


Article

An Enhanced Three-Factor User Authentication Scheme Using Elliptic Curve Cryptosystem for Wireless Sensor Networks

Chenyu Wang , Guoai Xu * and Jing Sun

New Research Activities Department, Beijing University of Posts and Telecommunications, Haidian District, Beijing 100876, China; wangchenyu@bupt.edu.cn (C.W.); sunjing@bupt.edu.cn (J.S.)

* Correspondence: xga@bupt.edu.cn

Received: 11 November 2017; Accepted: 4 December 2017; Published: 19 December 2017

Abstract: As an essential part of Internet of Things (IoT), wireless sensor networks (WSNs) have touched every aspect of our lives, such as health monitoring, environmental monitoring and traffic monitoring. However, due to its openness, wireless sensor networks are vulnerable to various security threats. User authentication, as the first fundamental step to protect systems from various attacks, has attracted much attention. Numerous user authentication protocols armed with formal proof are springing up. Recently, two biometric-based schemes were proposed with confidence to be resistant to the known attacks including offline dictionary attack, impersonation attack and so on. However, after a scrutinization of these two schemes, we found them not secure enough as claimed, and then demonstrated that these schemes suffer from various attacks, such as offline dictionary attack, impersonation attack, no user anonymity, no forward secrecy, etc. Furthermore, we proposed an enhanced scheme to overcome the identified weaknesses, and proved its security via Burrows–Abadi–Needham (BAN) logic and the heuristic analysis. Finally, we compared our scheme with other related schemes, and the results showed the superiority of our scheme.

Keywords: user authentication; smart card; offline dictionary attack

1. Introduction

With its strong self-organization, low-cost, resource-limited and data-centered, wireless sensor networks (WSNs) have been widely deployed in harsh environments such as military, industrial, transportation and even battlefields. Different to some systems such as the distributed architectures [1,2], there are three participants in WSNs. Each participant has different computational and storage power, and only the gateway can store the long-term key. Furthermore, most sensor nodes are distributed in an unattended environment, which means the sensor node is prone to be attacked. It also should be noted that the communications between users and sensor nodes are usually in an open channel, and the adversary can eavesdrop on or modify messages in the network. Therefore, the privacy and security of WSNs are always the thorny and vital issues. To deal with these security issues, it is a common practice to establish a security mechanism to share secret key between communicating parties and encrypt the data from remote parties. In this context, the remote user authentication protocol [3–7] with a session key is an essential security strategy for a secure and practical communication over an untrusted but complicated network. It guarantees that the communicating parties can verify the validity of each other and negotiate a session key for encrypting the future transmitted messages. The major challenge in designing an authentication protocol in WSNs is to balance the relationship between security, privacy and computational cost.

Generally, we authenticate a remote user from three aspects: what he knows, such as password; what he owns, such as a smart card; who he is, such as biometrics. A scheme using “X” aspects to verify

the remote user is called “X-factor” authentication protocol. With the development of biotechnology and the increasing demands on security, three-factor (password + smart card + biometrics) user authentication scheme gets widely applied.

1.1. Related Works

In 2009, Das [8] introduced a password-based scheme with a smart card for WSNs; it then aroused an intense discussion and greatly promoted the development of user authentication in WSNs. Many researchers [4,9–11] identified the security pitfalls in Das’s scheme [8] (such as being prone to offline password guessing attack, impersonation attack and insider attack), and then proposed many enhanced versions. However, none of these schemes was secure enough to resist against various attacks or achieved low computational cost.

In 2011, Fan et al. [12] criticized the weakness of previous schemes and designed a new scheme with lightweight operations. With lower computational cost, their scheme seems quite suitable for a resources-limited environment such as WSNs. In 2012, Das et al. [13] proposed a new scheme which supports the dynamical addition of new nodes and only involves some lightweight operations. It has to be admitted that Das et al.’s scheme provides many desired attributes. Unfortunately, Wang et al. [14] identified that the two schemes both are vulnerable to many attacks: Fan et al.’s scheme [12] can neither achieve user anonymity, nor avoid smart card lost attack and insider attack, etc.; Das et al.’s scheme cannot resist against insider attack, smart card lost attack, etc.

In 2013, Xue et al. [15] introduced an efficient authentication scheme with admirable features and lightweight computational cost. However, it was revealed by Wang et al. [3] that this scheme fails to achieve user anonymity. Furthermore, Li et al. [16] demonstrated its vulnerability to offline dictionary attack, insider attack, stolen-verifier attack, etc., and proposed a new scheme which is still insecure against offline dictionary attack. In the same year, Li et al. [17] identified the weakness (not resistant to dictionary attack and session key disclosure attack, etc.) in Yeh et al.’s scheme [18].

In 2014, Choi et al. [19] showed that a previous scheme [20] suffers from sensor energy exhausting attack, offline password guessing attack and the session key attack, and then proposed a new scheme. After demonstrating the security flaws in Xue et al.’s scheme [15], Jiang et al. [21] also designed an improved one. However, both the scheme of Choi et al. [19] and Jiang et al. [21] were discovered as not being secure as claimed by Wu et al. [22].

In 2015, He et al. [23] described a temporal-credential-based scheme for WSNs, yet soon was pointed out subject to impersonation attack, smart card lost attack and tracking attack. In the same year, Chang et al. [24] proposed an enhanced dynamic identity authentication, once again, it was proved not secure against offline password guessing attack, user impersonation attack, etc. by Jung et al. [25] and Park et al. [26]. To strengthen the security of the scheme, Jung et al. [25] and Park et al. [26] both added the biological characteristic as a new factor and proposed a three-factor enhanced version. Furthermore, they both proved the security of their scheme formally, so they were confident in the security of their scheme.

1.2. Motivations and Contributions

When revisiting Jung et al.’s scheme [25] and Park et al.’s scheme [26], it was regretful to find that the two schemes are still not as secure as claimed, though they both are equipped with the complete formal proof, and furthermore, add a biometric factor into the scheme to improve the security of the previous scheme. Ridiculously, the improved two schemes that are armed with a biometric factor and a formal proof, even cannot provide the same level security assurance as the previous ones. We find them vulnerable to offline password guessing attack, impersonation attack, and no user anonymity, no forward security, etc.

