Wireless Netw (2019) 25:1461–1475
DOI 10.1007/s11276-017-1604-0

Anonymous three-factor authenticated key agreement for wireless sensor networks

Yanrong Lu[1] · Guangquan Xu[1] · Lixiang Li[2] · Yixian Yang[2]

Published online: 3 November 2017

Abstract Secure information exchange in wireless sensor networks (WSN) is a continuing issue since the resource-constrained sensors generally deployed over an unattended environment. To access the real-time data from the sensors, user authentication and key agreement as an important tool for secure communications in WSN plays a vital role. Recently, Das proposed an efficient biometrics based security scheme by only using lightweight symmetric-key primitives. Their scheme is efficient in computation, but we find the scheme of Das is not actually achieve the three-factor security, thus failing to prevent the user impersonation attack. Additionally, the failure of user anonymity also gives an opportunity for the adversary to mount impersonation attacks. With the purpose of mitigating all the problems in Das's scheme, we present an anonymous three-factor key agreement using Elliptic Curve Cryptography. Using the Burrows–Abadi–Needham logic to ensure the mutual authentication properties. Through the rigorous security analysis, we show that the proposed scheme withstands various attacks. In addition, Automated Validation of Internet Security Protocols and Applications (AVIPSA) tool is used to verify its security.

✉ Yanrong Lu
luyanrong1985@163.com

Yixian Yang
yxyang@bupt.edu.cn

[1] Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianjin 300072, China

[2] Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Keywords Anonymous · Three-factor · Cryptanalysis · Wireless sensor networks

1 Introduction

With the advancement of wireless communication and sensor technologies, wireless sensor networks (WSN) [1] are widely used in many fields. Application fields of WSN comprise but are not limited to habitat monitoring [2], health environment monitoring [3, 4], military applications [5], indoor sensor networks [6], industrial and consumer applications [7] and preventing chemical, biological, or nuclear threats in an area [8–10]. In WSN, several dozens to thousands of highly resource-constrained sensor nodes are randomly deployed in a target field. The data collected by these sensor nodes can be given access to the external users. As a result, authentication and key agreement in this environment becomes an essential security mechanism to authenticate those who are authorized to access data when they demand. After an adequate progress in link layer security [11] and network layer security [12], the application layer security in WSN also attracted the attention of many researchers and many user authentication schemes were proposed for WSN. He et al. [13] proposed a lightweight two-factor authenticated key agreement scheme for WSN, where the sensor node is no need of registration for the gateway node. To protect user's identity, they also proposed an access control scheme using ring signature for WSN [14].

With the deepen of the network technology, on the one hand, the user's identity protection as a desirable security property is gradually receiving a lot of attention since the user is most likely not willing to be tracked his identity to the content he requests. This security attribute has led to

recent increasing popularity of privacy-preserving scheme for different applications, such as cloud computing [15, 16], GSM systems [17], wearable health monitoring systems [18], road networks [19–21], global mobility networks[22], location based service systems[23–25], and so on. On the other hand, Das [26] gave a new direction to the application layer in WSN by incorporating two-factor authentication concept namely the smart card and the password. This concept uncovers the potential threats which the server side needs to keep the password table for each user. Two-factor secure means the proposed scheme should be secure in condition that password or smart card is stolen, but not of both. This hot topic combines user anonymity attracting many researchers' attention, several two-factor authentication scheme with user privacy support have been proposed [27]. Wang et al. [28] disclosed the relationships among several security attributes by analyzing the representative two two-factor authentication schemes in 2015. And they proposed an ideal two-factor authentication scheme which satisfied 12 independent criteria under the formal proof[29].

Although the scheme of Das [26] proposed a solution avoiding the stolen verifier attack, a sequence of cryptanalysis about his scheme was performed. Nyang-Lee [30] identified that Das's scheme was vulnerable to the off-line password guessing and node capture attacks. To conquer the security pitfalls, Nyang-Lee [30] proposed an improvement without sacrificing any efficiency and usability based on Das's scheme. Also, Khan-Alghathbar [31] cryptanalyzed that Das's scheme [26] could not withstand the insider and GWN bypassing attacks. As a counter measure to these sufferings, Khan-Alghathbar [31] proposed security patches and improvements. They claimed that their enhanced scheme could withstand the insider attack, GW-node bypassing attack and provide mutual authentication. However, Yuan [32] came across some security problems in Khan-Alghathbar's scheme, like non resistance to lost smart card breach attack, failure of providing non-repudiation and achieving mutual authentication between the user and the GW-node. Yuan [31] then modified Khan-Alghathbar's scheme [31] to rectify its drawbacks and improve its security with increasing computation overhead. Subsequently, a number of authenticated key agreement schemes [33, 34] for WSN were proposed to deal with security vulnerabilities about two-factor anonymous authentication. Sun et al. [35] proposed a two-factor user authentication scheme to defeat the GW-node impersonation attack, the GW-node bypassing attack and the privileged-insider attack. Turkanovi et al. [36] proposed a scheme to enable a remote user to securely negotiate a session key with a general sensor node, using a lightweight key agreement protocol.

With the widespread of basic pattern recognition system, more and more biometrics based authentication schemes have been proposed [37, 38]. For example, Xie et al. [39] proposed an improved ECC-based three-factor authentication protocol for mobile networks which uses user's biometrics to transmit the user's identity and the authentication message in confidential manner. Besides, they used a random nonce to decrypt and encrypt messages without using the server's public key for reducing computation cost and avoiding the key management problem. Das [40] proposed a novel three-factor user authentication scheme suited for distributed WSN which supports efficiently updating password and biometric change phase without contacting the BS and dynamic node addition phase. Biometric methods include fingerprint scanning[41], facial recognition, hand geometry recognition or retinal scans, which can be used as a proof of user's identity. As compared to the traditional passwords keys[42], biometric keys have many advantages as follows [43] :

1. it is difficult to lose or forget biometric keys;
2. it is difficult to copy or share biometric keys;
3. it is difficult to forge or distribute biometrics;
4. it is difficult to guess biometric keys;
5. it is more difficult to break biometric keys.

Generally, the user's biometric should be preprocessed using biohashing [44] and fuzzy extractor [45] due to it is easily effected by surrounding environment. Due to its excellent advantages compared with password, biometrics based authenticated key agreement schemes were successively presented. He-Wang [46] proposed anonymous ECC based authentication and key agreement scheme for multi-server, but at the cost of a slight higher computation overheads. Li et al. [47] also proposed a fingerprint information based authentication scheme for IoT environments, where the fuzzy commitment scheme is utilized to check the legitimacy of fingerprint.

Recently, Das [48] also proposed a three-factor authentication only employing hash function. Undoubtedly, Das's scheme has a good performance compared to the public-key cryptographic technique based authentication. However, we point out the scheme of Das has several security pitfalls, such as no user anonymity, no three-factor security and user impersonation attack. Such security weaknesses makes their scheme cannot be applied in practical environment. Motivated by this, we present an anonymous three-factor key agreement using elliptic curve cryptography to mitigate all the problems of the scheme Das. We use BAN logic to show the proposed scheme achieves the mutual authentication. The formal security analysis and software verification demonstrate the proposed scheme is secure against many attacks. In

addition, we also compare the performance with the related works.

The remainder of paper is planned as follows: Sect. 2 is Preliminaries. Section 3 is regarding the review of Das's scheme. Section 4 is regarding the thorough cryptanalysis of Das's user authentication scheme for WSN. Section 5 presents an improvement based on Das's scheme. Section 6 is regarding the security analysis of our scheme. Section 7 shows the performance comparison of our scheme and previous schemes. Finally, the paper is concluded in Sect. 8.

2 Preliminaries

Given biometric input Bio, a fuzzy extractor could extract a random string μ_i. One important property of the fuzzy extractor is that it could output the same random string when the input changes, but it remains close. To recover μ_i from a new biometric input Bio, a uniformly random auxiliary string v_i will be generated and used in the following operations. The fuzzy extractor is formally defined as follows [45].

Fuzzy extractor A fuzzy extractor is given by two procedures (Gen, Rep).

$Gen(B_i) = (\mu_i, v_i)$: The probabilistic generation procedure Gen on input B_i outputs a random string μ_i and a random auxiliary string v_i;

$Rep(B_i', v_i) = \mu_i$: The deterministic reproduction procedure Rep on input B_i' which is reasonably close to B_i and the corresponding random auxiliary string v_i and finally recover μ_i.

3 Review of Das's scheme

This section reviews Das's user authentication scheme for WSN, which is based on one-way hash functions.

3.1 Pre-deployment

The gateway node GWN chooses a unique identity SID_j for every deployed sensor S_j. Then, GWN selects a master key MK_{S_j} and computes a secret key $K_j = h(SID_j, MK_{S_j})$ for each S_j. Finally, the GWN stores the pair $\{SID_j, K_j\}$ in its database and then deletes MK_{S_j}.

