

Step 4. Upon receiving the message M_{A_3}, the GW checks T', computes V_3 and checks whether it holds or not (lines 26). If so, then the GW checks the latest value of the transaction sequence number m and computes $m \leftarrow m + 1$ (line 27). Then GW saves $Ts_{ug_{new}}$ in its database and computes T_s, V_4 and SK'' as depicted in lines 28 to 30. Then, the GW forms message $M_{A_4} : GW \rightarrow U : \{SK'', V_4, T_s, x(ifreq)\}$. Finally the GW computes $K_{ug_{new}}$ and $K_{gs_{new}}$ (lines 32-33). Note: If the GW cannot find any Ts_{ug} in M_{A_1}, then the GW will try to recognize the sid_j in AID_u by comparing with the entries in its database. If GW can find sid_j, it retrieves KEM_{ug_j} associated with sid_j, and then validates V_1. And at the end, GW randomly generates a new shared key $K_{ug_{new}}$, and encodes it by KEM_{ug_j} and the user identity ID_u as depicted in line 35 and computes $V_4 = h(SK''\|N_u\|T_s\|x)$, then GW sends x and V_4 with other parameters in M_{A_4} to U. When the user receives the message M_{A_4}, the terminal computes V_4 and verifies if it holds or not (line 37). If so, the smart card computes $Ts_{ug_{new}}$ and $K_{ug_{new}}$ as depicted in lines 40 to 41 and then saves $K_{ug} = K_{ug_{new}}$ and $Ts_{ug} = Ts_{ug_{new}}$ for further communication (lines 42-43).

4.3 Password update phase

In this phase, U executes the following steps to update the password. The user needs to enter his identity ID_u, old password PSW_u, and the new password PSW_u^* to the smart card. The smart card will retrieve K_{ug}, KEM_{ug} and SID as follows.

- $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u))$;
- $KEM_{ug} = KEM_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u))$;
- $SID = SID^* \oplus h(h(ID_u) \oplus h(PSW_u))$.

It then computes K_{ug}^{**}, SID^{**} and SID^{**} as bellow.

- $K_{ug}^{**} = K_{ug} \oplus h(h(ID_u) \oplus h(PSW_u^*))$;
- $SID^{**} = SID \oplus h(h(ID_u) \oplus h(PSW_u^*))$;
- $KEM_{ug}^{**} = KEM_{ug} \oplus h(h(ID_u) \oplus h(PSW_u^*))$.

Finally, the smart card will replace K_{ug} with K_{ug}^{**}, SID with SID^{**}, and KEM_{ug} with KEM_{ug}^{**}.

4.4 Dynamic node addition phase

In this phase, a fresh sensor node will be deployed to the target field in order to continue the services in WSN. The GW generates an identity $SN_{id_i}^{new}$ and a key $K_{gs_i}^{new}$ for SN_i^{new}. Then the GW saves these parameters in the memory of SN_i^{new}. Hereafter, the GW computes $K_{gs_i}^{new*} = K_{gs_i}^{new} \oplus h(ID_G\|w\|SN_{id_i}^{new})$ and saves both $SN_{id_i}^{new}$ and $K_{gs_i}^{new*}$ in its database.

5 Security analysis of the Gope et al.'s protocol

Before analyzing the security of Gope et al.'s scheme [24], we should define the threat model which is based on the Dolev-Yao model [17]. Under this model, an adversary can intercept all messages transmitted through the channel and adversary can modify, delete or change the contents of the transmitted messages. The adversary has the ability to obtain the secret information stored in the smart card by side channel attacks. If an attacker captures a sensor node then he/she can know all the security parameters stored in the sensor's node memory.

In Gope et al.'s protocol, an adversary A can reveal the session key between the user U and the sensor node SN through the following scenario.

Step 1. A eavesdrops the message $M_{A_1} : U \rightarrow GW : \{AID_u, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$ sent by U, then he extracts the Ts_{ug} value from M_{A_1} which is equal to m.

Step 2. A eavesdrops the message $M_{A_4} : GW \rightarrow U : \{SK'', V_4, T_s, x(ifreq)\}$ and then extracts the T_s value which is equal to $T_s = h(K_{ug}\|ID_u\|N_u) \oplus Ts_{ug_{new}}$. As mentioned in Step 4 of authentication and key exchange phase of this paper, the GW checks the last value of the transaction sequence parameter m and computes $m \leftarrow m + 1$. Then GW stores $Ts_{ug_{new}} = m$ and computes $T_s = h(K_{ug}\|ID_u\|N_u) \oplus Ts_{ug_{new}}$.

Step 3. A can compute the $Ts_{ug_{new}}$ value just by incrementing the Ts_{ug} value obtained from Step 1 as: $Ts_{ug_{new}} = Ts_{ug} + 1$. Then A computes $h(K_{ug}\|ID_u\|N_u)$ using Ts in M_{A_4} message and $Ts_{ug_{new}}$ obtained from previous step as: $h(K_{ug}\|ID_u\|N_u) = Ts \oplus Ts_{ug_{new}}$.

Step 4. Finally adversary can reveal the session key SK using the SK'' in M_{A_4} message and $h(K_{ug}\|ID_u\|N_u)$ value obtained from Step 3 by using equation $SK = SK'' \oplus h(K_{ug}\|ID_u\|N_u)$.

Thus, Gope et al.'s protocol is vulnerable against session key disclosure attack.

6 Our proposed protocol

In this section, we describe an improvement of Gope et al.'s scheme in order to withstand the drawback found in their scheme. Our proposed scheme is a three-factor authentication scheme, which uses a user's personal biometric as compared to Gope et al.'s two-factor authentication scheme. Unfortunately, in two-factor authentication, smart cards may be lost or stolen, and the data stored in the smart card can be extracted or passwords

are vulnerable to off-line guessing attack, phishing, etc [28, 50]. Recently, biometric-based user authentication schemes along with passwords have drawn considerable attention in research [12, 33, 34]. Moreover, we add access control feature to allow legitimate users from different groups to access data they have privilege. So, we improve Gope's scheme in the following aspects.

1. We revise the authentication and key agreement phase to resist session key disclosure attack.
2. In our scheme, we use three-factor authentication and access control to strengthen the security and add features which are not provided by Gope et al.'s scheme. Our scheme keeps the original merits of Gope et al.'s scheme.

Like Gope's scheme, our scheme also consists of four phases: registration phase, anonymous authentication and key exchange phase, password and biometric update phase and dynamic node addition phase.

6.1 Registration phase

To register to the GW(a trusted entity in the network) in WSN by a legal user U the following steps need to be executed through a secure channel, as is shown in Fig. 3.