In fact, it is pretty common that a scheme with formal security proof was found insecure. Though the user authentication in wireless sensor networks have been developed over almost ten years since Das [8] first proposed a two-factor scheme, there is not yet a secure and practical scheme. Even more

alarming is the fact that many schemes violate some basic design principles that have been proposed. Such an unsatisfactory situation prompts us to design a secure but efficient scheme for wireless sensor networks. Furthermore, the common consensus on the system architecture, adversary model and security requirements should be reached. In conclusion, our contributions are as follows:

1. We depict the system architecture, adversary model and security requirements of wireless sensor networks. Though these factors are the basis of the authentication scheme, researchers usually ignore them.
2. We demonstrate that: (1) Jung et al.'s scheme cannot resist against offline password guessing attack, impersonation attack, and fails to achieve user anonymity and forward secrecy, etc.; (2) Park et al.'s scheme suffers from offline password guessing attack, and no user anonymity. Furthermore, we explain the inherent reason for these attacks.
3. We propose an improved scheme with various desirable attributes, and prove its security via BAN logic and heuristic analysis. Then, we compare our scheme with other related schemes. The results show the great advantage of our scheme.

1.3. Organization of the Paper

The remainder of this paper is organized as follows: we describe the system architecture and adversary model in Section 2, analyze Jung et al.'s scheme and Park et al.'s scheme in Sections 3 and 4, respectively; in Section 5, we propose an enhanced scheme; the security and performance analysis are given in Sections 6 and 7, respectively; and the conclusions are drawn in Section 8.

2. Preliminaries

This section introduces the preliminaries in the user authentication scheme including computational problems, system architecture, adversary model and security requirements.

2.1. Computational Problems

Given two large primes p and q , let \mathbb{F}_p be a finite field, E/\mathbb{F}_p be an elliptic curve over \mathbb{F}_p , and \mathbb{G} be a q -order subgroup of E/\mathbb{F}_p . Then, for $\alpha, \beta \in \mathbb{Z}_p^*$ and a point P in \mathbb{G} , we can define the discrete logarithm problem over the elliptic curve as follows:

1. Elliptic curve discrete logarithm problem: given $(P, \alpha P)$, it is impossible to compute α within polynomial time.
2. Elliptic curve computational Diffie–Hellman problem: given $(\alpha P, \beta P)$, it is impossible to compute $\alpha\beta P$ within polynomial time.

2.2. System Architecture

Wireless sensor networks (as shown in Figure 1) attract worldwide attention with the prevalence of Internet of Things (IoT). Generally, people may be more familiar with distributed systems, which involve two participants: a set of users and a single server, while there are three participants in the user authentication of WSNs: a number of sensor nodes, a gateway node and a set of users. In a wireless sensor network, there are tens to thousands of sensor nodes that are deployed in a particular area. They work together to collect the data from physical world and have limited computing and storage power. Furthermore, they are usually left in an unattended environment, so the adversary can easily capture them to acquire secret parameters. The gateway node acts like a registration center. In WSNs, an authentication protocol usually consists of four basic phases: registration, login, verification, and password update. Sometimes, the dynamic node addition phase is suggested for meeting the demand on increasing new sensor nodes. In the registration phase, users and sensor nodes submit their personal information to the gateway, then the gateway will issue users a smart card with some sensitive parameters physically (face to face or via the mail), and distribute a shared secret key to sensor nodes. When a user wants to access a sensor node, he/she can initialize an access request

to the gateway in the login phase. After checking the legitimacy of the user, the gateway informs the corresponding sensor node about the request. Then, the user and the sensor node verify the legitimacy of each other via (or not) the gateway and negotiate a session key in the verification phase. The user can change the password in the password update phase. In addition, the new sensor nodes can join the network in the dynamic node addition phase.

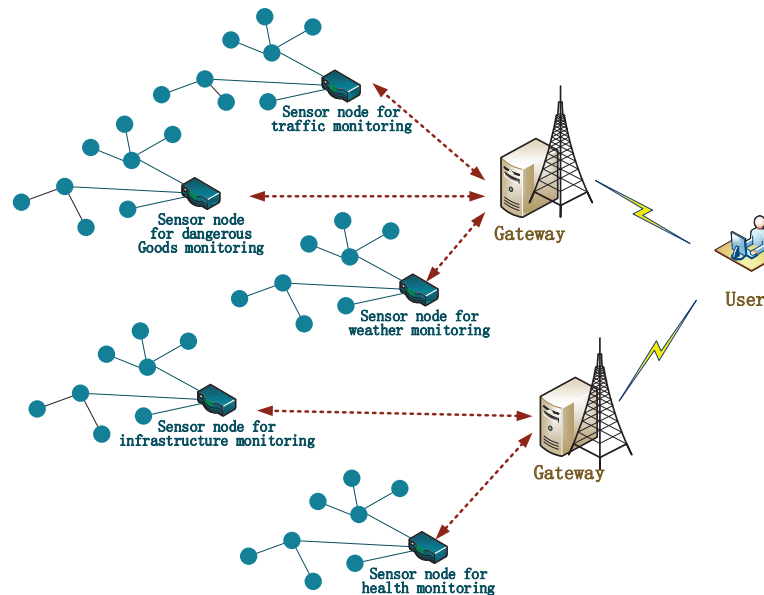


Figure 1. WSNs system architecture.

2.3. Adversary Models

When considering cryptanalysis of the user authentication schemes in WSNs, the adversary \mathcal{A} is also supposed to have the following capacities:

1. \mathcal{A} can fully control the open communication channel, i.e., \mathcal{A} can modify, intercept, delete, and resend the eavesdropped on messages over an open channel [9,27].
2. \mathcal{A} can enumerate all the items in $\mathcal{D}_{pw} * \mathcal{D}_{id}$ in polynomial time, where \mathcal{D}_{pw} and \mathcal{D}_{id} denote the password space and the identity space, respectively [28,29].
3. When evaluating forward secrecy, \mathcal{A} can get the long-term secret key [28,30].
4. \mathcal{A} can acquire the password of a legitimate user by a malicious card reader, or get the parameters in the smart card, but cannot achieve both [28,30].
5. \mathcal{A} can get the data in sensor nodes for they are usually left unattended [3,31].
6. \mathcal{A} can get the past session keys [30].
7. \mathcal{A} can get the user's biometrics [29,32].

The capacity of acquiring biometrics is the most controversial. Many researchers view it as a quite strong factor that cannot be broken. However, this is impractical. For example, the adversary can at least get the biometrics via a malicious terminal. Moreover, unlike the password that may change with the different applications, the biometrics is unique to every particular person. Thus, the adversary can collect one's biometrics via any biometric-based terminal. This indicates that the adversary can acquire the password and biometrics both, or the smart card and the biometrics both. Furthermore, this hypothesis has been accepted in many schemes, such as [29,32,33].

It should be noted that: a secure three-factor authentication scheme should guarantee that the breaking of any two of the three factors will not affect the other one, and the system is still secure.