3.2 User registration

UR_1: U_i submits his identity Id_i, password pw_i, and then imprints biometric Bio_i on the specific device.

UR_2: U_i computes $RPW_i = h(Id_i, pw_i, K)$ and sends the registration message $\{Id_i, RPW_i, ek_i\}$ to the GWN via a secure channel, where K is a nonce and ek_i is a symmetric key shared with the GWN.

UR_2: The GWN computes $f_i = h(Id_i, X_s)$ and issues a smart card which contains $\{f_i, h(), Gen(), Rep(), \mathcal{T}\}$ and sends it securely to U_i, where X_s is a 1024-bit nonce.

UR_3: U_i computes $Gen(Bio_i) = (\sigma_i, \tau_i), f_i^* = f_i \oplus h\ (Id_i, \sigma_i, K), e_i = h(Id_i, RPW_i, \sigma_i),\ r_i = h(Id_i, \sigma_i) \oplus K, BE_i = h\ (Id_i, \sigma_i) \oplus ek_i$. U_i then replaces f_i with f_i^* in the smart card before storing the information $\{\tau_i, e_i, r_i, BE_i\}$ in smart card. Thus, the smart card finally containing the information $\{\tau_i, e_i, r_i, BE_i, f_i^*, h(), Gen(), Rep(), \mathcal{T}\}$.

3.3 Login

U_i enters his smart card into a smart card reader and imprints biometric Bio_i. U_i then keys his Id_i and pw_i. The smart card computes $\sigma_i' = Rep(Bio_i, \tau_i),\ K' = r_i \oplus h(Id_i, \sigma_i'), RPW_i' = h(Id_i, pw_i, K_i)$ and $e_i = h(Id_i, RPW_i', \sigma_i')$. The smart card checks the condition $e_i \stackrel{?}{=} e_i'$. If it is true, the smart card sends the login message $\{Id_i, req\}$ to the GWN via a public channel, where req is a request.

3.4 Authentication and key agreement

AK_1: When the login message is received, GWN checks Id_i. If it is valid, the GWN sends the message $\{R\}$ to U_i, where R is a nonce.

AK_2: The smart card computes $ek_i = BE_i \oplus h(Id_i, \sigma_i)$ and sends the message $\{E_{ek_i}(R, T_1, SID_j)\}$ to the GWN, where T_1 is the current timestamp.

AK_3: The GWN decrypts $E_{ek_i}(R, T_1, SID_j)$ to retrieve (R, T_1, SID_j) using the key ek_i stored in its database. The GWN then verifies whether T_1 is within the tolerable time interval. If it is true, the GWN further checks the validness of the decrypted R.

AK_4: The GWN computes $f_i^* = h(Id_i, h(X_s)), f_i^{**} = h(SID_j, f_i^*)$ and $Y_j = E_{K_j}(Id_i, SID_j, T_1, T_2, f_i^{**})$, where T_2 is the current timestamp of GWN side. The GWN then sends the message $\{Id_i, Y_j\}$ to S_j.

AK_5: S_j decrypts Y_j to retrieve $(Id_i, SID_j, T_1, T_2, f_i^{**})$ using its secret key K_j. S_j then checks the validity of the timestamp T_2 and (Id_i, SID_j). If they are valid, S_j computes a session key as $sk_{ij} = h(f_i^{**}, Id_i, SID_j, T_1, T_3)$ and sends the message $\{h(sk_{ij}), T_3\}$ back to U_i, where T_3 is the current timestamp of S_j.

AK_6: U_i checks the validity of the timestamp T_3 by the condition $T_3 - T_3^* < \Delta T$, where T_3^* is U_i's the current timestamp of U_i. U_i computes $f_i' = f_i^* \oplus h(\sigma_i', Id_i, K), f_i'' = h(SID_j, f_i')$ and $sk_{ij}' = h(f_i'', Id_i, SID_j, T_1, T_3)$. U_i then checks whether $h(sk_{ij}) \stackrel{?}{=} h\ (sk_{ij})$. If it holds, U_i confirms the session

key sk_{ij} with S_j and uses it to secure a subsequent communications.

3.5 Password change

P_1: U_i enters his smart card into a smart card reader and imprints biometric Bio_i. U_i then keys his Id_i and pw_i. The smart card computes $\sigma_i' = Rep(Bio_i, \tau_i)$, $K' = r_i \oplus h(Id_i, \sigma_i')$, $RPW_i' = h(Id_i, pw_i, K_i)$ and $e_i = h(Id_i, RPW_i', \sigma_i')$. The smart card checks the condition $e_i \stackrel{?}{=} e_i'$. If it is true, U_i inputs new password pw_i^{New} and imprints new biometrics Bio_i^{New}. The smart card computes $RPW_i^{new} = h(Id_i, pw_i^{New}, K^{New})$, $Gen(Bio_i^{New}) = (\sigma_i^{New}, \tau_i^{New})$, $e_i^{New} = h(Id_i, RPW_i^{New}, \sigma_i^{New})$, $r_i^{New} = h(Id_i, \sigma_i^{New}) \oplus K^{New}, BE_i^{New} = h(Id_i, \sigma_i^{New}) \oplus ek_i, f_i^{New} = h(Id_i \oplus h(X_s)) \oplus h(\sigma_i^{New}, Id_i, K^{New})$. Finally, U_i replaces τ_i, e_i, r_i, BE_i and f_i^* with $\tau_i^{New}, e_i^{New}, r_i^{New}, BE_i^{New}$ and f_i^{New}, respectively.

4 Security analysis of Das's scheme

This section highlight the security risks in Das's scheme if the unauthorized malicious user $\mathbb{E}$ has the ability to intercept, alter, delete, block, or insert any messages exchanged in the channel [49]. The possible attacks are described as follows.

4.1 No provision of user anonymity

Since authentication of a user is done via an unsecure open channel, an ideal scheme should ensure user anonymity thus disabling any private information leakage if $\mathbb{E}$ would eavesdrop the communication. However, Das's scheme fails to do this, which offers convenience for the next subsection attack.

4.2 Three-factor security violation attack along with user impersonation attack

Three-factor security prevents $\mathbb{E}$ who has learned at most two components of the triple (password, smart card, biometric) from mounting a masquerading attack. Unfortunately, in Das's scheme, the smart card and biometric breach will not only lead to the leakage of the shared symmetric-key ek_i but perform an impersonation attack. $\mathbb{E}$ executes the attack with the following operations.

A_1. $\mathbb{E}$ could extract the secrets $\{\tau_i, e_i, r_i, BE_i, f_i^*, h(), Gen(), Rep(), \mathrm{T}\}$ through the differential power attack [50] and recover σ_i from τ_i with the user's biometrics. Owing to the exposure of user's identity in the channel, $\mathbb{E}$ could derive ek_i by $BE_i \oplus h(Id_i, \sigma_i)$ and K by $r_i \oplus h(Id_i, \sigma_i)$.

A_2. $\mathbb{E}$ guesses a password pw_i' and verifies whether $h(Id_i, h(Id_i, pw_i', K), \sigma_i)$ is equal to e_i. If it is true, $\mathbb{E}$ finds the correct password; otherwise, $\mathbb{E}$ repeats steps A_1 and A_2 until the correct password is found.

With the correct password and the data (Id_i, ek_i), thus $\mathbb{E}$ could impersonate the U_i, he performs as follows.

A_3. $\mathbb{E}$ sends $\{Id_i, request\}$ to GWN, GWN sends a random challenge R to U_i to respond the query data;

A_4. $\mathbb{E}$ sends the encrypted (SID_j, T_1, R) with the shared symmetric key ek_i to GWN;

A_5. The message will go through the verification of GWN by checking the validness of T_1. Then, GWN proceeds to next step and sends $\{Id_i, Y_j\}$ to the nearest sensor node;

A_6. S_j now decrypts Y_j and checks the validity of T_2, Id_i and SID_j. If it is true, the session key is computed by $sk_{ij} = h(Id_i, f'', SID_j, T_3, T_1)$ and is sent back by hashing along with T_3 to responds to $\mathbb{E}$'s query;

A_7. After checking the correctness of the session key, $\mathbb{E}$ who is not a legitimate user of the sensor network system, enjoys the resources as an authorized user without being a member of the system.

In this regard, three-factor security violation attack is rather effective and threatens to Das's scheme.

5 Proposed scheme

This section presents an authentication and key agreement for WSN using ECC. We employ ECC headed from the results in [51] indicate that ECC provides some advantages with respect to memory and computing cost, and hence is suitable for WSNs. The purpose of this part is to erase the security risk found in Das's scheme. Before executing the system, it is considered GWN is the trusted third party and it holds a master key K_j with S_j.

5.1 Pre-deployment

GWN stores the corresponding secret password-key $K_j = h(SID_j, X_{GWN})$ shared with each sensor node, where X_{GWN} is a randomly-generated highly secure password-key and is known only to the GWN and is secretly stored in the memory of the GWN. Finally, GWN deletes X_{GWN}.