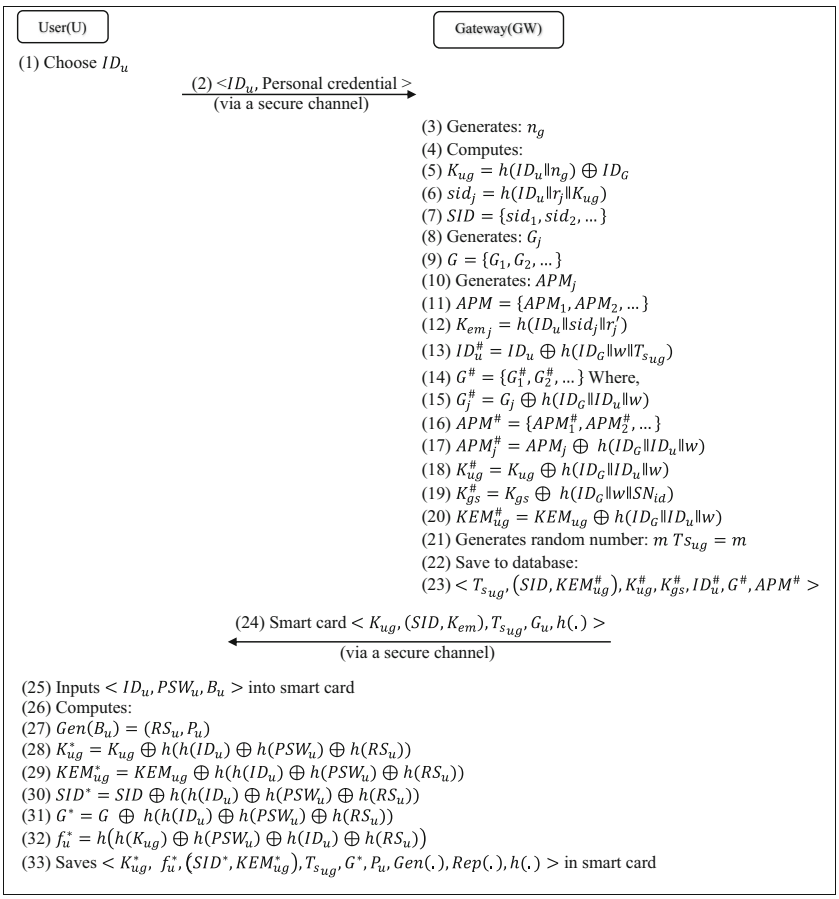

Fig. 3 Registration phase of our scheme

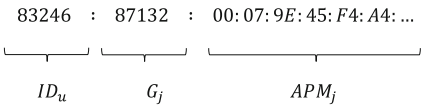

Fig. 4 An example of user access list. APM is a bitmap, for example, if the first bit of APM represents the 'temperature' parameter, an '1' in this bit indicates that the 'temperature' parameter is available for all members of this group

Step 1. U sends his identity ID_u to the GW.

Step 2. GW generates the random number n_g and computes $K_{ug} = h(ID_u \| n_g) \oplus ID_G$ and also generates a set of shadow-IDs $SID = \{sid_1, sid_2, \ldots\}$, and a set of emergency keys $KEM_{ug} = \{KEM_{ug_1}, KEM_{ug_2}, \ldots\}$, where $sid_j = h(ID_u \| r_j \| K_{ug})$ and $KEM_{ug_j} = h(ID_u \| sid_j \| r_j')$. Then, the GW generates a random sequence number of 64-bit Ts_{ug}. This parameter is used by GW to define exactly who is the user. Depending on the probable user query, the GW prepares a user access list pool. The access list defines the user's access privilege. A typical access list is composed of ID_u, G_j and user access privilege mask APM_j as is shown in Fig. 4. G_j is a unique random number used to identify a particular access group. Multiple users who have similar

task and access privilege can be organized in to the same group. A user can be member of one or more groups. User access privilege mask is a number of binary bit represents a specific information or service. Then, GW generates a set of group-IDs $G = \{G_1, G_2, \ldots\}$, and a set of access privilege masks $APM = \{APM_1, APM_2, \ldots\}$, where $G_j \in G$ is a 128-bit random number and $APM_j \in APM$ is a 128-bit random number except first 16-bits (high order) which each bit defines different task or service. It is also to be noted that each $APM_j \in APM$ corresponds to a particular $G_j \in G$.

Step 3. The GW will encode K_{ug}, K_{gs}, KEM_{ug}, ID_u, G and APM as depicted in lines 13 to 20 of Fig. 3 and then stores these values in its database. Hereafter, the GW issues a smart card with $\{K_{ug}, (SID, KEM_{ug}), Ts_{ug}, G, h(.)\}$ to U.

Step 4. After receiving smart card SC, U chooses a password PSW_u and imprints the biometric B_u. U