2.4. Security Requirements

Understanding the security requirements of the user authentication is a fundamental step to analyze or design a protocol. Thus, we summarize the security requirements of user authentication in the wireless sensor network:

- S1** Mutual authentication. It is an essential requirement in all authentication schemes. It requires the participants to authenticate each other [34,35].
- S2** User anonymity. It is a privacy protection requirement for individual users, not directly related to system security. Many systems have such a requirement including distributed system [36]. While the privacy protection in wireless sensor networks is more severe, since the information among sensor nodes (usually unreliable) is transmitted in a way of broadcasting. Protecting user anonymity is to stop \mathcal{A} from computing the user's identity or linking the transcript to a same user. Note that such a requirement is not applied to the gateway, but to sensor nodes for they are untrusted.
- S3** Key agreement. It is also an essential requirement in most authentication schemes. The session key is used to encrypt the further communications to achieve confidentiality.
- S4** Forward secrecy. It is for the final collapse of the whole system, and it requires that the previous communications will be secure, even the system collapses (usually refers to the adversary that owns the long-term key of the system).
- S5** User friendly. It is an additional requirement to improve the user experience with the development of the network. A user friendly scheme usually includes: let the user U_i select the password freely, and change it locally [30]; when U_i finds the smart card insecure, let he/she revoke it and re-register to the system with original identity.
- S6** No stolen-verifier attack. It is a requirement related to the security of the whole system (so as the following attacks), which requires that the verifier table does not expose any sensitive information for \mathcal{A} to impersonate the participants or learn/control the session key.
- S7** No insider attack. It requires that the participants cannot get any sensitive information, which may provoke an attack.
- S8** No dictionary attack. It requires that \mathcal{A} cannot conduct a brute force attack.
- S9** No replay attack. It stops \mathcal{A} from conducting an attack via replaying the history message, which requires that the participants can check the freshness and validity of the received message.
- S10** No parallel session attack. This requirement is a bit similar to the replay attack, but it considers a condition where \mathcal{A} conducts an attack via initiating multi-session simultaneously.
- S11** No de-synchronization attack. The synchronization attack in wireless sensor networks is more destructive than that in traditional networks, since a gateway may connect even hundreds of sensor nodes. It requires that the parameters among corresponding participants are consistent.
- S12** No impersonation attack. It is a very important requirement in authentication, which requires that the outside adversary (inside adversary has been considered in insider attack) will not be able to impersonate any participants. A scheme resistant to impersonation attack requires that the participants verify whether the corresponding communication party is a counterfeit one. Note that: the occasion where \mathcal{A} performs a user impersonation attack using the password from a dictionary attack is not included—such an attack belongs to dictionary attack.
- S13** No known key attack. It is an attack related to the session key, which requires \mathcal{A} , who knows that the current session key cannot compute the keys in others.

3. Cryptanalysis of Jung et al.'s Scheme

In 2017, Jung et al. [25] demonstrated several attacks against Chang et al.'s [24] two-factor user authentication scheme in WSNs. To improve the security and practicability of the scheme, they devised an enhanced one over Chang et al.'s scheme [24] by "employing biometrics information with the biohashing technique". They proved their scheme secure to various attacks such as offline dictionary attack using the Burrows-Abadi-Needham (BAN) logic. However, as we will show in this section,

Jung et al.'s scheme still suffers from offline dictionary attack, impersonation attack, etc., which is even less secure than the previous one. For convenience of illustration, some notations are listed in Table 1.

Table 1. Notations and abbreviations.

Symbol	Description
U_i	i th user
GW	the gateway node
S_j	j th sensor node
\mathcal{A}	malicious attacker
ID_i	identity of user U_i
PW_i	password of user U_i
BIO_i	biometrics of user U_i
P_j	the shared secret key between GW and S_j
x, y	the secret key of remote server GW
\oplus	the bitwise exclusive OR (XOR) operation
\parallel	the string concatenation operation
$H(BIO_i)$	collision free one-way hash function to the biometrics
$h(\cdot)$	collision free one-way hash function
$Gen(BIO_i)$	one part of fuzzy extraction function, output a biometric key R_i and a helper string P_i
$Rep(BIO_i, P_i)$	one part of fuzzy extraction function, output the biometric key R_i in $Gen(BIO_i)$
\rightarrow	a insecure channel
\Rightarrow	a secure channel

3.1. A Brief Review of Jung et al.'s Scheme

In this section, we review Jung et al.'s scheme [25] briefly, their scheme consists of four phases: registration, login, verification and password change. The password change phase was omitted, since it has little relevance to this work.

3.1.1. Registration Phase

In Jung et al.'s paper, there is only a user registration phase as follows:

1. $U_i \Rightarrow GW: \{TID_i, HPW_i\}$, where $TID_i = h(ID_i || u)$, $HPW_i = h(PW_i || H(BIO_i))$ and u is a random number chosen by the user U_i .
2. $GW \Rightarrow U_i$: a smart card containing $\{A_i, E_i, C_i, h(\cdot), H(\cdot)\}$, where $HID_i = h(TID_i || x) \oplus HPW_i$, $A_i = h(HPW_i || TID_i) \oplus HID_i$, $E_i = h(HPW_i || HID_i)$, $C_i = HID_i \oplus x$.
3. U_i stores D_i in the smart card, where $D_i = u \oplus H(BIO_i)$.

However, according to the paper, the sensor node S_j preserves a private key X_{S_j} . So we deduce that the sensor node registration phase was missed. For the integrity, we add it as below:

1. $S_j \Rightarrow GW: \{SID_j\}$.
2. $GW \Rightarrow S_j: X_{S_j} = h(SID_j || x)$.
3. S_j stores X_{S_j} as a secret key.

3.1.2. Login Phase and Verification Phase

1. $U_i \rightarrow GW: \{DID_i, M_{U_i, G}, C_i, T_1\}$. U_i inputs the ID_i and PW_i , and his biometrics BIO_i ; then, the smart card computes:

$$\begin{aligned}
 HPW_i^* &= h(PW_i || H(BIO_i)), \\
 u &= D_i \oplus H(BIO_i), \\
 TID_i &= h(ID_i || u), \\
 HID_i^* &= A_i \oplus h(HPW_i^* || TID_i^*), \\
 E_i &= h(HPW_i^* || HID_i^*).
 \end{aligned}$$

Finally, the card checks $E_i^* \stackrel{?}{=} E_i$. If it is equal, the card computes $DID_i = TID_i \oplus HID_i^*$ and $M_{U_i,G} = h(TID_i || HPW_i^* || HID_i^* || T_1)$, and sends $\{DID_i, M_{U_i,G}, C_i, T_1\}$ to GW. Otherwise, it ends the session.