5.2 User registration

RU_1. U_i first inputs his chosen identity Id_i, password pw_i and then imprints the biometrics Bio_i at the sensor of a specific device. U_i sends the registration request $\{Id_i, h(pw_i, \mu_i)\}$ to GWN, where μ_i is computed by $Gen(Bio_i) = (\mu_i, v_i)$ via a secure channel;

RU_2. GWN computes $RPW_i = h(Id_i)h(Id_i, h(K))$ and sends back it to U_i stored in the smart card, where K is a nonce;

RU_3. U_i stores $f_i = h(Id_i, h(pw_i, \mu_i))$ and RPW_i into the smart card. All the related values $\{RPW_i, f_i, v_i, Gen(), Rep(), h()\}$ are stored into the smart card. Figure 1 depicts the processes of this phase.

5.3 Sensor node registration

RS_1. S_j sends its identity SID_j to GWN;

RS_2. GWN returns $h(SId_j, K_j)$ to S_j via a secure channel;

RS_3. S_j stores $h(SId_j, K_j) \oplus N_j$ into its database.

5.4 Login

U_i inserts the smart card into terminal device and inputs his identity Id_i, password pw_i and imprints his biometrics Bio_i^* at the sensor. Next, the smart card derives μ_i with $Rep(Bio_i^*, v_i) = \mu_i$ and verifies whether $h(Id_i, h(pw_i, \mu_i)) \stackrel{?}{=} f_i$. If the check is passed, the smart card computes

$$A_i = \alpha_i P, B_i = Id_i \oplus \alpha_i X_{Pub}, V_i = h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$$

and sends $\{A_i, RPW_i, B_i, V_i, T_1\}$ to GWN, where α_i is a nonce and X_{Pub} is the public key of GWN.

5.5 Authentication and key agreement

AK_1. When receiving the request, GWN derives Id_i by computing $X_s A_i \oplus B_i$. After that, GWN computes $h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$ and checks whether it is equal to the received V_i, where X_s is the private key of GWN. If it holds, GWN computes $RPW_i^* = h(Id_i)h(Id_i, h(K^*))$, $V_j = E_{h(SId_j, K_j)}(Id_i, \alpha_i P, T_2)$ and sends them with T_2 to S_j, where K^* is a new nonce;

AK_2. Once receiving $\{V_j, RPW_i^*, T_2\}$, S_j decrypts V_j to get $(Id_i, \alpha_i P, T_2)$ and checks the validness of T_2. If it is valid, S_j further derives $h(Id_i, h(K^*))$ by computing $h(Id_i)^{-1} RPW_i^*$ and computes $C_i = E_{h(Id_i, h(K^*))}(Id_i, \beta_j P, sk, T_3)$ and $sk = h(\alpha_i \beta_j P)$. Subsequently, S_j sends $\{C_i, RPW_i^*, T_3\}$ to U_i;

AK_3. After receiving the request, U_i derives $h(Id_i, h(K^*))$ by computing $h(Id_i)^{-1} RPW_i^*$ and replaces the old temporary identity RPW_i with a new temporary identity RPW_i^* into the smart card. Then, U_i decrypts C_i using $h(Id_i, h(K^*))$ to derive $(Id_i, \beta_j P, sk, T_3)$. Next, U_i computes $sk = h(\alpha_i \beta_j P)$ and checks whether it is equal to the decrypted sk. If it is valid, U_i accepts the communication requests from S_j and agrees the session key sk as their shared session key. Figure 2 depicts the processes of this phase.

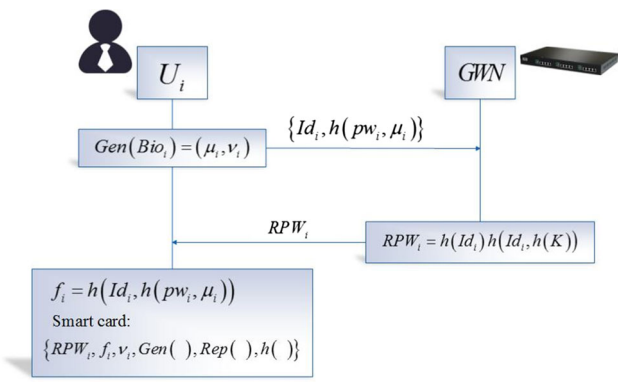

Fig. 1 User registration

5.6 Password change

The user has the ability to freely choose and if desired change his/her password in the proposed scheme. The following steps are required for a user to change his password:

P_1. U_i inserts the smart card into the device and provides his identity Id_i, old password pw_i and imprints his biometrics Bio_i^*.

P_2. The smart card derives μ_i with $Rep(Bio_i^*, v_i) = \mu_i$ and checks whether $h(Id_i, h(pw_i, \mu_i)) \stackrel{?}{=} f_i$. If it holds, U_i keys his new password pw_i, the smart card stores $f_i^{new} = h(Id_i, h(pw_i^{new}, \mu_i))$ before deleting f_i.

5.7 Dynamic sensor node addition

If some sensor nodes expire or to be added to the network, GWN updates its password-key X_{GWN} as a new X_{GWN}^{new} and stores $K_j^{new} = h(SID_j^{new}, X_{GWN}^{new})$ into its database and deletes the old information.

6 Security analysis

In this section, the security of the proposed scheme for WSN is analyzed. The BAN-logic [52] is used to show the scheme is valid and practical. Detailed analysis also shows the proposed scheme could withstand various attacks and satisfy security requirements in WSNs. Also, the formal security analysis and simulation are performed to demonstrate the proposed scheme can achieve high level security.

6.1 Authentication proof With BAN logic

To apply the BAN logic, we first introduce the basic notations listed in Table 1, where M, N as participators and X as a formula.

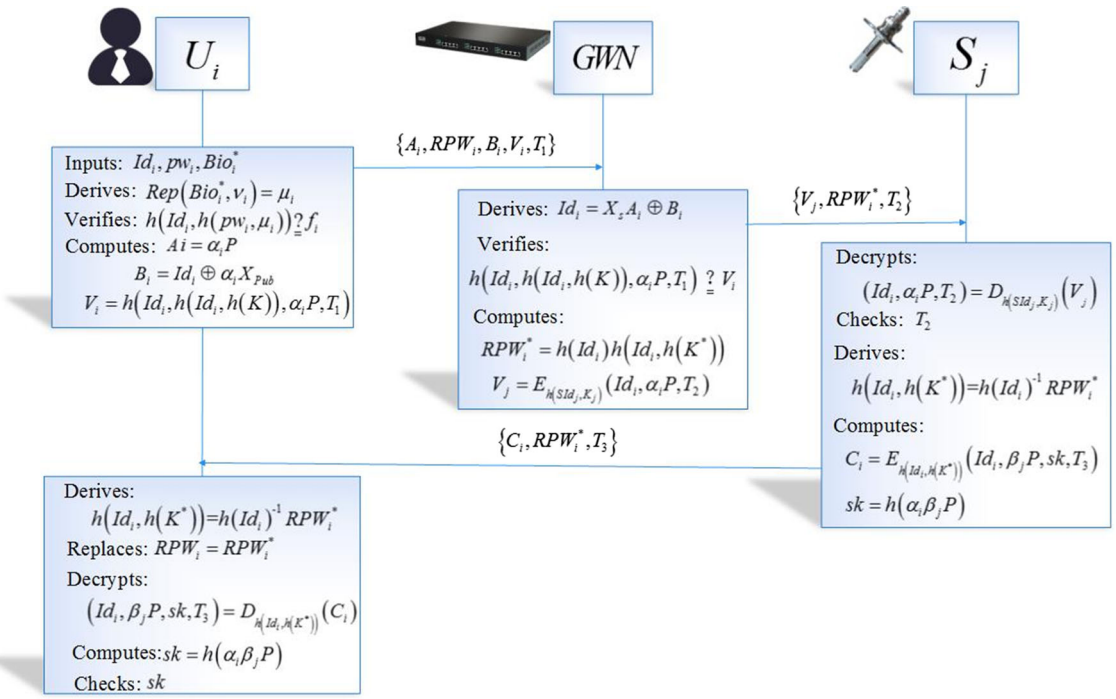

Fig. 2 Authentication and key agreement

Table 1 Notations

$\#X$	X is fresh
$M\| \equiv X$	M believes the truthfulness of X
$M\| \sim X$	M once said A
$M \lhd X$	M holds X
$M \stackrel{X}{\leftrightarrow} N$	X is a shard key between M and N
$(X, Y)_K$	X and Y are encrypted by key K
$< X >_Y$	X combined with Y
$\frac{M\| \equiv M \stackrel{K}{\leftrightarrow} N,\ M \lhd \{X\}_K}{M\| \equiv \| N \sim X}$	Message-meaning rule
$\frac{M\| \equiv \#X,\ M\| \equiv N\| \sim X}{M\| \equiv N\| \equiv X}$	Nonce-verification rule
$\frac{M\| \equiv X,\ M\| \equiv Y}{M\| \equiv (X, Y)}$	Belief rule
$\frac{M\| \equiv \#X}{M\| \equiv \#(X, Y)}$	Fresh conjuncatenation rule
$\frac{M\| \equiv N \Rightarrow X,\ M\| \equiv N\| \equiv X}{M\| \equiv X}$	Jurisdiction rule