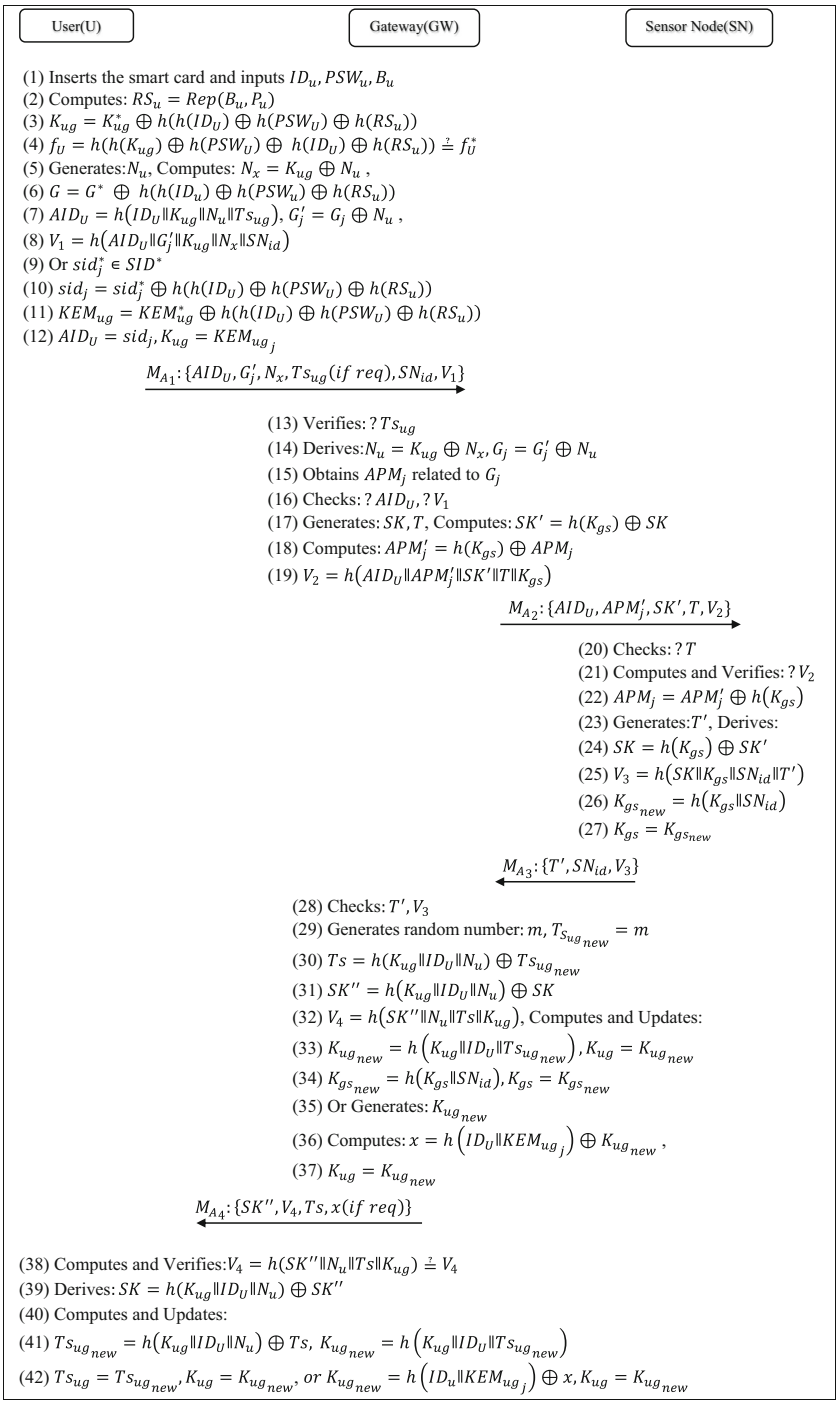

Fig. 5 Authentication and key agreement phase of our scheme

applies the fuzzy extractor function $Gen(.)$ to generate secret data key RS_u and a random auxiliary parameter P_u as $Gen(B_u) = (RS_u, P_u)$. U then computes: K^*_{ug}, KEM^*_{ug}, SID^*, G^* and f^*_u as represented in lines 28 to 32. U stores fuzzy extractor $Gen(.)$, $Rep(.)$, P_u and $h(.)$ in the smart card. Further, U replaces K_{ug} with K^*_{ug}, KEM_{ug} with KEM^*_{ug}, SID with SID^*, G with G^* and f_u with f^*_u. Finally, SC of U contains the information shown in line 33.

6.2 Anonymous authentication and key exchange phase

In this phase, U and SN mutually authenticate each other and finally, both U and SN establish a common session key between them, as is shown in Fig. 5.

Step 1. U first inserts his smart card SC to a terminal and then enters his identity ID_u, password PSW_u and imprints the biometric information B_u. SC computes $RS_u = Rep(B_u, P_u)$ using B_u, and the parameter P_u stored in its memory. SC further computes K_{ug}, f_u as depicted in lines 3 to 4 of Fig. 5, and then checks if $f^*_u = f_u$ holds or not. If it does not hold, this ensures that U does not pass verifications, otherwise, SC generates a random number N_u and computes N_x, G as represented in lines 5 to 6, then user U choose a group-ID G_j from G and encode it as $G'_j = G_j \oplus N_u$ (line 7). Then U computes AID_u, and V_1 (lines 7-8). In case of loss of synchronization between U and GW, the user needs to choose one of the unused pair of $(sid^*_j, KEM^*_{ug_j})$ from (SID^*, KEM^*_{ug}) and then submits his ID_u, PSW_u, B_u and computes sid_j and KEM_{ug_j} (lines 10-11). Then user U assigns the sid_j as AID_u and KEM_{ug_j} as K_{ug}. Finally, the user forms a request message $M_{A_1} : U \rightarrow GW : \{AID_u, G'_j, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$.

Step 2. After receiving the message in Step 2, the GW checks the validity of Ts_{ug}. If the GW cannot find the Ts_{ug} provided by the user in M_{A_1} in its database, it terminates the connection, otherwise, the GW selects the related tuple to the user via Ts_{ug} value. Then, the GW decodes the user's identity ID_u as $ID_u = ID^\sharp_u \oplus h(ID_G \| w \| Ts_{ug})$, K_{ug} as $K_{ug} = K^\sharp_{ug} \oplus h(ID_G \| ID_u \| w)$, respectively. Therefore, the GW exactly can identify who is the user, then the GW checks the validity of V_1. If so, the GW computes N_u and G_j as $N_u = K_{ug} \oplus N_x$ and $G_j = G'_j \oplus N_u$, respectively (line 14). Then GW checks AID_u, if the verification of AID_u is successful then GW will find the user group's G_j related APM_j and encode it as $APM'_j = h(K_{gs}) \oplus APM_j$ (line 18), otherwise the GW terminates the connection. Then, GW generates a session key SK and encodes it as $SK' = h(K_{gs}) \oplus SK$. Subsequently, GW generates a time stamp T and finally computes V_2 as depicted in line 19 and sends message $M_{A_2} : GW \rightarrow SN : \{AID_u, APM'_j, SK', T, V_2\}$ to SN_{id}.

Step 3. Upon receiving the message M_{A_2}, SN checks the validity of timestamp T and V_2 (lines 20-21). If so, SN decodes APM'_j and SK', respectively as $APM_j = APM'_j \oplus h(K_{gs})$ and $SK = h(K_{gs}) \oplus SK'$ (lines 22-24). Hereafter SN updates its shared secret key as $K_{gs} = K_{gs_{new}}$, where $K_{gs_{new}} = h(K_{gs} \| SN_{id})$ (line 26-27).

Step 4. After receiving message M_{A_3}, the GW checks the validity of V_3 (line 28). If so, the GW generates a 64 bits of random m and updates $Ts_{ug_{new}}$ as $Ts_{ug_{new}} = m$ (line 29). Hereafter, GW encodes $Ts_{ug_{new}}$ and SK, respectively as represented in lines 30 to 31. Finally, the GW forms message $M_{A_4} : GW \rightarrow U : \{SK'', V_4, T_s, x(ifreq)\}$. The continue of this step (lines 38-42) is the same as step 4 of section 4.2 (lines 37-43).

Step 5. Both the user U and the sensor node SN will communicate securely using the session key SK. SN responds to the query of the user U depending upon the access privilege mask APM_j stored for user U using the session key SK. Finally, at the end of this phase, SN deletes APM_j from its memory for security reasons.

6.3 Password and biometric update phase

For security reasons, a user U should be allowed to update his/her password as well as personal biometrics without any help of the GW. When the user wants to update the old password PSW_u and old biometric B_u, the following steps are executed:

Step 1. U first inserts his smart card into the terminal. U then inserts his identity ID_u, old password PSW_u, old biometric B_u, the new password PSW^*_u and the new biometric B^*_u.

Step 2. Smart card computes $RS_u = Rep(B_u, P_u)$, and then will retrieve K_{ug}, KEM_{ug}, SID, G and f_u as follows.

- $K_{ug} = K^*_{ug} \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$;
- $KEM_{ug} = KEM^*_{ug} \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$;
- $SID = SID^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$;
- $G = G^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$;
- $f_u = f^*_u \oplus h(h(K_{ug}) \oplus h(PSW_u) \oplus h(ID_u) \oplus h(RS_u))$.

Step 3. Smart card computes $Gen(B^*_u), K^{**}_{ug}$, $SID^{**}, KEM^{**}_{ug}, G^{**}$ and f^{**}_u as bellow.