2. $GW \rightarrow S_j: \{DID_i, M_{G,S_j}, M_j, T_2\}$. GW first checks the freshness of T_1 , then computes:

$$\begin{aligned} TID_i^* &= DID_i \oplus C_i \oplus x, \\ HID_i &= C_i \oplus x \\ HPW_i^* &= HID_i \oplus h(TID_i^* || x), \\ M_{U_i,G}^* &= h(TID_i^* || HPW_i^* || HID_i || T_1), \end{aligned}$$

and further tests $M_{U_i,G}^* \stackrel{?}{=} M_{U_i,G}$. If the condition is not satisfied, GW rejects the request. Otherwise, it computes $X_{S_j} = h(SID_j || x)$, $M_j = R \oplus X_{S_j}$, $SK = f(DID_i, R)$ and $M_{G,S_j} = h(DID_i || SID_j || X_{S_j} || SK || T_2)$, and sends $\{DID_i, M_{G,S_j}, M_j, T_2\}$ to S_j .

3. $S_j \rightarrow GW: \{M_{S_j,G}, T_3\}$. S_j first checks T_2 , and computes:

$$\begin{aligned} R^* &= M_j \oplus X_{S_j}, \\ SK^* &= f(DID_i, R^*), \\ M_{G,S_j}^* &= h(DID_i || SID_j || X_{S_j} || SK^* || T_2). \end{aligned}$$

If $M_{G,S_j}^* = M_{G,S_j}$, S_j further computes $k_j = h(X_{S_j} || T_3)$, $M_{S_j,G} = h(k_j || X_{S_j} || SK^* || T_3)$, and sends $\{M_{S_j,G}, T_3\}$ to the GW. Otherwise, it exits the session.

4. $GW \rightarrow U_i: \{k_i, M_{G,U_i}, T_4\}$. GW first checks T_3 , and computes:

$$\begin{aligned} k_j^* &= h(X_{S_j} || T_3), \\ M_{S_j,G}^* &= h(k_j^* || X_{S_j} || SK || T_3). \end{aligned}$$

If $M_{S_j,G}^* = M_{S_j,G}$, GW further computes $k_i = R \oplus h(TID_i^* || x)$, $M_{G,U_i} = h(SK || k_i || T_4)$, and sends $\{k_i, M_{G,U_i}, T_4\}$ to the GW. Otherwise, it exits the session.

5. U_i first checks T_4 , and computes $R^* = k_i \oplus HPW_i \oplus HID_i^*$, $SK^* = f(DID_i, R^*)$ and $M_{G,U_i}^* = h(SK^* || k_i || T_4)$. If $M_{G,U_i}^* = M_{G,U_i}$, U_i believes the legitimacy of GW and the authentication phase ends successfully. Otherwise, the authentication fails.

3.2. Security Flaws in Jung et al.'s Scheme

Jung et al. [25] criticized that Chang et al.'s scheme [24] fails to resist against offline password guessing attack and the session key attack. Thus, they add a new factor to enhance the security of the previous two-factor scheme, and formed a three-factor one. Despite armed with the biometrics factor and provable security proof, their scheme suffers from the same (even more serious) security issues.

3.2.1. Offline Dictionary Attack

Offline dictionary attack is exactly what most schemes suffer from and also the major security requirement of a user authentication protocol. Jung et al. [25] showed that Chang et al.'s scheme [24] cannot resist against this attack once the adversary breaches the victim's smart card and eavesdrops on the message from the open channel. Unfortunately, as we show below, the same attack also works for Jung et al.'s own scheme. In addition, Jung et al.'s scheme is vulnerable to other kinds of offline dictionary attacks with less attack cost.

According to the adversary capabilities mentioned in Section 2.3, it is natural to suppose that the adversary \mathcal{A} somehow possessed U_i 's smart card and then revealed the message $\{A_i, E_i, C_i, D_i\}$ in it; acquired U_i 's biometric BIO_i by a malicious terminal or other ways; and intercepted transcripts $\{DID_i, M_{U_i,G}, C_i, T_1\}$ via the public channel. Then, \mathcal{A} can obtain U_i 's password PW_i as follows:

1. Guesses the value of PW_i to be PW_i^* and ID_i to be ID_i^* from the dictionary space \mathcal{D}_{pw} and \mathcal{D}_{id} , respectively. In fact, according to Wang et al. [28], once an adversary picks the victim's (U_i) smart card, it is easy to learn the corresponding identity ID_i of the user U_i .

2. Computes $HPW_i^* = h(PW_i^* || H(BIO_i))$.
3. Computes $u = D_i \oplus H(BIO_i)$, where D_i is from the smart card.
4. Computes $TID_i^* = h(ID_i^* || u)$.
5. Computes $HID_i^* = A_i \oplus h(HPW_i^* || TID_i^*)$, where A_i is from the card.
6. Computes $M_{U_i,G}^* = h(TID_i^* || HPW_i^* || HID_i^* || T_1)$, where T_1 is from the public channel.
7. Verifies the correctness of PW_i^* and ID_i^* by checking if the computed $M_{U_i,G}^*$ is equal to the intercepted $M_{U_i,G}$.
8. Repeats Steps 1–7 of this procedure until the correct value of PW_i and ID_i is found.

The time complexity of the above attack is $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (5T_H))$. T_H is the running time for hash computation. $|\mathcal{D}_{pw}|$ denotes the number of passwords in \mathcal{D}_{pw} . $|\mathcal{D}_{pw}|$ and $|\mathcal{D}_{id}|$ are very limited, generally $|\mathcal{D}_{id}| < |\mathcal{D}_{pw}| < 10^6$ [30,37], so the above attack is quite efficient.

Besides the above kind of offline dictionary attack, Jung et al.'s scheme still suffers from another kind of offline dictionary attack where the adversary \mathcal{A} obtained the victim's smart card and the biometrics BIO_i . Then, \mathcal{A} can conduct another offline dictionary attack as follows (Steps 1–5 are the same with the above attack, so they are omitted):

1. Step 6. Computes $E_i = h(HPW_i^* || HID_i^*)$, where A_i is from the card.
2. Step 7. Verifies the correctness of PW_i^* and ID_i^* by checking if $E_i^* = E_i$.
3. Step 8. Repeats Steps 1–7 of this procedure until the correct value of PW_i and ID_i is found.

The time complexity of the attack is the same as the former attack. Actually, these two attack strategies are not new, and many researchers [32,36,38–40] have captured these two attack scenarios to break numerous schemes. However, these kinds of adversaries are still rampant.

Remark 1. As we mentioned before, a true three-factor authentication scheme should ensure that even if any two of the three factors are compromised, the other factor cannot be breached and the entire system is still secure. Obviously, this protocol is intrinsically not a three-factor protocol. It indicates that the biometric factor is not a master key to settle the problem in user authentication. On the contrary, a scheme armed with biometrics factor may even cannot provide the same security level as a two-factor authentication. The way to add more factors into the authentication protocol is not the essential way to design a more secure protocol.