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

G_1. $GWN| \equiv U_i| \equiv Id_i$
G_2. $GWN| \equiv Id_i$
G_3. $S_j| \equiv U_i \stackrel{sk}{\longleftrightarrow} S_j$
G_4. $U_i| \equiv S_j| \equiv U_i \stackrel{sk}{\longleftrightarrow} S_j$
G_5. $U_i| \equiv U_i \stackrel{sk}{\longleftrightarrow} S_j$
G_6. $S_j| \equiv U_i| \equiv U_i \stackrel{sk}{\longleftrightarrow} S_j$

The transmitted messages of authenticated key agreement in the idealized form are as follows:

$U_i \rightarrow GWN$: $\alpha_i P, < Id_i >_{\alpha_i X_{Pub}}, (Id_i, \alpha_i P, T_1)_{h(Id_i, h(K))}, h(Id_i)_{h(Id_i, h(K))}$;
$GWN \rightarrow S_j$: $(Id_i, \alpha_i P, T_2)_{h(SId_j, K_j)}$;
$S_j \rightarrow U_i$: $(Id_i, \beta_j P, sk, T_3)_{h(Id_i, h(K^*))}, h(Id_i)_{h(Id_i, h(K^*))}$

We then make the following assumptions to analyze the proposed scheme:

A_1: $U_i| \equiv \alpha_i$
A_2: $U_i| \equiv T_1$
A_3: $U_i| \equiv Id_i$
A_4: $GWN| \equiv K_j$
A_5: $GWN| \equiv X_s$
A_6: $GWN| \equiv T_1$
A_7: $GWN| \equiv U_i \Rightarrow Id_i$
A_8: $U_i| \equiv T_2$
A_9: $S_j| \equiv \beta_i$
A_{10}: $S_j| \equiv T_3$
A_{11}: $S_j| \equiv K_j$
A_{12}: $S_j| \equiv T_2$
A_{13}: $S_j| \equiv GWN \Rightarrow \alpha_i P$
A_{14}: $U_i| \equiv U_i \stackrel{h(Id_i, h(K))}{\longleftrightarrow} GWN$
A_{15}: $GWN| \equiv U_i \stackrel{h(Id_i, h(K))}{\longleftrightarrow} GWN$

A_{16}: $U_i| \equiv U_i \overset{h(Id_i,h(K^*))}{\longleftrightarrow} GWN$

A_{17}: $GWN| \equiv U_i \overset{h(Id_i,h(K^*))}{\longleftrightarrow} GWN$

A_{18}: $S_j| \equiv S_j \overset{h(SId_j,h(K_j))}{\longleftrightarrow} GWN$

A_{19}: $GWN| \equiv S_j \overset{h(SId_j,h(K_j))}{\longleftrightarrow} GWN$

A_{20}: $U_i| \equiv S_j \Rightarrow (U_i \overset{sk}{\longleftrightarrow} S_j)$

We now use BAN logic to analyze the proposed scheme:

S_1: Since $GWN \triangleleft (Id_i, \alpha_i P, T_1)_{h(Id_i,h(K))}$, A_{15} and the message-meaning rule, we could draw a conclusion: $GWN| \equiv U_i| \sim <Id_i, \alpha_i P, T_1>$

S_2: Since S_1, A_6 and the fresh conjuncatenation rule, we could draw a conclusion: $GWN| \equiv U_i| \equiv <Id_i, \alpha_i P>$

G_1: Since S_2 and the belief rule, we could draw a conclusion: $GWN| \equiv U_i| \equiv Id_i$

G_2: Since A_7, G_1 and the jurisdiction rule, we could draw a conclusion: $GWN| \equiv Id_i$

S_3: Since $S_j \triangleleft (Id_i, \alpha_i P, T_2)_{h(SId_j,K_j)}$, A_{18} and the message-meaning rule, we could draw a conclusion: $S_j| \equiv GWN| \sim <Id_i, \alpha_i P, T_2>$

S_4: Since A_{12}, S_3 and the fresh conjuncatenation rule, we could draw a conclusion: $S_j| \equiv GWN| \equiv <Id_i, \alpha_i P, T_2>$

S_5: Since S_4, A_{12} and the belief rule, we could draw a conclusion: $S_j| \equiv GWN| \equiv <\alpha_i P, Id_i>$

S_6: Since S_5, A_{13} and the jurisdiction rule, we could draw a conclusion: $S_j| \equiv \alpha_i P, S_j| \equiv Id_i$

According to $sk = h(\alpha_i \beta_i P)$, we could draw a conclusion:

G_3: $S_j| \equiv U_i \overset{sk}{\longleftrightarrow} S_j$

S_7: Since $U_i \triangleleft (Id_i, \beta_j P, sk, T_3)_{h(Id_i,h(K^*))}$, A_{16} and the message-meaning rule, we could draw a conclusion: $U_i| \equiv S_j| \sim <Id_i, \beta_j P, U_i \overset{sk}{\longleftrightarrow} S_j, T_3>$

S_8: Since S_7, A_3 and the fresh conjuncatenation rule, we could draw a conclusion: $U_i| \equiv S_j| \equiv <Id_i, \beta_j P, U_i \overset{sk}{\longleftrightarrow} S_j, T_3>$

G_4: Since S_8 and the belief rule, we could draw a conclusion: $U_i| \equiv S_j| \equiv U_i \overset{sk}{\longleftrightarrow} S_j$

G_5: Since A_{20}, G_4 and the jurisdiction rule, we could draw a conclusion: $U_i| \equiv U_i \overset{sk}{\longleftrightarrow} S_j$

G_6: Since S_2, S_5 and $sk = h(\alpha_i \beta_j P)$, we could draw a conclusion: $S_j| \equiv U_i| \equiv U_i \overset{sk}{\longleftrightarrow} S_j$

6.2 Discussion

This subsection shows that the proposed scheme can solve the security pitfalls troubled in Das's authentication. Besides, the proposed scheme possesses other security features.

6.2.1 Resilience against trace attack

In the proposed scheme, U_i's real identity is concealed in $B_i = Id_i \oplus \alpha_i X_{Pub}$, $V_i = h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$ and $RPW_i = h(Id_i)h(Id_i, h(K))$. If an adversary tries to learn Id_i from these authentic messages, he has to be able to know K or the private key X_s of GWN. Without the knowledge of private key X_s or the nonce K, it is impossible for an adversary to get user identity successfully. Therefore, the presented scheme can withstand trace attack.

6.2.2 Three-factor security

This topic is discussed in three sides: smart card and password, biometric and password, smart card and biometric.

a. Assume that an adversary has the user's smart card and password.

Undoubtedly, the adversary could extract the secrets $\{RPW_i, f_i, v_i, Gen(), Rep(), h()\}$ by side-channel attack [39]. However, the adversary could not derive the identity Id_i using owned information due to the lack of the nonce K and the user's biometric. Therefore, the adversary could not perform the user impersonation.

b. Assume that an adversary has the user's biometric and password.

The adversary could intercept the login message $\{A_i, RPW_i, B_i, V_i, T_1\}$ and attempt to derive Id_i from $B_i = Id_i \oplus \alpha_i X_{Pub}$. Unfortunately, the adversary has no ability to come true his wish without knowledge of the private key of GWN or the auxiliary string v_i stored in the smart card. Therefore, the adversary could not derive μ_i.

c. Assume that an adversary has the user's smart card and biometric.

The adversary also reads [50] the information $\{A_i, RPW_i, B_i, V_i, T_1\}$ from the smart card. Obviously, he could retrieve μ_i by computing $Rep(Bio_i, v_i)$. At the very least, he could derive Id_i and pw_i by checking $h(Id_i, h(pw_i^*, \mu_i^*)) \overset{?}{=} f_i$. However, he is not possible to impersonate a legal user without knowledge of the nonce K. Once modified the value $V_i = h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$, the GWN will detect the attack from the user.

6.2.3 Resilience against insider attack

An adversary plans to patch the insider attack by preferring user to submit $Id_i, h(pw_i, \mu_i)$. However, the proposed scheme could not suffer from insider attack as the insider of GWN cannot guess pw_i of U_i without μ_i. Therefore, the proposed scheme is secure against the insider attack.