- $Gen(B^*_u) = (RS^*_u, P^*_u)$;
- $K^{**}_{ug} = K_{ug} \oplus h(h(ID_u) \oplus h(PSW^*_u) \oplus h(RS^*_u))$;
- $SID^{**} = SID \oplus h(h(ID_u) \oplus h(PSW^*_u) \oplus h(RS^*_u))$;

- $KEM_{ug}^{**} = KEM_{ug} \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$;
- $G^{**} = G \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$;
- $f_u^{**} = h(h(K_{ug}) \oplus h(PSW_u^*) \oplus h(ID_u) \oplus h(RS_u^*))$.

Step 4. Finally, smart card will replace K_{ug}^* with K_{ug}^{**}, SID^* with SID^{**}, KEM_{ug}^* with KEM_{ug}^{**}, G^* with G^{**}, f_u^* with f_u^{**}, and P_u with P_u^*.

6.4 Dynamic node addition phase

In this phase, a fresh sensor node will be deployed to the target field in order to continue the services in WSN. This phase is similar to Gope et al.'s scheme.

7 Security analysis of the proposed scheme

In this section, through both informal and formal security analysis, we show that our scheme has the ability to withstand relevant attacks.

In the informal security analysis, we argue the security and soundness of the proposed protocol against the known attacks, beside the point the informal security consists of trial and error methods to find security flaws in the protocol. However, the formal method analyzes the cryptographic protocols based on mathematics and logic. There are a number of logic tools such as BAN-logic [6], GNY-logic [21], Proverif tool [5] and AVISPA tool [4]. In this paper, we use BAN-logic to prove the security correctness of our scheme.

7.1 Informal security analysis

The following subsections show that our scheme has the ability to withstand relevant known attacks.

7.1.1 Stolen smart card attack

In our proposed scheme, if the smart card is lost or stolen, the attacker can easily extract all the sensitive information K_{ug}^*, KEM_{ug}^*, SID^*, G^* and f_u^* from the lost/stolen smart card, since the smart card in our scheme is not tamper-resistant. Note that ID_u, PSW_u and RS_u are unknown to the attacker, and without knowing these parameters, the attacker cannot compute $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$ and $f_u = h(h(K_{ug}) \oplus h(PSW_u) \oplus h(ID_u) \oplus h(RS_u))$, since it is computationally infeasible to derive the ID_u, PSW_u and the biometric secret data key RS_u of the user U due to collision-resistant property of the one-way hash function. Thus, our scheme has the capability to prevent the stolen smart card attack.

7.1.2 Stolen-verifier attack

In stolen-verifier attack, an attacker steals or modifies the verification data (e.g., plaintext passwords, hashed passwords, biometric data, hashed biometric data) stored in the GW. In our protocol, the GW maintains a table consists of $< ID_u^\sharp, K_{ug}^\sharp, K_{gs}^\sharp, G^\sharp, APM^\sharp, (SID, KEM_{ug}^\sharp), Ts_{ug} >$, which contains no information related to the password, so an adversary cannot steal or modify any information from GW. Therefore, our scheme is secure against stolen-verifier attack.

7.1.3 Privileged insider attack

During the user registration in our protocol, U only submits ID_u as a registration message and all the public messages are independent of U's password. Therefore, the insider person has no way to get U's password. Thus, our scheme protects the privileged insider attack.

7.1.4 Identity guessing attack

It is noted that the real identity ID_u of a user U is only known to that user and the GW, which stores encode of ID_u in its database. However, ID_u is never transmitted over the public channel for authentication purpose. Instead of that, the one-time alias identity AID_u or the random shadow-ID sid_j is used in the public communications. The attacker has no way to obtain any useful information related to ID_u for verifying his guessing. Therefore, our scheme is secure against identity guessing attack.

7.1.5 Three-factor security

In the three-factor security model, the main goals of an adversary A are to impersonate a legal user even if he/she has any two factors of the triple (SC, PSW_u, B_u) [41]. We just need to show that A cannot generate a legal request message $M_{A_1} : U \rightarrow GW : \{AID_u, G'_j, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$. Since $AID_u = h(ID_u \| K_{ug} \| N_u \| Ts_{ug})$ or $AID_u = sid_j$, then we just need to show A cannot compute K_{ug} or sid_j without three factors.

Case 1. A **has user's password and smart card:** Upon getting the smart card, A could extract the secret value K_{ug}^*, KEM_{ug}^*, SID^*, G^* and f_u^* stored in the smart card, where $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, and $SID = SID^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. If A wants to impersonate the user, he has to compute K_{ug} from K_{ug}^* or sid_j from SID^*. However, A cannot recover RS_u from P_u since he does not have biometrics of the user. Then, A has no ability to generate AID_u.

Case 2. A **has user's biometrics and a smart card:** A could extract the secret value K_{ug}^*, KEM_{ug}^*, SID^*, G^* and f_u^* stored in the smart card, where $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, and $SID = SID^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. A could recover RS_u from P_u since he has the user's biometric. A may guess password PSW_u, ID_u and computes $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. However, A cannot verify if PSW_u and ID_u is correct. Then, A has no ability to generate correct AID_u.

Case 3. A **has user's password and biometrics:** For the same reason, A could recover RS_u from P_u since he has the user's biometric. A may guess ID_u and computes $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$. However, A cannot verify if ID_u is correct. Therefore, A cannot impersonate the user. From the given discussion, we know that the adversary A cannot generate a legal message $M_{A_1} : U \rightarrow GW : \{AID_u, G'_j, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$ with only two factors. Therefore, our proposed scheme could provide three-factor scheme.

7.1.6 Password and biometric change attack

In order to change the password and biometric of a user U, an attacker needs to pass three-factor (ID_u, PSW_u, B_u). Without these information, it is computationally infeasible to compute $K_{ug} = K_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, $SID = SID^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, $KEM_{ug} = KEM_{ug}^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$, $G = G^* \oplus h(h(ID_u) \oplus h(PSW_u) \oplus h(RS_u))$ and $f_u = f_u^* \oplus h(h(K_{ug}) \oplus h(PSW_u) \oplus h(ID_u) \oplus h(RS_u))$ where $RS_u = Rep(B_u, P_u)$. Because an attacker does not know the K_{ug}, KEM_{ug} and SID, so he/she cannot compute $K_{ug}^{**} = K_{ug} \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$, $SID^{**} = SID \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$, $KEM_{ug}^{**} = KEM_{ug} \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$, $G^{**} = G \oplus h(h(ID_u) \oplus h(PSW_u^*) \oplus h(RS_u^*))$, and $f_u^{**} = h(h(K_{ug}) \oplus h(PSW_u^*) \oplus h(ID_u) \oplus h(RS_u^*))$, where $(RS_u^*, P_u^*) = Gen(B_u^*)$. Our scheme is thus secure against the password and biometric change attack.