In the scheme of Jung et al. [25], the obstacles to compute the verification value $M_{U_i,G}$ for an adversary \mathcal{A} is the PW_i and the ID_i , so \mathcal{A} can guess the value of the PW_i and the ID_i , then verify the guessed value by comparing the computed $M_{U_i,G}^*$ with the intercepted $M_{U_i,G}$. This is exactly the essential reason for the former kind of offline dictionary attack. Similarly, E_i is also the fuse of the latter kind of attack. However, the function of the two parameters is quite different: the $M_{U_i,G}$ is the key of the GW to authenticate U_i , while the E_i contributes to changing the password locally and detecting incorrect input timely. Therefore, the $M_{U_i,G}$ is indispensable to an authentication protocol, and the E_i conduces to improve the usability of a scheme. Furthermore, the “public-key principle” is necessary to resist the former attack [41]; and a way of “honeywords” + “fuzzy-verifiers” is suggested by Wang et al. [30] to deal with the latter attack.

3.2.2. Impersonation Attack

Suppose an adversary \mathcal{A} was also a legal user U_a , then he could get the secret key x as follows:

1. Computes $u = D_a \oplus H(BIO_a)$, where D_a is from the smart card.
2. Computes $TID_a = h(ID_a || u)$.
3. Computes $HPW_a = h(PW_a || H(BIO_i))$.
4. Computes $HID_a = A_a \oplus h(HPW_a || TID_a)$, where A_a is from the card.
5. Computes $x = C_a \oplus HID_a$, where C_a is from the card.

Obviously, the time complexity of the above attack is $\mathcal{O}(5T_H + 3T_R)$, where T_R is the running time for exclusive-or operation. With the secret x , \mathcal{A} has the same capacity as the GW , thus \mathcal{A} can impersonate as the GW or the S_j ; this indicates that the security of the whole system collapsed.

Actually, not only can an insider legal user carry out such an attack, but also an adversary who has gotten the PW and ID of any users by “offline dictionary attack” can also perform such an attack. The C_i ($C_i = HID_i \oplus x$) is the fundamental reason for such an attack. To a legitimate user who knows the HID_i , the secret key x is actually exposed. Therefore, the only “XOR” operation on x is a risky behavior which is far from enough to protect such a significant parameter.

3.2.3. User Anonymity

User anonymity is of great significance to privacy protection. It requires that the adversary can neither confirm who transmits the messages nor recognize whether the messages come from the same user. In wireless sensor networks, numerous sensor nodes are deployed in an unattended environment. In addition, the information is transmitted in a way of broadcasting. Therefore, user anonymity in WSNs is an essential requirement. However, in Jung et al.’s scheme [25], user-specific parameters DID_i and C_i are transmitted via an open channel. Thus, following DID_i or C_i , the adversary \mathcal{A} identifies the transmitted messages with the DID_i and C_i from a large amount of messages in the open channel, and links them to the user U_i . Then, for the purpose of marketing or even other terrible attempts, the \mathcal{A} can learn the user U_i ’s habits, such as the time to initiate an access request, the kinds of sensor nodes to visit. Therefore, Jung et al.’s scheme fails to achieve user anonymity.

3.2.4. Forward Secrecy

Forward secrecy requires that even if the long-term secret key was exposed, the adversary still cannot compute the previous session key. In other words, when the long-term key is compromised, the protocol cannot promise the security of further communications, but it can guarantee the security of the previous communication. Forward secrecy is the last umbrella of system security, but Jung et al.’s scheme fails to achieve it.

Supposing that an adversary \mathcal{A} got the secret key x and intercepted the parameters DID_i and M_j in the channel, \mathcal{A} could perform an attack to get the previous session key as follows:

1. Computes $X_{s_j} = h(SID_j || x)$.
2. Computes $R = M_j \oplus X_{s_j}$, where M_j is from the open channel.
3. Computes $SK = f(DID_i, R)$, where DID_i is from the open channel.

Remark 2. In this scheme, the session key consists of a fixed parameter DID_i and a random number R from GW . As DID_i is exposed to an open channel, the only challenge in computing the session key is the value of R . On one hand, the sensor node S_j has to know R to form the session key. This means that the S_j is capable of computing R . On the other hand, S_j ’s special or only secret parameter is X_{s_j} , where $X_{s_j} = h(SID_j || x)$. Thus, once acquiring X_{s_j} and the transmitted message in an open channel, anyone can compute the session key. Therefore, when an adversary learns the long-term key x , he/she has the same capability as the S_j . Of course, he/she can compute the correct session key. In fact, it is a more secure way to set up the session key with the security mechanism of challenge-response for the two sides of communication. Anyway, all this corroborates that a protocol without any exponentiation operations conducted on the server side cannot achieve forward secrecy [41].

4. Cryptanalysis of Park et al.’s Scheme

Similar to Jung et al., Park et al. [26] also criticized Chang et al.’s scheme [24], and improved this two-factor scheme into a three-factor one. They claimed their new scheme overcomes the weaknesses in [24], and proved the security of the scheme via BAN logic. Unfortunately, we once again found this scheme also insecure: no resistance to two kinds of offline dictionary attacks and no user anonymity.

4.1. A Brief Review of Park et al.'s Scheme

This section describes Park et al.'s scheme [26] briefly.

4.1.1. Registration Phase

Note that the sensor node registration phase is the same with Jung et al.'s [25], so it is omitted here.

1. $U_i \Rightarrow GW: \{ID_i, HPW_i\}$, where $(R_i, P_i) = Gen(BIO_i)$, $HPW_i = h(PW_i || R_i)$.
2. $GW \Rightarrow U_i$: a smart card containing $\{A_i, B_i, C_i, TID_i, h(\cdot)\}$, where $HID_i = h(ID_i || x)$, $X_{s_i} = h(HID_i || x)$, $A_i = h(HPW_i || X_{s_i}) \oplus HID_i$, $B_i = h(HPW_i || X_{s_i})$, $C_i = X_{s_i} \oplus h(ID_i || HPW_i)$. Furthermore, GW stores (TID_i, TID_i°) in database, and TID_i is a random number, TID_i° is initialized to NULL.
3. U_i inputs P_i into the smart card. Note that, in Park et al.'s scheme [26], this step is not mentioned. But, according to the scheme, this step is necessary. We speculate it is missed.