6.2.4 Resilience against node capture attack

Assume that an adversary could eavesdrop the communication messages $\{V_j, RPW_i^*, T_2\}$ and $\{C_i, RPW_i^*, T_3\}$ and steal the stored information $h(SId_j, K_j) \oplus N_j$. However, it is useless to try to impersonate as a sensor node using these messages due to the lack of important secrets, such as K_j, N_j and X_s. For example, if the adversary does not know N_j, he cannot derive $h(SId_j, K_j)$ which is the verification between *GWN* and S_j. In other words, the adversary cannot be authenticated by S_j without knowledge of N_j even if he has owned the information $h(SId_j, K_j) \oplus N_j$. Therefore, the adversary tries to launch the node capture attack is hard in the proposed scheme.

6.2.5 Session key security

Even though an adversary could intercept all the communication messages $\{A_i, RPW_i, B_i, V_i, T_1\}$, $\{V_j, RPW_i^*, T_2\}$ and $\{C_i, RPW_i^*, T_3\}$, he is impossible to compromise the session key $sk = h(\alpha_i \beta_j P)$. To derive $\beta_j P$ from $C_i = E_{h(Id_i, h(K^*))}(Id_i, \beta_j P, sk, T_3)$, the adversary needs to know the user's identity Id_i and sensor node's nonce K^*. To say the least, suppose the adversary has compromised $\alpha_i P$ and $\beta_j P$, he still cannot compute sk without knowing α_i and β_j. Therefore, the proposed scheme achieves the session key security.

6.2.6 Resilience against replay attack

Let the adversary intercepts the login request $\{A_i, RPW_i, B_i, V_i, T_1\}$ during the login phase, where $A_i = \alpha_i P, B_i = Id_i \oplus \alpha_i X_{Pub}, V_i = h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$, $RPW_i = h(Id_i)h(Id_i, h(K))$. Suppose the adversary attempts to reply this login request, *GWN* will easily detect the attack by checking the validity of T_1. Further, assume that the adversary generates a new timestamp, he can be found by checking the correctness of V_1 due to T_1 is implied in V_1. As a result, the proposed scheme can prevent the replay attack.

6.2.7 Resilience against man-in-the middle attack

Assume that an adversary intercepts the login request $\{A_i, RPW_i, B_i, V_i, T_1\}$. Unfortunately, it is futile that the adversary plans to send the forged message by changing certain parameters. Once the *GWN* verifies the validity of $V_i = h(Id_i', h(Id_i, h(K))', \alpha_i' P, T_1')$, he will immediately find the attack because *GWN* computes $h(Id_i', h(Id_i, h(K)), \alpha_i' P, T_1')$ is not equal to the received V_i, where Id_i', α_i' and T_1' are the forged messages, $h(Id_i, h(K))$ is the original message only known by the user and the gateway node. Similarly, $h(SId_j, K_j)$ is also only known by the gateway node and the sensor node, any modified message V_j from *GWN* to S_j could be found by S_j. Therefore, the proposed scheme is secure against man-in-the middle attack.

6.2.8 Resilience against many logged-in users with the same login-ID attack

Even if the two users have the same identity and password, the proposed scheme could resist this attack. Since the two hash values $h(Id_i, h(pw_i, \mu_i))$ and $h(Id_i, h(pw_i, \mu_i'))$ are different, where μ_i and μ_i' are produced by the fuzzy extractor of each biometrics.

6.3 Formal security analysis

In this section, an extended security model for authentication and key agreement will be achieved to guarantee three-factor security. Specifically, we will expand the Corrupt query. The so-called three-factor security means that the adversary can get the users smart card and password, or the users biometric and password, or the users smart card and biometric, but not all.

6.3.1 Participant and partner

In our proposed scheme, there are three types of protocol participants, one is the user set $\vee_{User}$, one is the server collection $\vee_{Server}$, and the last one is gateway note(*GWN*) which is a trusted third party. There are multiple instances respectively in the user and server collections which can execute the protocol concurrently. If two instances $U_i \in \vee_{User}$ and $S_j \in \vee_{Server}$ meet the following points, then we call these two instances a partnership: (1) Both U_i and S_j are in the accepted state which means they have computed a session key respectively; (2) Both U_i and S_j have the same session identifier sid which is the connection of all messages accepted and sent by the instance U_i or S_j; (3) U_i and S_j are each other's partners.

In these participants, every user instance U_i has their own password which is chosen from a uniformly distributed dictionary D, and the gateway note (*GWN*) has a public key X_{Pub} and a private key X_s. When the user registers with the *GWN*, *GWN* computes RPW_i and sends back it to U_i stored in the smart card. When the server registers with the *GWN*, *GWN* returns $h(SId_j, K_j)$ to S_j and S_j stores $h(SId_j, K_j) \oplus N_j$ into its database.

6.3.2 Query

We make the following five queries which the adversary sends to the protocol participants U_i and S_j, and the

participants need to answer all these queries from the adversary.

(1) $Send(U_i/S_j/GWN, m)$ This query simulates the active attack. When the adversary sends this query $Send(U_i/S_j/GWN, m)$ with the message m to the instance U_i, S_j or GWN, they will answer with a corresponding response message to the adversary.

(2) $Execute(U_i, S_j)$ This query simulates the passive attack. When the adversary initiates this query, the messages will be returned to the adversary, where the messages were exchanged in the process of implementing the proposed protocol between the U_i and S_j.

(3) Re$veal(U_i)$ This query simulates the known key attack. In this query, it is required to return the U_i' session key to the adversary.

(4) $Corrupt(U_i, 3)$ This query simulates the three-factor security. Specifically, the adversary can get the users smart card and password, or the users biometric and password, or the users smart card and biometric, which depends on the value of a.

1. If a = 1, this query answers with the messages $\{RPW_i, f_i, v_i, Gen(), Rep(), h()\}$ stored in U_i' smart card and the user U_i' password pw_i to the adversary.
2. If a = 2, this query answers with the user U_i' biometric Bio_i and password pw_i to the adversary.
3. If a = 3, this query answers with the messages $\{RPW_i, f_i, v_i, Gen(), REP(), h()\}$ stored in U_i' smart card and the user U_i' biometric Bio_i to the adversary.

(5) $Test(U_i)$ This query tests the authentication key exchange security of U_i' session key. In this query, a random bit $b \in \{0, 1\}$ will be threw. If b = 1, the U_i' session key will be returned to the adversary; otherwise, if b = 0, the adversary can only learn a random value as the same length as the session key. It should be noted that the adversary can only ask this query up to once.

6.3.3 Semantic security

After the above queries, the adversary guesses the value of b related to the query $Test(U_i)$, in which the instance U_i need be ***fresh*** and the session key has been defined. We let Corr denote the event that the adversary guesses the bit b correctly and let D denote the U_i' uniformly distributed password dictionary. Therefore, the advantage of the adversary A against the semantic security of the proposed protocol **P** is defined as $Adv_{P,D}(A) = 2\Pr[Corr] - 1$.

The so-called instance ***freshness*** needs to meet the following conditions: (1)The instance U_i has computed a session key; (2)The user instance U_i and his/her partner are not be made Reveal-query; (3) The adversary can only query $Corrupt(U_i, 1)$ or $Corrupt(U_i, 2)$, but not both.

6.3.4 The security proof of our scheme

In this section, we will analyse the formal security analysis of the proposed protocol for WSN through a game between the adversary and the protocol participants, which will show that our proposed scheme is secure and practical.

Theorem 1 *$Adv_{tps,D}(A)$ represents the advantage that an adversary is against the proposed scheme under a uniformly distributed dictionary. Our proposed scheme is secure as long as the following inequality holds, i.e. the probability $Adv_{tps,D}(A)$ is small enough:*

$$Adv_{tps,D}(A) \le Adv_{E_h}^{sym}(t) + \frac{q_h^2 + (q_{send} + q_{exe})^2}{p} + \frac{2q_{send}}{p} + \frac{2q_{send}}{|D|} + 2q_h Adv_G^{CDH}(t + (q_{send} + q_{exe} + 1) \cdot \tau_G),$$

where tps, D, A, |D|, G, t and τ_G represent the proposed scheme, a uniformly distributed dictionary, an adversary against our proposed scheme, the number of elements in dictionary D, a finite cycle group, the time bound that an adversary runs against the proposed scheme and the scalar multiplication calculating time inG respectively. We denote $Adv_{E_h}^{sym}(t)$ and Adv_G^{CDH} as the advantage that an adversary is against symmetric encryption using a key which is a hash output and the advantage that an adversary solves CDH problem in G respectively. We also denote q_{send}, q_{exe} and q_h as the quantity of Send-queries, Execute-queries and Hash-queries respectively.