7.1.7 Resistant to forgery attack

We consider the following cases:

1. **User impersonation attack:** An adversary A may eavesdrop the message $M_{A_1} : U \rightarrow GW : \{AID_u, G'_j, N_x, Ts_{ug}(ifreq), SN_{id}, V_1\}$ of the previous sessions. Then A can forge a forged message and send it to the GW. After receiving the forged message, the GW can verify the legitimacy of the user U by verifying the $AID_u = h(ID_u \| K_{ug} \| N_u \| Ts_{ug})$. A has to posses ID_u and K_{ug} to forge AID_u. However, without the knowledge of the correct ID, password, biometric key and the possession of smart card, A cannot compute a valid AID_u. Therefore, our scheme could resist user impersonation attack.
2. **GW impersonation attack:** To forge the message $M_{A_2} : GW \rightarrow SN : \{AID_u, APM'_j, SK', T, V_2\}$ in Step 2, the adversary A needs to compute $V_2 = h(AID_u \| SK' \| T \| K_{gs})$. However, A cannot compute V_2 without knowing K_{gs} which is the shared key between user U and sensor node SN. Hence, the adversary A cannot forge the message. Thus, our scheme resist GW forgery attack.
3. **Sensor impersonation attack:** For the same reason, to forge the message $M_{A_3} : SN \rightarrow GW : \{T', SN_{id}, V_3\}$ in Step 3, the adversary A needs to compute $V_3 = h(SK \| K_{gs} \| SN_{id} \| T')$. However, A cannot compute V_3 without the knowledge of the correct K_{gs}. Therefore, our scheme is secure against SN forgery attack.

7.1.8 Session key security

Suppose an attacker eavesdrops the messages $M_{A_2} : GW \rightarrow SN : \{AID_u, APM'_j, SK', T, V_2\}$ and $M_{A_4} : GW \rightarrow U : \{SK'', V_4, T_s, x(ifreq)\}$ during the authentication and key agreement phase. The secret session key $SK' = h(K_{gs}) \oplus SK$ and $SK'' = SK \oplus h(K_{ug} \| ID_u \| N_u)$ is protected by the one-way hash function $h(.)$. In order to compute SK, an adversary needs to know K_{gs} to reveal SK as $SK = SK' \oplus h(K_{gs})$ or he/she needs to know K_{ug}, ID_U and N_u to derive SK as $SK = SK'' \oplus h(K_{ug} \| ID_u \| N_u)$. Due to the collision-resistant one-way property of $h(.)$, it is a computationally infeasible problem for the attacker to derive SK. Thus, our scheme provides the session key security.

7.1.9 Protection of user anonymity

In our scheme a user U's real identity ID_u is never transmitted over an insecure channel. Instead of that, the one-time alias identity $AID_u = h(ID_u \| K_{ug} \| N_u \| Ts_{ug})$ with random transaction sequence number T_s or the random shadow-ID sid_j with emergency key pair is used in the public communications. Due to the collision-resistant one-way property of $h(.)$, it is a computationally infeasible problem for the attacker to derive ID_u from AID_u. Therefor, our scheme achieves user anonymity.

7.1.10 Protection of user untraceability

User untraceability means that the adversary can neither figure out who the user is nor tell apart whether two

sessions are originated by the same(unknown) user. In our scheme, there is not relationship between the one-time alias identity AID_u, T_s and shadow-ID sid_j. Besides, it can also be noticed that during the execution of our authentication protocol all the parameters in the message M_{A_1} are random. Therefor, the attacker is unable to tell whether two protocol runs have the same user involved, and our scheme achieves user untraceability.

7.1.11 Perfect forward secrecy

Perfect forward secrecy means that previously established session key remains secure even when the long-term keys of the server and the user are disclosed [16]. In our scheme, U and SN can compute the session key SK as $SK = SK'' \oplus h(K_{ug} \| ID_u \| N_u)$ and $SK = SK' \oplus h(K_{gs})$, respectively, which will be used to secure the data communication between U and SN. The session key SK is a random number which is different for each session and is unknown to other parties except U, SN and GW. That is, the session key SK is generated independently for each login session. Hence, even if some session keys are revealed, the previous session keys are still secure.

7.1.12 Replay attack

An attacker could replay the eavesdropped messages, such as U's request M_{A_1} or M_{A_2}, M_{A_3} and M_{A_4}. However, the Ts_{ug} value in M_{A_1} and the V_4 value in M_{A_4} prevents any replay attempt from any attacker. The valid period of messages M_{A_2} and M_{A_3} is limited by the timestamp T and T', respectively, so a replay attack can be easily detected by checking the freshness of the timestamp. Therefor, replay attack cannot succeed in our scheme.

7.1.13 Resilience against node capture attack

The resilience against node capture attack of a user authentication scheme in WSN is measured by estimating the fraction of total secure communications that are compromised by a capture of c nodes not including the communication in which the compromised nodes are directly involved [13]. Since the sensor nodes are not equipped with tamper-resistant hardware, the adversary can easily compromise all the secret information including the captured sensor node's secret key K_{gs_i} and session key SK_{ij} established between the user U_j and node S_{n_i}. The session key is generated by the GW using the random number, and thus each established session key between a user and a sensor node is distinct throughout the network. Hence, the compromise of a sensor node does not reveal any other information about other sensor nodes and users in order to compromise any other secure communication between the users and the non-compromised nodes in the network, since the adversary has the ability to compromise the secret key of that captured sensor node only. Therefore, other non-captured sensor nodes have the ability to communicate with 100% secrecy with the actual real-time data to the legitimate users. Hence, our scheme is unconditionally secure against node capture attack.

7.2 Formal security analysis of our proposed scheme

We conduct security analysis of our proposed scheme using BAN-logic [6]. The BAN-logic notations used in the proof are shown in Table 2. The rules that we use in our analysis are as follows:

R1 (Shared key rule) $\frac{P|\equiv P \overset{k}{\longleftrightarrow} Q, P \triangleleft [X]_k}{P|\equiv Q|\sim X}$, if P believes that he/she shared the key K with Q, and P received the message $[X]_k$, then P believes that Q has sent X.

R2 (Belief rule) $\frac{P|\equiv Q|\sim (X,Y)}{P|\equiv Q|\sim X}$, if P believes Q sends the message set (X, Y), then P believes Q sends the message X.

Our formal security analysis involves the following steps:

Step 1. Messages of the protocol

PM1: $AID_u, G'_j, N_x, Ts_{ug}, SN_{id}, V_1$
PM2: $AID_u, APM'_j, SK', T, V_2$
PM3: T', SN_{id}, V_3
PM4: SK'', V_4, Ts, x

Step 2. Idealizing the messages of the protocol This step converts the messages of the protocol to the idealized form of the messages according to the BAN-logic notations.