4.1.2. Login Phase and Verification Phase

1. $U_i \rightarrow GW: \{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$. U_i inputs the ID_i and PW_i , and the biometrics BIO_i , and then the smart card computes:

$$\begin{aligned} R_i^* &= Rep(BIO_i, P_i), \\ HPW_i^* &= h(PW_i || R_i^*), \\ X_{s_i}^* &= C_i \oplus h(ID_i || HPW_i^*), \\ B_i^* &= h(HPW_i^* \oplus X_{s_i}^*). \end{aligned}$$

If $B_i^* == B_i$, the card selects a random number $\alpha \in Z_p^*$, and computes $X_i = \alpha P$, $k_i = h(X_{s_i}^* || T_i)$, $DID_i = h(HPW_i^* || X_{s_i}^*) \oplus k_i$ and $M_{U_i,G} = h(A_i || X_{s_i}^* || X_i || T_i)$, and sends $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$ to GW . Otherwise, it ends the session.

2. $GW \rightarrow S_j: \{DID_i, M_{G,S_j}, X_i, T_G\}$. GW first checks T_i , then gets HID_i and computes:

$$\begin{aligned} X'_{s_i} &= h(HID_i || x), \\ k'_i &= h(X'_{s_i} || T_i). \end{aligned}$$

If $M_{U_i,G} \neq h(h(DID_i \oplus k'_i \oplus HID_i) || X'_{s_i} || X_i || T_i)$, GW rejects the request. Otherwise, it computes $X'_{s_j} = h(SID_j || x)$, $M_{G,S_j} = h(DID_i || SID_j || X'_{s_j} || X_i || T_G)$, and sends $\{DID_i, M_{G,S_j}, X_i, T_G\}$ to S_j .

3. $S_j \rightarrow GW: \{M_{S_j,G}, Y_j, T_j\}$. S_j first checks T_G , and if $M_{G,S_j} \neq h(DID_i || SID_j || X'_{s_j} || X_i || T_G)$, rejects it. Otherwise, S_j chooses $b \in Z_p^*$ and computes:

$$\begin{aligned} Y_j &= \beta P, \\ k_j &= h(X_{s_j} || T_j), \\ Z_i &= M_{G,S_j} \oplus k_j \\ SK_j &= h(DID_i || k_j || \beta X_i), \\ M_{S_j,G} &= h(Z_i || X'_{s_j} || X_i || Y_j || T_j), \end{aligned}$$

and sends $\{M_{S_j,G}, Y_j, T_j\}$ to the GW .

4. $GW \rightarrow U_i: \{e_i, M_{G,U_i}, Y_j, T'_G\}$. GW first checks T_j , and computes $k'_j = h(X'_{s_j} || T_j)$, $Z'_i = M_{G,S_j} \oplus k'_j$, if $M_{S_j,G} == h(Z'_i || X'_{s_j} || X_i || X_j || T_j)$, GW further computes:

$$\begin{aligned} e_i &= k_j \oplus h(k_i), \\ M_{G,U_i} &= h(DID_i || M_{U_i,G} || k'_j || X'_{s_i} || X_i || Y_j || T'_G), \\ TID_{i_{new}} &= h(HID_i || T_i), \end{aligned}$$

and updates (TID_i, TID_i°) as $(TID_{i_{new}}, TID_i)$, then sends $\{e_i, M_{G,U_i}, Y_j, T'_G\}$ to the GW . Otherwise, it exits the session.

5. U_i checks T'_G , and computes $k_j^* = e_i \oplus h(k_i)$, if $M_{G,U_i} == h(DID_i || M_{U_i,G} || k_j^* || X_{S_i}^* || X_i || Y_j || T'_G)$, computes $SK = h(DID_i || k_j^* || \alpha Y_j)$, and updates TID_i as $h(HID_i || T_i)$. Otherwise, it exits the session.

4.2. Security Flaws in Park et al.'s Scheme

Compared with Jung et al. [25], Park et al. [26] deployed an elliptic curve cryptosystem trying to achieve user anonymity and resist against offline dictionary attack. Though Wang et al. [3,41] pointed out that a public key algorithm is necessary to achieve user anonymity and offline dictionary attack, it does not mean that, once the public key algorithm is added, the system will be secure. Deploying the public key algorithm requires some skills, and we will propose a sound scheme as an example to explain such skills in Section 5. In this section, we proved that Park et al.'s scheme suffers from many attacks, including offline dictionary attack and no user anonymity.

4.2.1. Offline Dictionary Attack

Suppose the adversary \mathcal{A} got the message $\{A_i, B_i, C_i, P_i, TID_i\}$ in the card; and also acquired U_i 's biometrics BIO_i in addition to intercepted transcripts $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$. Then, \mathcal{A} conducts an offline dictionary attack as follows:

1. Guesses PW_i to be PW_i^* and ID_i to be ID_i^* .
2. Computes $R_i^* = Rep(BIO_i, P_i)$, where P_i is from the smart card.
3. Computes $HPW_i^* = h(PW_i^* || R_i^*)$.
4. Computes $X_{S_i}^* = C_i \oplus h(ID_i^* || HPW_i^*)$.
5. Computes $M_{U_i,G}^* = h(A_i || X_{S_i}^* || X_i || T_i)$, where A_i is from the card, X_i and T_i are from the channel.
6. Verifies the correctness of PW_i^* and ID_i^* by checking whether $M_{U_i,G}^* == M_{U_i,G}$.
7. Repeats Step 1–7 of this procedure until the correct value of PW_i and ID_i is found.

The time complexity of the above attack is $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (3T_H + T_{RE}))$. T_{RE} is the running time of fuzzy extraction computation. Thus, the above attack is quite efficient.

Similar to the analysis in Section 3.2.1, the adversary can also select B_i as the verification to test the guessed value of PW_i^* and ID_i^* .

4.2.2. User Anonymity

Park et al. [26] attempted to update some parameters to provide user anonymity. However, such a method is not as desirable as they expected. On one hand, the gateway has to update the database in every session, which is efficient; on the other hand, if the adversary \mathcal{A} acquires the verifier table $\{TID_i, TID_i^c, HID_i\}$, and intercepts $\{DID_i, X_i, M_{U_i,G}, T_i, TID_i\}$, then \mathcal{A} can find the TID_i from the verifier table and get the corresponding HID_i . Now, \mathcal{A} can compute the $TID_{i_{new}}$ in the next session as $h(HID_i || T_i)$. Thus, \mathcal{A} can link the login request to the same user who has ever used TID_i via computing $TID_{i_{new}}$ in every session. Thus, Park et al.'s scheme cannot achieve user anonymity.