Proof We firstly define six mix experiments $\exp_0$ to $\exp_5$ which correspond to different situations. Then we let $Success_i$ denote the event the adversary guesses the value of b related to the Test-query, where $i = 0, 1, \ldots, 5$. What's more, we use Δ_i to denote the distance between $\exp_i$ and $\exp_{i+1}$. □

$\exp_0$: It is the first experiment corresponding to the actual attack, so in this experiment, we can get the following inequality holds by the above definition:

$$\begin{aligned} Adv_{tps,D}(A) &= 2\Pr[Success_0] - 1 \\ &= 2\Pr[Success_4] - 1 + 2(Pr[Success_0] \\ &\quad - \Pr[Success_4]) \le 2\Pr[Success_4] - 1 + 2\sum_{i=0}^{4-1} \Delta_i \end{aligned}$$

exp_1: In this experiment, we simulate various queries including two hash-queries $h(m)$ and $h'(m)$, one Send-query, one Execute-query, one Reveal-query and one Test-query, where $h'(m)$ will appear in exp_4. The simulation of these queries is shown in (1) (2) (3) respectively. The identity of the user is protected by V_j and C_i in the process of transmitting. However, once the adversary distinguishes the plain text Id_i in the cipher text V_j or C_i, he/she will break the symmetric key algorithm E_h, so we can get the following inequality to be established: $\Delta_0 \leq Adv_{E_h}^{sym}(t)$.

exp_2: In this experiment, we simulate the above queries as the same as the exp_1. In the process of simulating, we will stop executing once the transcript of the messages collides like $((A_i, RPW_i, B_i, V_i, T_1), (V_j, RPW_i^*, T_2), (C_i, RPW_i^*, T_3))$. The output of the hash queries and the transcript of the transmitted messages are both likely to collide. According to the birthday paradox, the maximum probability of hash queries is $\frac{q_h^2}{2p}$. Using the same analytical method, the maximum collision probability of the transcript on transmitted messages is $\frac{(q_{send}+q_{exe})^2}{2p}$. Therefore, we have $\Delta_1 \leq \frac{q_h^2+(q_{send}+q_{exe})^2}{2p}$.

exp_3: In this experiment, we will stop executing once the adversary guesses the values f_i, V_i and sk for authentication correctly. The two experiments exp_2 and exp_3 are indistinguishable unless the user or server rejects a valid authentication value, so we have $\Delta_2 \leq \frac{q_{send}}{p}$.

exp_4: In this experiment, we introduce and use a new hash-query h' on $\alpha_i P$ or $\beta_j P$ instead of using the hash-query h on which can get the session key. Therefore, the value of session key sk is completely independent of h and $\alpha_i\beta_j P$. Specifically, one can get $sk = h(\alpha_i P)$ or $sk = h(\beta_j P)$ in the Execute-query. We let $QueryH_{in-4}$ denote the event that the adversary makes a hash-query h on $\alpha_i\beta_j P$ in exp_4. The experiment exp_3 and exp_4 are indistinguishable unless the event $QueryH_{in-4}$ occurs. Therefore, we can derive $\Delta_3 \leq \Pr[QueryH_{in-4}]$. What's more, the choice of b related to the Test-query is random and independent of all session executions, so the equation $Pr[Success_4] = \frac{1}{2}$ holds.

exp_5: In this experiment, similarly, we firstly let $QueryH_{in-5}$ denote the event that the adversary makes a hash-query h on $\alpha_i\beta_j P$ in exp_5 as the same as $QueryH_{in-4}$ in exp_4, so we can get $\Pr[QueryH_{in-4}] = \Pr[QueryH_{in-5}]$. And then we use the random self-reducibility of the Diffie-Hellman problem to simulate the executions. Specifically, given a CDH instance(A, B), we choose two random number α, β and compute $A_i = \alpha A$ and $B_j = \beta B$. Thus we can draw $\alpha_i\beta_j P = CDH(A_i, B_j) = CDH(\alpha \cdot A, \beta \cdot B) = \alpha \cdot \beta \cdot CDH(A, B)$, where $CDH(A_i, B_j)$ or $CDH(A, B)$ is one solution of the CDH instance (A_i, B_j) or (A, B). Furthermore, if the adversary has gotten all the information existing in the user's smart card and the user U_i' biometric Bio_i, he/she will not be allowed to get the user's password. In other words, once the adversary has made a query on $Corrupt(U_i, 3)$, he/she cannot query $Corrupt(U_i, 2)$ and $Corrupt(U_i, 2)$, so in each transcript, the adversary can only test one password. Therefore, we can finally get $\Pr[QueryH_{in-5}] \leq \frac{q_{send}}{|D|} + q_h Adv_G^{CDH}(t + (q_{send} + q_{exe} + 1) \cdot \tau_G)$.

(1) *Simulation of hash-query including h() and h'()*

- On a hash-query $h(m)$, if there is a record (m, *result*) existing in a list Λ_h, then return the value of *result*, otherwise, choose an nonce *result* arbitrarily, add the record (m, *result*) to the list Λ_h, and return *result*.
- On a hash-query $h'(m')$, if there is a record $(m', result')$ existing in a list $\Lambda_{h'}$, then return the value of *result'*, otherwise, choose an nonce *result'* arbitrarily, add the record $(m', result')$ to the list $\Lambda_{h'}$, and return *result'*.

(2) *Simulation of Send-query*

- On a Send-query $Send(U_i, start)$, assuming U_i is in the correct state, we carry out the following steps: Derive μ_i with $Rep(Bio_i^*, v_i) = \mu_i$ and verify whether $h(Id_i, h(pw_i, \mu_i)) \stackrel{?}{=} f_i$. If this check is not passed, the user terminates without accepting; otherwise, choose a nonce α_i, and compute $Ai = \alpha_i P$, $B_i = Id_i \oplus \alpha_i X_{Pub}$ and $V_i = h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1)$. Then this query is answered with $\{A_i, RPW_i, B_i, V_i, T_1\}$.
- On a Send-query $Send(GWN, (A_i, RPW_i, B_i, V_i, T_1))$, assuming GWN is in the correct state, we carry out the following steps: Derive $Id_i = X_s A_i \oplus B_i$ and verify whether $h(Id_i, h(Id_i, h(K)), \alpha_i P, T_1) = V_i$ holds. If this check is not passed, the GWN terminates without accepting; otherwise, compute $RPW_i^* = h(Id_i)h(Id_i, h(K^*))$ and $V_j = E_{h(SId_j, K_j)}(Id_i, \alpha_i P, T_2)$. Then this query is answered with $\{V_j, RPW_i^*, T_2\}$.
- On a Send-query $Send(S_j, (V_j, RPW_i^*, T_2))$, assuming S_j is in the correct state, we carry out the following steps: Decrypt $(Id_i, \alpha_i P, T_2) = D_{h(SId_j, K_j)}(V_j)$ and check the validness of T_2, if this check is not passed, the server terminates without accepting; otherwise, derive $h(Id_i, h(K^*)) = h(Id_i)^{-1} RPW_i^*$ and compute $C_i = E_{h(Id_i, h(K^*))}(Id_i, \beta_j P, sk, T_3)$ and $sk = h(\alpha_i\beta_j P)$. Then this query is answered with $\{C_i, RPW_i^*, T_3\}$.
- On a Send-query $Send(U_i, (C_i, RPW_i^*, T_3))$, assuming U_i is in the correct state, we carry out the following steps: Derive $h(Id_i, h(K^*)) = h(Id_i)^{-1} RPW_i^*$, replace $RPW_i = RPW_i^*$, decrypt $(Id_i, \beta_j P, sk, T_3) = D_{h(Id_i, h(K^*))}(C_i)$, compute $sk = h(\alpha_i\beta_j P)$ and check whether it is

equal to the decrypted sk. If this check is not passed, the user terminates without accepting.

(3) *Simulation of Execute-query, Reveal-query* ***C****orrupt*(U_i, a) *and Test-query*

- On a Execute-query $Execute(U_i, S_j)$, we carry out the following steps:

$$(A_i, RPW_i, B_i, V_i, T_1) \leftarrow Send(U_i, start)$$
$$(V_j, RPW_i^*, T_2) \leftarrow Send(GWN, (A_i, RPW_i, B_i, V_i, T_1))$$
$$(C_i, RPW_i^*, T_3) \leftarrow Send(S_j, (V_j, RPW_i^*, T_2))$$

  The transcript $((A_i, RPW_i, B_i, V_i, T_1), (V_j, RPW_i^*, T_2), (C_i, RPW_i^*, T_3))$ will be returned to the adversary.
- On a Reveal-query Re$veal(U_i)$, we carry out the following steps: If the user U_i has accepted, the session key sk will be returned to the adversary.
- On a Corrupt-query $Corrupt(U_i, a)$, we carry out the following steps:
  a. If a = 1, the messages $\{RPW_i, f_i, v_i, Gen(), REP(), h()\}$ stored in U_i' smart card and the user U_i' password pw_i will be returned to the adversary.
  b. If a = 2, the user U_i' biometric Bio_i and password pw_i will be returned to the adversary.
  c. If a = 3, the messages $\{RPW_i, f_i, v_i, Gen(), REP(), h()\}$ stored in U_i' smart card and the user U_i' biometric Bio_i will be returned to the adversary.
- On a Test-query $Test(U_i)$, we carry out the following steps: Throw a coin $b \in \{0, 1\}$. If b=1, we return the session key sk of the user which is obtained from Re$veal(U_i)$ query; otherwise, we return a random value as the same length as the session key sk.