IM1 $(U \rightarrow GW)$**:** $GW \triangleleft (AID_U, G'_j, N_x, SN_{id})_{K_{ug}}$
IM2 $(U \rightarrow GW)$**:** $GW \triangleleft (ID_U, N_U, Ts_{ug})_{K_{ug}}$

Table 2 BAN-logic notations

Notation	Description
$P \mid\equiv X$	P believes X
$P \triangleleft X$	P receives X
$P \mid\sim X$	P sends X
$P \overset{k}{\rightleftharpoons} X$	The formula K is a secret known only to P and X and only P and X may use K to prove their identities to one another
$\sharp(X)$	X is fresh
$\{X\}_k$	X is encrypted by the secret k
$(X)_k$	X is hashed by the secret k
$P \overset{k}{\leftrightarrow} Q$	P and Q have a shared secret k
$\frac{P}{Q}$	If P then Q

IM3 $(U \rightarrow GW)$**:** $GW \triangleleft \{N_U\}_{K_{ug}}$
IM4 $(U \rightarrow GW)$**:** $GW \triangleleft \{G_j\}_{N_U}$
IM5 $(GW \rightarrow SN)$**:** $SN \triangleleft (AID_U, APM'_j, SK', T)_{K_{gs}}$
IM6 $(GW \rightarrow SN)$**:** $SN \triangleleft \{APM_j\}_{K_{gs}}$
IM7 $(GW \rightarrow SN)$**:** $SN \triangleleft \{SK\}_{K_{gs}}$
IM8 $(SN \rightarrow GW)$**:** $GW \triangleleft (SK, SN_{id}, T')_{K_{gs}}$
IM9 $(GW \rightarrow U)$**:** $U \triangleleft (SK'', N_u, Ts)_{K_{ug}}$
IM10 $(GW \rightarrow U)$**:** $U \triangleleft \{SK\}_{K_{ug}, ID_U, N_u}$

Step 3. Explicit assumptions Initial explicit assumptions of the protocol are given below:

A1: $U \mid\equiv \sharp(N_u)$
A2: $GW \mid\equiv \sharp(SK, T, m)$
A3: $SN \mid\equiv \sharp(T')$
A4: $U \mid\equiv U \overset{K_{ug}}{\longleftrightarrow} GW$
A5: $GW \mid\equiv GW \overset{K_{ug}}{\longleftrightarrow} U$
A6: $GW \mid\equiv GW \overset{K_{gs}}{\longleftrightarrow} SN$
A7: $SN \mid\equiv SN \overset{K_{gs}}{\longleftrightarrow} GW$
A8: $U \mid\equiv U \overset{ID_U}{\rightleftharpoons} GW$

Step 4. Security goals of the protocol According to analytic procedures of BAN logic and the requirement of authentication protocol for WSNs, our protocol should satisfy the following goals:

G1: $GW \mid\equiv U \mid\sim N_x$
G2: $GW \mid\equiv U \mid\sim AID_u$
G3: $GW \mid\equiv U \mid\sim N_u$
G4: $GW \mid\equiv U \mid\sim G_j$
G5: $SN \mid\equiv GW \mid\sim T$
G6: $SN \mid\equiv GW \mid\sim AID_u$
G7: $SN \mid\equiv GW \mid\sim APM_j$
G8: $SN \mid\equiv GW \mid\sim SK$
G9: $GW \mid\equiv SN \mid\sim T'$
G10: $GW \mid\equiv SN \mid\sim SN_{id}$
G11: $U \mid\equiv GW \mid\sim Ts$
G12: $U \mid\equiv GW \mid\sim SK$

Step 5. Deducing the security goals of the protocol In this step, by applying logical rules to the idealized messages and the initial premises mentioned in the previous steps, we analyze the security of the protocol as follows:

According to IM1, A5, and R1:

Result1: $GW \mid\equiv U \mid\sim (AID_U, G'_j, N_x, SN_{id})$

By Result1 and R2, we have:

Result2: $GW \mid\equiv U \mid\sim N_x$ (satisfy **G1**);
Result3: $GW \mid\equiv U \mid\sim AID_U$ (satisfy **G2**);

According to IM3, A5, and R1:

Result4: $GW \mid\equiv U \mid\sim N_u$ (satisfy **G3**);

By Result4 and IM4, we have:

Result5: $GW \mid\equiv U \mid\sim G_j$ (satisfy **G4**);

According to IM5, A7, and R1:

Result6: $SN \mid\equiv GW \mid\sim (AID_U, APM'_j, SK', T)$

By Result6 and R2, we have:

Result7: $SN \mid\equiv GW \mid\sim T$ (satisfy **G5**);
Result8: $SN \mid\equiv GW \mid\sim AID_U$ (satisfy **G6**);

According to IM6, A7, and R1:

Result9: $SN \mid\equiv GW \mid\sim APM_j$ (satisfy **G7**);

According to IM7, A7, and R1:

Result10: $SN \mid\equiv GW \mid\sim SK$ (satisfy **G8**);

According to IM8, A6, and R1:

Result11: $GW \mid\equiv SN \mid\sim (SK, SN_{id}, T')$

By Result11 and R2, we have:

Result12: $GW \mid\equiv SN \mid\sim T'$ (satisfy **G9**);
Result13: $GW \mid\equiv SN \mid\sim SN_{id}$ (satisfy **G10**);

According to IM9, A4, and R1:

Result14: $U \mid\equiv GW \mid\sim (SK'', N_u, Ts)$

By Result6 and R2, we have:

Result15: $U \mid\equiv GW \mid\sim Ts$ (satisfy **G11**);

According to IM10, A1, A4, A8 and R1:

Result16: $U \mid\equiv GW \mid\sim SK$ (satisfy **G12**);

It is clear that we can satisfy all the goals respectively. So, our proposal is secure.

8 Performance comparison

In this section, we compare the features, computational overhead and communication overhead of our proposed scheme with other related schemes, such as Yeh et al.'s scheme [49], Xue et al.'s scheme [48], Das et al.'s scheme [10], Jiang et al.'s scheme [29], Das et al.'s scheme [11] and Gope et al.'s scheme [24].

8.1 Feature comparison

In Table 3, we have presented a number of security attacks as well as functionality requirement to compare the proposed protocol with other similar protocols. In Table 3, the symbol 'yes' represents that the scheme prevents attack

Table 3 Feature comparison

Security features	[49]	[48]	[10]	[29]	[11]	[24]	Ours
User anonymity without exhaustive search	No	No	No	Yes	Yes	Yes	Yes
User untraceability	No	No	Yes	Yes	Yes	No	Yes
Resilient against replay attack	No	Yes	Yes	Yes	Yes	Yes	Yes
Privacy against eavesdrops	No	No	Yes	No	Yes	Yes	Yes
Support of dynamic node addition	No	No	Yes	No	Yes	Yes	Yes
Robustness against insider attack	Yes	No	Yes	No	Yes	Yes	Yes
Robustness against lost smart card	No	No	Yes	No	Yes	Yes	Yes
Perfect forward secrecy	No	No	No	No	No	Yes	Yes
Resilient to DoS attack	No	No	No	No	No	Yes	Yes
Resilient to forgery attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resilient against node capture attack	No	Yes	Yes	Yes	Yes	Yes	Yes
Resilient against stolen-verifier attack	No	Yes	Yes	Yes	Yes	Yes	Yes
Resilient against identity guessing attack	No	No	Yes	Yes	Yes	Yes	Yes
Support of three-factor security	No	No	Yes	No	Yes	No	Yes
Supports correct password and biometric update	No	No	Yes	No	Yes	No	Yes
Resilient against session key disclosure attack	Yes	Yes	Yes	Yes	Yes	No	Yes
Access control	No	No	No	No	No	No	Yes

or satisfies the attribute, and the symbol 'No' represents that the scheme does not resist attack or failed to satisfy that attribute. We have found in Table 3 that the protocols in [10, 11, 29, 48, 49] suffers from the DoS attack, and also cannot ensure the perfect forward secrecy property, which is highly imperative for any session-key based authentication protocol [23]. Protocols in [24, 29, 48, 49] does not support three-factor authentication. From this table, it is clear that our scheme provides better security features and higher security level. Thus, our scheme is superior in terms of features as compared to those for other schemes.