5. Proposed Scheme

In this section, we proposed a new enhanced scheme (as shown in Figure 2) which not only provides some desirable attributes but also can resist against the known attacks. Furthermore, we improve the scheme from the following aspects:

1. based on Wang et al. [3,41], we apply a public key algorithm for resisting against offline dictionary attack via the verification from the open channel. In such an attack, as we analyzed above, the key solution is about the way to construct the verification parameter between the user and the gateway node. Once the verification parameter consists of a "challenge" that is deployed a public key algorithm, a trap door will be built. Therefore only the one who owns the corresponding secret key can compute the correct "challenge" (i.e., X in our scheme). In Park et al.'s scheme, though a public key algorithm is deployed, it is not used to construct a "challenge" for authentication. More

specifically, all the parameters in the verification $M_{U_i,G}$ ($=h(A_i||X_{S_i}||X_i||T_i)$) can be computed with the static or open knowledge in the user side and the open channel, so \mathcal{A} can compute all parameters (A_i, X_{S_i}, X_i, T_i) with guessed password and then use $M_{U_i,G}$ to verify the guessed value. While, in our new scheme, a “challenge” X is built. Besides the static or open knowledge on the user side, \mathcal{A} has to know the dynamic α or the long-term key to compute X , and thus fails to conduct such an attack;

- as introduced in Section 3.2.1, we use “honeywords” + “fuzzy-verifiers” to resist against offline dictionary attack via verification from the smart card [30,42];
- we do not protect user anonymity via updating parameters as Park et al., but deploy a dynamic identity technique via a public key algorithm [3].

The details of our scheme is described as follows:

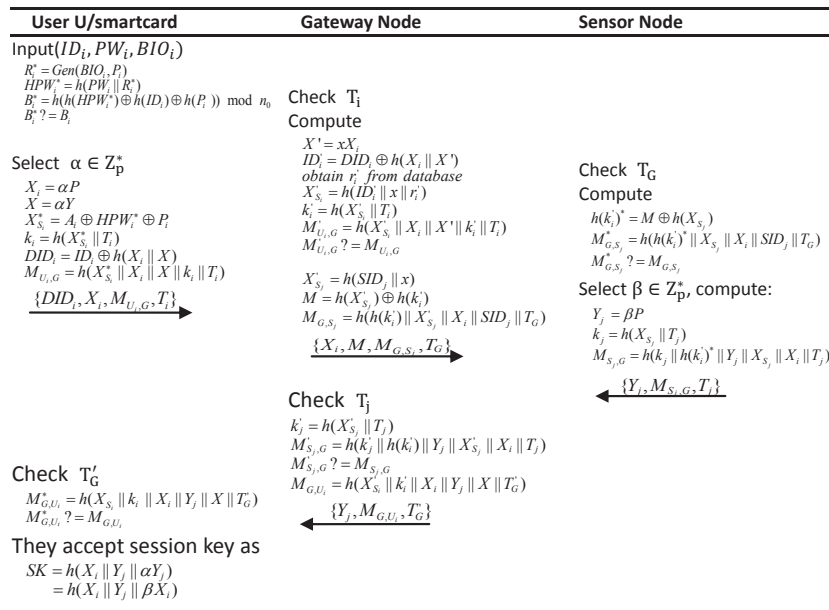


Figure 2. Proposed scheme.

5.1. Registration Phase

The registration phase to the sensor node is similar to Jung et. al. [25] and Park et. al. [26], so it is omitted. When a new user wants to be a legitimate user of the system, then he/she may submit his/her personal information on the gateway to initiate a user registration phase as follows:

- $U_i \Rightarrow GW: \{ID_i, HPW_i\}$, where $(R_i, P_i) = Gen(BIO_i)$, $HPW_i = h(PW_i || R_i)$.
- $GW \Rightarrow U_i$: a smart card containing $\{A_i, B_i, n_0, Y, P\}$, where $X_{S_i} = h(ID_i || x || r_i)$ (r_i is a random number), $A_i = X_{S_i} \oplus HPW_i \oplus P_i$, $B_i = h(h(HPW_i) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0$, $Y = xP$. Furthermore, GW stores $(ID_i, r_i, Honey_List)$ in database, and $Honey_List$ is supposed to count the number of failing in user login phase and it is initialized to NULL. Once its value is bigger than the predetermined threshold, the corresponding smart card will be discarded till the user re-registers.
- U_i inputs P_i into the smart card.

5.2. Login Phase and Verification Phase

After being legitimated, the user U_i can login to the system with the password, identity and biometrics, and get authenticated via exchanging information with the corresponding communication parties. Finally, after finishing the authentication successfully, the user and the sensor node will build a session key to protect the security of the subsequent communications.

1. $U_i \rightarrow GW: \{DID_i, X_i, M_{U_i,G}, T_i\}$. U_i inputs his/her identity ID_i , password PW_i , and biometrics BIO_i ; then, the smart card computes:

$$\begin{aligned} R_i^* &= Rep(BIO_i, P_i), \\ HPW_i^* &= h(PW_i || R_i^*), \\ B_i^* &= h(h(HPW_i^*) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0. \end{aligned}$$

If $B_i^* == B_i$, the card accepts the user, and selects a random number $\alpha \in Z_p^*$, computes:

$$\begin{aligned} X_i &= \alpha P, \\ X &= \alpha Y, \\ X_{s_i}^* &= A_i \oplus HPW_i^* \oplus P_i, \\ k_i &= h(X_{s_i}^* || T_i), \\ DID_i &= ID_i \oplus h(X_i || X), \\ M_{U_i,G} &= h(X_{s_i}^* || X_i || X || k_i || T_i), \end{aligned}$$

and then sends $\{DID_i, X_i, M_{U_i,G}, T_i\}$ as a login request to GW. Otherwise, it ends the session.

2. $GW \rightarrow S_j: \{X_i, M, M_{G,S_j}, T_G\}$. GW first checks the freshness of T_i , computes:

$$\begin{aligned} X' &= xX_i, \\ ID_i' &= DID_i \oplus h(X_i || X'), \end{aligned}$$

and then finds r_i' and $Hony_List$ via ID_i' . If $Hony_List \geq$ the preset value (for example 10), the GW thinks this smart card has been suspended and rejects the request. Otherwise, GW computes $X_{s_i}' = h(ID_i' || x || r_i')$, $k_i' = h(X_{s_i}' || T_i)$. If $M_{U_i,G} \neq h(X_{s_i}' || X_i || X' || k_i' || T_i)$, GW rejects the request and sets $Hony_List = Hony_List + 1$. Once $Hony_List$ is bigger than the preset value, the corresponding smart card is suspended. Otherwise, it computes:

$$\begin{aligned} X_{s_j}' &= h(SID_j || x), \\ M &= h(X_{s_j}' \oplus h(k_i')), \\ M_{G,S_j} &= h(h(k_i') || X_{s_j}' || X_i || SID_j || T_G), \end{aligned}$$

and sends $\{X_i, M, M_{G,S_j}, T_G\}$ to S_j to convey U_i 's request.