6.4 Simulation results using AVISPA tool

AVISPA (Automated Validation of Internet Security Protocols and Applications) Tool is a push-button tool used to verify the robustness and efficiency of a cryptographic protocol. And it provides a modular role-based expressive formal language called the HLPSL (High level protocol specification language) for implementing security protocols. A HLPSL specification is translated into the Intermediate Format (IF), using a translator called HLPSL2IF. The IF specification of a protocol is then input to the backends of the AVISPA tool to analyze if the security goals are satisfied or not. The AVISPA tool comprises four backends: On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

To evaluate the security of the proposed scheme by the AVISPA tool, the specifications for the user U_i, the sensor node S_j , the gate-way node GWN, the session, goal and the environment have been implemented in HLPSL. The proposed scheme is simulated under the OFMC and CL-AtSe backends using the SPAN a security protocol animator for AVISPA. The designd goals, GWN authenticates U_i through Id_i, S_j authenticates U_i through Id_i, the secrecy of session key, user's identity and password are all achieved. In OFMC backend, as illustrated in Fig. 3, the total number of visited nodes is 9657 and the depth of search is 13 which require 51.42 seconds. In CL-AtSe backend, as shown in Fig. 4, 8621 states were analyzed and 1573 states were reachable. Further, CL-AtSe backend took 0.53 seconds for translation and 0.91 s for computation. Simulation results demonstrate that the proposed scheme is SAFE.

7 Performance comparison

This section compares the performance of the proposed scheme with other four biometrics-based schemes using elliptic curve cryptography.

Table 2 shows the comparison of the security features among the proposed scheme and other biometrics-based schemes, such as He and Wang's scheme [46], Xie et al.'s scheme [39], Jiang et al.'s scheme [44] and Li et al.'s