8.2 Overall Computational overhead comparison

In WSNs, energy is a constraint and thus, the user authentication protocol must be lightweight in terms of computation. Our protocol used the hash function, and the fuzzy extractor function $(Gen(.), Rep(.))$, which are efficient. According to experiments results executed in [24], each modular exponential operation in ECC-160 algorithm takes 1.2 Ws energy and 11.69 ms execution time. The approximate running time and energy consumption of symmetric key encryption/decryption (128 bit AES-CBC) and hash function (SHA-256) are $t_{sym} = 4.62$ ms with .72 Ws and $t_{Hash} = 1.06$ ms with 0.27 Ws, respectively. This experiments is performed using the modular sensor board MSB-430 with the TI MSP430 micro controller [24]. Also, as in [26], we assume that the time t_f for executing a fuzzy extractor is about 17.1 ms. In Table 3, we have summarized the computational cost of the proposed protocol and existing protocols in [10, 11, 24, 29, 48, 49], for user, GW and sensor node, where our proposed scheme requires only $20 * t_{Hash} + t_f$ operations. Although our proposed scheme requires a few more operations in the authentication phase than [24, 29, 48] schemes, the extra operations are justifiable considering that our protocol remedies their security vulnerabilities and also provides more security features that are absent in [24, 29, 48]

Table 4 Overall Computational overhead comparison

Scheme	User	GW	Sensor node	Total cost	Rough estimation
Yeh et al. [49]	$2t_{Exp} + t_{Hash}$	$4t_{Exp} + 4t_{Hash}$	$2t_{Exp} + 3t_{Hash}$	$8t_{Hash} + 8t_{Exp}$	100 ms
Xue et al. [48]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$	$22t_{Hash}$	23 ms
Das [10]	$7t_{Hash} + t_f$	$t_{Sym} + 2t_{Hash}$	$t_{Sym} + 2t_{Hash}$	$11t_{Hash} + 2t_{Sym} + t_f$	38 ms
Jiang et al. [29]	$7t_{Hash}$	$10t_{Hash}$	$5t_{Hash}$	$22t_{Hash}$	23 ms
Das [11]	$9t_{Hash} + t_f$	$11t_{Hash}$	$5t_{Hash}$	$25t_{Hash} + t_f$	43 ms
Gope et al. [24]	$7t_{Hash}$	$9t_{Hash}$	$3t_{Hash}$	$19t_{Hash}$	20 ms
Ours	$8t_{Hash} + t_f$	$9t_{Hash}$	$3t_{Hash}$	$20t_{Hash} + t_f$	38 ms

Table 5 Computational and communicational cost of the sensor node

Scheme	Computational cost	Execution time	Communication cost
Yeh et al. [49]	3.21 Ws	26.56 ms	51 byte
Xue et al. [48]	1.35 Ws	5.3 ms	51 byte
Das [10]	1.53 Ws	7.8 ms	35 byte
Jiang et al. [29]	1.35 Ws	5.3 ms	51 byte
Das [11]	1.62 Ws	6.36 ms	51 byte
Gope et al. [24]	0.81 Ws	3.18 ms	35 byte
Ours	0.81 Ws	3.18 ms	35 byte

schemes like three-factor user authentication, which is essential for the successful deployment of a secure WSN. As shown in Table 4, it can be stated that, the computational overhead of the proposed scheme is significantly less from previous three-factor user authentication scheme [11].

8.3 Computational and communicational cost of the sensor node

As WSNs is energy constraint environment and the battery power of sensor nodes are very low than the gateway node, the computation and communication costs of the sensor nodes should be as minimum as possible for achieving better efficiency. In Table 3, we have compared the computational overhead sensor node of the proposed protocol and existing protocols [10, 11, 29, 48, 49] for the sensor node. Table 5 shows that, for Yeh et al.'s scheme [49], Xue et al.'s scheme [48], Das et al.'s scheme [10], Jiang et al.'s scheme [29], Das et al.'s scheme [11] and Gope et al.'s scheme [24] and our scheme, a sensor node SN requires the computational cost during the authentication phase as $2t_{Exp} + 3t_{Hash}(26.56\ ms)$, $5t_{Hash}(5.3\ ms)$, $t_{Sym} + 3t_{Hash}(7.8\ ms)$, $5t_{Hash}(5.3\ ms)$, $6t_{Hash}(6.36\ ms)$, $3t_{Hash}(3.18\ ms)$ and $3t_{Hash}(3.18\ ms)$, respectively. Thus our scheme is much less than the other schemes. Finally, in Table 5, we have compared the communicational cost of our scheme with other schemes for the authentication phase. We assume that, the length of the identity of the sensor node is $128 - bit$, and the length of the timestamp value is $24 - bit$. Then, the length of the message sends by each sensor node in our proposed scheme and [10] is $35 - byte$, which is significantly smaller than [11, 29, 48, 49] schemes. Therefore, our scheme is much suited for the resource-constrained sensor node due to computational and communicational cost.

9 Conclusion

In this paper, we have analyzed a recent user authentication scheme proposed by Gope et al. for wireless sensor networks, and showed how their scheme is vulnerable to session key disclosure attack. In order to withstand the vulnerability found in Gope et al.'s scheme, we have proposed an improved scheme. We also have enhanced Gope et al.'s scheme security features using three-factor user authentication and access control while retaining the original merits of Gope et al.'s scheme. We have demonstrated that our scheme provides more security features and high security level, which is evident through both informal and formal security analysis. The performance analysis showed that our protocol is efficient than existing protocols from the aspects of energy consumption of the sensor node, communication cost and running time, which make our scheme very suited for resource constrained sensor nodes. Overall, higher security along with low communication and computation costs make our scheme appropriate for WSN applications.

References

1. Amin R, Islam SH, Biswas G, Khan MK, Leng L, Kumar N (2016) Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput Netw 101:42–62
2. Anastasi G, Conti M, Di Francesco M, Passarella A (2009) Energy conservation in wireless sensor networks: a survey. Ad Hoc Netw 7(3):537–568
3. Arasteh S, Aghili SF, Mala H (2016) A new lightweight authentication and key agreement protocol for internet of things. In: 2016 13th international iranian society of cryptology conference on information security and cryptology (ISCISC). IEEE, pp 52–59
4. Armando A, Basin D, Boichut Y, Chevalier Y, Compagna L, Cuéllar J, Drielsma PH, Héam PC, Kouchnarenko O, Mantovani J et al (2005) The AVISPA tool for the automated validation of internet security protocols and applications. In: International conference on computer aided verification. Springer, pp 281–285
5. Blanchet B (2014) Automatic verification of security protocols in the symbolic model: the verifier proverif. In: Foundations of security analysis and design VII. Springer, pp 54–87
6. Burrows M, Abadi M, Needham R (1990) A logic of authentication. ACM Transactions on Computer Systems (TOCS) 8(1):18–36
7. Chang CC, Le HD (2016) A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wirel Commun 15(1):357–366

8. Chen TH, Shih WK (2010) A robust mutual authentication protocol for wireless sensor networks. ETRI journal 32(5):704–712
9. Das AK (2015) A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. Int J Commun Syst 30(1):1–25
10. Das AK (2015) A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. Wirel Pers Commun 82(3):1377–1404
11. Das AK (2016) A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. Peer-to-peer Networking and Applications 9(1):223–244
12. Das AK, Chatterjee S, Sing JK (2015) A new biometric-based remote user authentication scheme in hierarchical wireless body area sensor networks. Adhoc & Sensor Wireless Networks 28:21–256
13. Das AK, Sharma P, Chatterjee S, Sing JK (2012) A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. J Netw Comput Appl 35(5):1646–1656
14. Das ML (2009) Two-factor user authentication in wireless sensor networks. IEEE Trans Wirel Commun 8(3):1086–1090
15. Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
16. Diffie W, Oorschot PC, Wiener MJ (1992) Authentication and authenticated key exchanges. Des Codes Crypt 2(2):107–125
17. Dolev D, Yao A (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208
18. Ekici E, Gu Y, Bozdag D (2006) Mobility-based communication in wireless sensor networks. IEEE Commun Mag 44(7):56
19. Fan R, He DJ, Pan XZ et al (2011) An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks. Journal of Zhejiang University SCIENCE C 12(7):550–560
20. Fan R, Ping LD, Fu JQ, Pan XZ (2010) A secure and efficient user authentication protocol for two-tiered wireless sensor networks. In: 2010 second pacific-asia conference on circuits, communications and system (PACCS), vol 1. IEEE, pp 425–428
21. Gong L, Needham R, Yahalom R (1990) Reasoning about belief in cryptographic protocols. In: 1990 IEEE computer society symposium on research in security and privacy, 1990. Proceedings. IEEE, pp 234–248
22. Gope P, Hwang T (2015) A realistic lightweight authentication protocol preserving strong anonymity for securing rfid system. Comput Secur 55:271–280
23. Gope P, Hwang T (2016) Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks. IEEE Syst J 10(4):1370–1379
24. Gope P, Hwang T (2016) A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. IEEE Trans Ind Electron 63(11):7124–7132
25. He D, Gao Y, Chan S, Chen C, Bu J (2010) An enhanced two-factor user authentication scheme in wireless sensor networks. Ad hoc & Sensor Wireless Networks 10(4):361–371
26. He D, Kumar N, Lee JH, Sherratt R (2014) Enhanced three-factor security protocol for consumer usb mass storage devices. IEEE Trans Consum Electron 60(1):30–37
27. Huang HF, Chang YF, Liu CH (2010) Enhancement of two-factor user authentication in wireless sensor networks. In: 2010 sixth international conference on intelligent information hiding and multimedia signal processing (IIH-MSP). IEEE, pp 27–30
28. Huang X, Xiang Y, Chonka A, Zhou J, Deng RH (2011) A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Trans Parallel Distrib Syst 22(8):1390–1397
29. Jiang Q, Ma J, Lu X, Tian Y (2015) An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Networking and Applications 8(6):1070–1081
30. Karl H, Willig A (2007) Protocols and architectures for wireless sensor networks. Wiley, New York
31. Khan MK, Alghathbar K (2010) Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors 10(3):2450–2459
32. Kumar P, Choudhury AJ, Sain M, Lee SG, Lee HJ (2011) Ruasn: a robust user authentication framework for wireless sensor networks. Sensors 11(5):5020–5046
33. Li CT, Hwang MS (2010) An efficient biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 33(1):1–5
34. Li X, Niu JW, Ma J, Wang WD, Liu CL (2011) Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 34(1):73–79
35. Lloyd EL, Xue G (2007) Relay node placement in wireless sensor networks. IEEE Trans Comput 56(1):134–138
36. Nyang D, Lee MK (2009) Improvement of das's two-factor authentication protocol in wireless sensor networks. IACR Cryptology ePrint Archive 2009:631
37. Odelu V, Das AK, Goswami A (2014) A secure effective key management scheme for dynamic access control in a large leaf class hierarchy. Inform Sci 269:270–285
38. Qi J, Zhuo M, Jianfeng M, Guangsong L (2012) Security enhancement of robust user authentication framework for wireless sensor networks. China Communications 9(10):103–111
39. Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
40. Sun DZ, Li JX, Feng ZY, Cao ZF, Xu GQ (2013) On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. Pers Ubiquit Comput 17(5):895–905
41. Tan Z (2014) A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. J Med Syst 38(3):16
42. Vaidya B, Makrakis D, Mouftah H (2016) Two-factor mutual authentication with key agreement in wireless sensor networks. Security and Communication Networks 9(2):171–183
43. Vaidya B, Makrakis D, Mouftah HT (2010) Improved two-factor user authentication in wireless sensor networks. In: 2010 IEEE 6th international conference on wireless and mobile computing, networking and communications (wimob). IEEE, pp 600–606
44. Wang CH, Lin CY (2011) An efficient delegation-based roaming payment protocol against denial of service attacks. In: 2011 international conference on electronics, communications and control (ICECC). IEEE, pp 4136–4140
45. Wang D, Wang P (2014) On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. Comput Netw 73:41–57
46. Watro R, Kong D, Cuti SF, Gardiner C, Lynn C, Kruus P (2004) Tinypk: securing sensor networks with public key technology. In: Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks. ACM, pp 59–64
47. Wong KH, Zheng Y, Cao J, Wang S (2006) A dynamic user authentication scheme for wireless sensor networks. In: IEEE international conference on sensor networks, ubiquitous, and trustworthy computing, 2006, vol 1. IEEE, p 8
48. Xue K, Ma C, Hong P, Ding R (2013) A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J Netw Comput Appl 36(1):316–323
49. Yeh HL, Chen TH, Liu PC, Kim TH, Wei HW (2011) A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 11(5):4767–4779

50. Yu J, Wang G, Mu Y, Gao W (2014) An efficient generic framework for three-factor authentication with provably secure instantiation. IEEE Trans Inf Forensics Secur 9(12):2302–2313

AmirHosein Adavoudi-Jolfaei received his B.S. degree in Information Technology Engineering from Shahid Rajaee University of Tehran in 2008. Now he is an M.S. student at University of Isfahan since 2015. His main research interests include lightweight cryptography and cryptanalysis, WSNs security, RFID authentication protocols.

Maede Ashouri-Talouki is an Assistant Professor of IT Engineering department of University of Isfahan (UI). She received her B.S., M.S., and Ph.D. degrees from University of Isfahan in 2004, 2007 and 2012, respectively. In 2013, she joined University of Isfahan. Her research interests include mobile networks security, user privacy and anonymity, cryptographic protocols and network security.

Seyed Farhad Aghili received his M.S. degree in Electrical Engineering from Shahid Rajaee Teacher Training University (SRTTU), 2013. He is currently a Ph.D. student at the IT Engineering department of the University of Isfahan. His current research interest includes RFID security.