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/l-aka.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 51.42s
 visitedNodes: 9657 nodes
 depth: 13 plies
```

Fig. 3 The result of OFMC backend

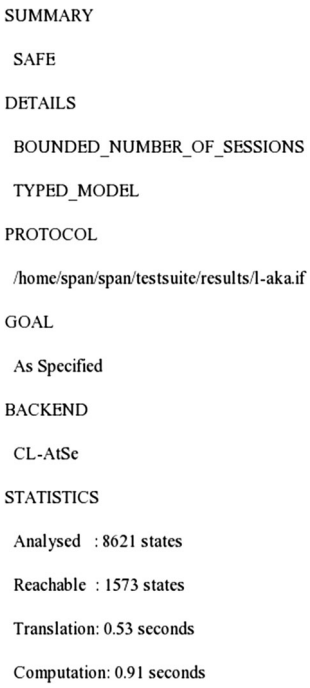

Fig. 4 The result of CL-AtSe backend

scheme [47]. It is clear that all these schemes cannot resist many logged-in users with the same login-ID attack. He and Wang's[46] scheme even cannot provide session key security. Besides, Xie et al.'s scheme [39], Jiang et al.'s scheme [44] and Li et al.'s scheme [47] are unable to provide three-factor security. Both the two schemes [44, 47] are vulnerable to man-in-the middle attack. From the table, it can be see that the proposed scheme could provide the desired security features and defend various existing attacks.

Table 3 has compared the computation cost of the proposed scheme with other four schemes during registration, login and authentication, and key agreement phases. The following notations for computing the computational costs of the proposed scheme and other schemes: H: hash computation, S: symmetric operation, E: elliptical curve scale multiplication operation. According to Table 3, it can be see that the proposed scheme needs a slight higher computation cost than the schemes [39] and [47] but lower than the schemes [44] and [47]. To make a comparison more clearly, Fig. 5 shows the comparison graph for the computation cost in login and authentication(LAA) phases and the total cost(TC). Here, we utilize the arithmetic mean to perform the primitive operations for thousand executions each based on the jPBC library 2.0.0 [53], where the experiment lays on Windows 10 operating system, Pentium 3.20 GHz CPU, and 4.0 GB RAM. The running time of H is 0.0359 ms, E is 10.5129 ms and S is 0.1755 ms.

Generally speaking, the security of the scheme is the first important. Thus, we consider that the proposed scheme reaches a balance between efficiency and security properties.

Table 2 Security features comparison

	Proposed	[46]	[39]	[44]	[47]
Resist insider attack	Yes	Yes	Yes	Yes	Yes
Provide user anonymity	Yes	Yes	Yes	Yes	Yes
Provide three-factor security	Yes	Yes	No	No	No
Provide mutual authentication	Yes	Yes	Yes	Yes	Yes
Provide session key security	Yes	No	Yes	Yes	Yes
Resist replay attack	Yes	Yes	Yes	Yes	Yes
Resist many logged-in users with the same login-ID attack	Yes	No	No	No	No
Resist man-in-the middle attack	Yes	Yes	Yes	No	No
Resist user impersonation attack	Yes	Yes	Yes	Yes	Yes

Table 3 Computational cost comparison

	Proposed	[46]	[39]	[44]	[47]
User registration	5H	2H	5H	4H+1E	5H
Server registration		1H			
Sensor node registration	2H				1H
Login and authentication	13H+6E+4S	23H+8E	13H+4E+4S	12H+6E	22H+3E
Total cost	20H+6E+4S	26H+8E	18H+4E+4S	16H+7E	28H+3E

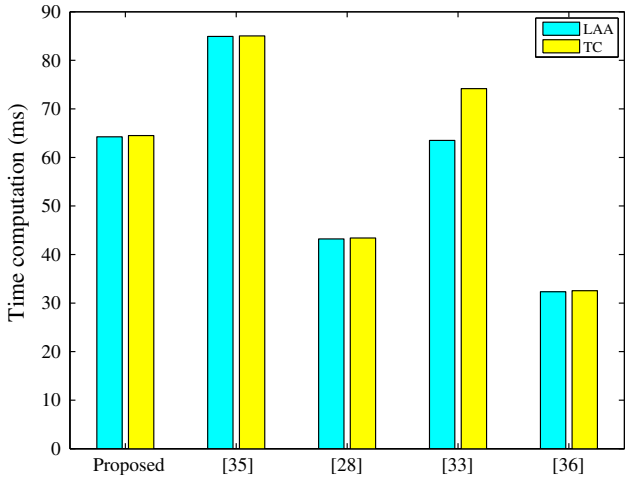

Fig. 5 Performance comparison

8 Conclusion

This paper has shown that the Das's mutual authentication scheme for WSN has several security pitfalls and may suffer from some attacks. In the Das's scheme, there is no provision of user anonymity, it is susceptible to the user impersonation attack due to the failure of three-factor security. Thus, this paper provides improvements to fix the two loopholes so that the proposed scheme should be secure enough to be used in WSNs.

References

1. Hayajneh, T., Doomun, R., Al-Mashaqbeh, G., & Mohd, B. J. (2014). An energy efficient and security aware route selection protocol for wireless sensor networks. *Security and Communication Networks*, *7*(11), 2015–2038.
2. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications* (pp. 88–97).
3. Otto, C., Milenkovic, A., Sanders, C., & Jovanov, E. (2006). System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, *1*(4), 307–326.
4. Hayajneh, T., Mohd, B. J., Imran, M., Almashaqbeh, G., & Vasilakos, A. V. (2016). Secure authentication for remote patient monitoring with wireless medical sensor networks. *Sensors*, *16*(4), 424.
5. ARGUS, U. A. F. (2009). Advanced remote ground unattended sensor systems. Department of Defense. http://www.globalsecurity.org/intell/systems/arguss.htm.
6. Carlson, J., Han, R., Lao, S., Narayan, C., & Ghani, S. (2003). Rapid prototyping of mobile input devices using wireless sensor nodes. In *Proceedings of the 5th IEEE workshop on mobile computing systems and applications (WMCSA '03)* (pp. 21–29).
7. Chen, J., Salim, M., & Matsumoto, M. (2011). A single mobile target tracking in voronoi-based clustered wireless sensor network. *Journal of Information Processing Systems*, *7*(1), 17–28.
8. Akhtar, R., Leng, S., Memon, I., Ali, M., & Zhang, L. (2015). Architecture of hybrid mobile social networks for efficient content delivery. *Wireless Personal Communications*, *80*(1), 85–96.
9. Claycomb, W., & Shin, D. (2011). A novel node level security policy framework for wireless sensor networks. *Journal of Network and Computer Applications*, *34*, 418–428.
10. Memon, I., Ali, Q., Zubedi, A., & Mangi, F. A. (2017). DPMM: Dynamic pseudonym-based multiple mix-zones generation for mobile traveler. *Multimedia Tools and Applications*, *76*(22), 24359–24388.
11. Sastry, N., & Wagner, D. (2004). Security considerations for IEEE 802.15.4 networks. In *Proceedings of the ACM workshop wireless security* (pp. 32–42). ACM Press.
12. Perrig, A., Szewczyk, R., Wen, V., Culler, D., & Tygar, D. (2002). SPINS: Security protocols for sensor networks. *ACM Wireless Networks*, *8*(5), 521–534.
13. 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.
14. He, D., Bu, J., Zhu, S., Chan, S., & Chen, C. (2011). Distributed access control with privacy support in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *10*(10), 3472–3481.
15. Xia, Z. H., Wang, X. H., Zhang, L. G., Qin, X., Sun, X. M., & Ren, K. (2016). A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. *IEEE Transactions on Information Forensics and Security*, *11*(11), 2594–2608. https://doi.org/10.1109/TIFS.2016.2590944.
16. Fu, Z., Ren, K., Shu, J., et al. (2016). Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Transactions on Parallel and Distributed Systems*, *27*(9), 2546–2559.
17. Memon, I., Mohammed, M. R., Akhtar, R., Memon, H., Memon, M. H., & Shaikh, R. A. (2014). Design and implementation to authentication over a GSM system using certificate-less public key cryptography (CL-PKC). *Wireless Personal Communications*, *79*(1), 661–686.
18. Jiang, Q., Ma, J. F., Yang, C., Ma, X. D., Shen, J., & Chaudhry, S. A. (2017). Efficient end-to-and authentication protocol for wearable health monitoring systems. *Computers & Electrical Engineering*. https://doi.org/10.1016/j.compeleceng.2017.03.016.
19. Memon, I. (2015). A secure and efficient communication scheme with authenticated key establishment protocol for road networks. *Wireless Personal Communications*, *85*(3), 1167–1191.
20. Arain, Q. A., Zhongliang, D., Memon, I., Arain, S., Shaikh, F. K., Zubedi, A., et al. (2017). Privacy preserving dynamic pseudonym-based multiple mix-zones authentication protocol over road networks. *Wireless Personal Communications*, *95*(2), 505–521.
21. Memon, I., Arain, Q. A., Memon, H., & Mangi, F. A. (2017). Efficient user based authentication protocol for location based services discovery over road networks. *Wireless Personal Communications*, *95*(4), 3713–3732.
22. Chen, C., Huang, H., Liu, C., & Lai, C. (2014). User authentication with anonymity fore roaming service with smart cards in global mobility networks. *Ad-Hoc & Sensor Wireless Networks*, *20*(1–2), 5–19.
23. Memon, I. (2015). Authentication user's privacy: An integrating location privacy protection algorithm for secure moving objects in location based services. *Wireless Personal Communications*, *82*(3), 1585–1600.
24. Memon, I., Hussain, I., Akhtar, R., & Chen, G. (2015). Enhanced privacy and authentication: An efficient and secure anonymous communication for location based service using asymmetric cryptography scheme. *Wireless Personal Communications*, *84*(2), 1487–1508.

25. Kamenyi, D. M., Wang, Y., Zhang, F., Memon, I., & Gustav, Y. H. (2013). Authenticated privacy preserving for continuous query in location based services. *Journal of Computational Information Systems*, *9*(24), 9857–9864.
26. Das, M. (2009). Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *8*(3), 1086–1090.
27. Xie, Q., Wong, D. S., Wang, G., Tan, X., Chen, K. F., & Fang, L. M. (2017). Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model. *IEEE Transactions on Information Forensics & Security*, *12*(6), 1382–1392.
28. Wang, D., & Wang, P. (2016). Two birds with one stone: two-factor authentication with security beyond conventional bound. *IEEE Trans. on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2016.2605087.
29. Wang, D., He, D. B., Wang, P., & Chu, C. H. (2015). Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Transactions on Dependable and Secure Computing*, *12*(4), 228–442.
30. Nyang, D. H., & Lee, M. K. (2009). Improvement of Das's two-factor authentication protocol in wireless sensor networks. *In Cryptology ePrint Archive*, 631.
31. Khan, M. K., & Alghathbar, K. (2010). Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. *Sensors*, *10*(3), 2450–2459.
32. Yuan, J. (2014). An enhanced two-factor user authentication in wireless sensor networks. *Telecommunication Systems*, *55*(1), 105–113.
33. Wang, D., & Wang, P. (2014). Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Networks*, *20*, 1–15. https://doi.org/10.1016/j.adhoc.2014.03.003.
34. Xie, Q., Dong, N., Wong, D. S., & Hu, B. (2016). Cryptanalysis and security enhancement of a two-factor authentication and key agreement protocol. *International Journal of Communication Systems*, *29*(3), 478–487.
35. Sun, D., Li, J., Feng, Z., Cao, Z., & Xu, G. (2013). On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. *Personal and Ubiquitous Computing*, *17*(5), 895–905.
36. Turkanović, M., Brumen, B., & Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks*, *20*, 96–112.
37. Jiang, Q., Zeadally, S., Ma, J. F., & He, D. B. (2017). Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks. *IEEE Access*, *5*(1), 3376–3392.
38. Xue, K., Ma, C., Hong, P., et al. (2013). A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *Journal of Network and Computer Applications*, *36*(1), 316–323.
39. Xie, Q., Tang, Z. X., & Chen, K. F. (2017). Cryptanalysis and improvement on anonymous three-factor authentication scheme for mobile networks. *Computers and Electrical Engineering*, *59*, 218–230.
40. Das, A. (2015). A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wireless Personal Communications*, *82*(3), 1377–1404.
41. Yuan, C. S., Sun, X. M., & Lv, R. (2016). Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Communications*, *13*(7), 60–65. https://doi.org/10.1109/CC.2016.7559076.
42. Wang, D., Cheng, H. B., Wang, P., Huang, X. Y., & Jian, G. P. (2017). Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, *12*(11), 2776–2791.
43. Li, C., & Hwang, M. (2010). An efficient biometric-based remote authentication scheme using smart cards. *Journal of Network and Computer Applications*, *33*(1), 1–5.
44. Jiang, Q., Chen, Z., Li, B., et al. (2017). Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-017-0516-2.
45. Dodis, Y., Reyzin, L., Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Advances in cryptology-Eurocrypt*, 523–540.
46. He, D., & Wang, D. (2014). Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal*, *9*(3), 816–823.
47. Li, X., Niu, J., Kumari, S., et al. (2017). A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *Journal of Network and Computer Applications*. https://doi.org/10.1016/j.jnca.2017.07.001.
48. Das, A. (2017). A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *International Journal of Communication Systems*. https://doi.org/10.1002/dac.2933.
49. Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, *24*(11), 770–772.
50. Kim, T., Kim, C., & Park, I. (2012). Side channel analysis attacks using AM demodulation on commercial smart cards with SEED. *Journal of Systems and Software*, *85*(12), 2899–2908.
51. Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Cryptographic hardware and embedded systems-CHES* (pp. 119–132).
52. Burrow, M., Abadi, M., & Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, *8*, 18–36.
53. Java Pairing Based Cryptography Library (jPBC). Available at http://gas.dia.unisa.it/projects/jpbc.

Yanrong Lu received the M.S. degree in cryptography from Xidian University, Xi'an, China , in 2012, and the Ph.D. degree in cryptography from Beijing University of Posts and Telecommunications of China, Beijing, China, in 2017. She is an assistant professor in Tianjin University, Tianjin, China. Her research interests is focused on information security and cryptography, in particular, cryptographic protocols.

Guangquan Xu is an associate professor in cyber security at the Tianjin University, and the head of Network Security Joint Lab in TJU. He received his Ph.D. in 2008 from the Tianjin University, and then joined School of Compuer Science and Technology, Tianjin University. He was selected as HIF fellow by Hosei University during 2013.4–2014.4, and was appointed as the vice director of the administration committee of Shihezi High Tech. Zone since 2015.12 and returned to Tianjin University on December 1, 2016. His research is in cyber security, especially offensive and defensive practice. His research work focuses on mobile security / intelligence security, web security, trusted computing, network security, and so on

Lixiang Li received the M.S. degree in circuit and system from Yanshan University, Qinhuangdao, China, in 2003, and the Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2006. She is currently a professor at the School of Computer Science, Beijing University of Posts and Telecommunications, China. Her research interests include swarm intelligence, information security and network security. She is the co-author of 70 scientific papers and 10 Chinese patents.

Yixian Yang received the M.S. degree in applied mathematics in 1986 and the Ph.D. degree in electronics and communication systems in 1988 from Beijing University of Posts and Telecommunications, Beijing, China. He is the Managing Director of information security center, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include network security, information security and coding theory. He is the co-author of 300 scientific articles and 50 patents.