3. $S_j \rightarrow GW: \{Y_j, M_{S_j,G}, T_j\}$. S_j first checks the valid of T_G , and computes $h(k_i')^* = M \oplus h(X_{s_j})$. If $M_{G,S_j}^* \neq h(h(k_i')^* || X_{s_j}' || X_i || SID_j || T_G)$, S_j does not believe GW and rejects the session. Otherwise, S_j chooses $\beta \in Z_p^*$ and computes:

$$\begin{aligned} Y_j &= \beta P, \\ k_j &= h(X_{s_j} || T_j), \\ SK_j &= h(X_i || Y_j || \beta X_i), \\ M_{S_j,G} &= h(k_j || h(k_i')^* || Y_j || X_{s_j}' || X_i || T_j), \end{aligned}$$

and sends $\{Y_j, M_{S_j,G}, T_j\}$ to GW.

4. $GW \rightarrow U_i: \{Y_j, M_{G,U_i}, T_G'\}$. GW first checks T_j . Then, it computes $k_j' = h(X_{s_j}' || T_j)$, and if $M_{S_j,G} == h(k_j || h(k_i')^* || Y_j || X_{s_j}' || X_i || T_j)$, GW further computes $M_{G,U_i} = h((X_{s_j}' || k_i' || X_i || Y_j || X || T_G')$, and then sends $\{Y_j, M_{G,U_i}, T_G'\}$ to U_i to transmit S_j 's responds. Otherwise, it exits the session.
5. U_i first checks T_G' , and if $M_{G,U_i} == h((X_{s_j}' || k_i' || X_i || Y_j || X || T_G')$, U_i authenticates the GW, and computes $SK_i = h(X_i || Y_j || \alpha Y_j)$ to finish the authentication successfully. Otherwise, the authentication fails.

5.3. Password Change Phase

Once the user wants to change password for security consideration, he/she can achieve it through the following steps:

1. U_i inputs ID_i , PW_i and new password PW_i^{new} .

2. The card computes:

$$\begin{aligned} R_i^* &= \text{Rep}(BIO_i, P_i), \\ HPW_i^* &= h(PW_i || R_i^*), \\ B_i^* &= h(h(HPW_i^*) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0. \end{aligned}$$

If $B_i^* \neq B_i$, the card does not permit U_i to change the password. Otherwise, it further computes:

$$\begin{aligned} HPW_i^{new} &= h(PW_i^{new} || R_i^*), \\ B_i^{new} &= h(h(HPW_i^{new}) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0, \\ A_i^{new} &= A_i \oplus HPW_i^* \oplus HPW_i^{new}, \end{aligned}$$

and finally replaces A_i, B_i with A_i^{new}, B_i^{new} .

5.4. Revocation Phase

Revocation phase, as the emergency response strategy, is of great significance to the security of the system. It provides an efficient way to protect the account from being abused. When the user finds his/her smart card breached, he/she can revoke the smart card as follows:

1. U_i firstly get authenticated by the card as the way to the step 1 in Section 5.2.
2. $U_i \rightarrow GW: \{DID_i, X_i, M_{U_i,G}, T_i, revoke_request\}$. As described in Section 5.2, the smart card computes $DID_i, X_i, M_{U_i,G}$ and sends $\{DID_i, X_i, M_{U_i,G}, T_i, revoke_request\}$ to the gateway.
3. After receiving the revocation request from U_i , GW first verifies U_i . If GW authenticates U_i successfully, it sets *Honey_List* to a big number, which is bigger than the preset value. Then, the smart card will be revoked, and nobody can login to the system with the card unless U_i re-register. Otherwise, GW rejects the request.

5.5. Re-Register Phase

If a user U_i with correct password and identity is still rejected by S_j , then can re-register as follows:

1. $U_i \Rightarrow GW: \{ID_i, HWR_i, P_i, re - register\}$.
2. Firstly, GW looks for ID_i from *User - list*, checks whether *Honey_List* \geq the preset value. If so, GW believes the card is suspended, then performs the corresponding steps in Section 5.1.

6. Security Analysis

To prove the security of our scheme, we analyze it from two aspects: a formal way using the Burrows–Abadi–Needham (BAN) logic [43]; a informal/heuristic way. Through the formal way, we prove our scheme achieves four basic security goals. These goals ensure that the user and the sensor node are mutual trust, and they both compute the session key successfully; furthermore, the session keys computed by them are equal. Through the informal/heuristic way, we prove that our scheme not only satisfies many desired attributes such as user anonymity and forward security, but also is resistant to various attacks such as offline dictionary attack, impersonation attack, and de-synchronized attack.

6.1. Formal Analysis Based on BAN Logic

The BAN logic [43] is a simple and efficient way to analyze the design logic and security of a protocol. It has a set of particular notions (shown in Table 2) to depict the logic of the protocol. We will prove the security of our scheme according to its notions and processes.

Table 2. Notations in BAN logic.

$P \models X$	P believes X , i.e., the principal P believes the statement X is true.
$P \triangleleft X$	P sees X , i.e., the principal P receives a message that contains X .
$P \mid\Rightarrow X$	P has jurisdiction over X , i.e., the principal P can generate or compute X .
$P \mid\sim X$	P said X , i.e., the principal P has sent a message containing X .
$\#(X)$	X is fresh, i.e., X is sent in a message only at the current run of the protocol, it is usually a timestamp or a random number.
$P \xleftrightarrow{K} Q$	K is the shared key for P and Q .
$P \stackrel{Y}{\equiv} Q$	Y is the secret known only to P and Q or some principals trusted by them.
$\langle X \rangle_Y$	X combined with Y , and Y is usually a secret.
$\{X\}_K$	X encrypted with K .
$\frac{P \models P \xleftrightarrow{K} Q, P \triangleleft \{X\}_K}{P \models Q \mid\sim X}$ or $\frac{P \models P \stackrel{Y}{\equiv} Q, P \triangleleft \langle X \rangle_Y}{P \models Q \mid\sim X}$	RULE(1): the message-meaning rule. This rule will be used in the proving process.
$\frac{P \models \#(X), P \models Q \mid\sim X}{P \models Q \models X}$	RULE(2): the nonce-verification rule. This rule will be used in the proving process.
$\frac{P \models Q \models X, P \models Q \models X}{P \models X}$	RULE(3): the jurisdiction rule. This rule will be used in the proving process.
$\frac{P \models \#(X)}{P \models \#(X, Y)}$	RULE(4): the freshness-conjunction rule. This rule will be used in the proving process.

In BAN logic, the goals of our authentication scheme are defined as:

- Goal 1: $U_i \models S_j \mid\equiv (U_i \xleftrightarrow{SK} S_j)$.
- Goal 2: $U_i \models (U_i \xleftrightarrow{SK} S_j)$.
- Goal 3: $S_j \models U_i \mid\equiv (U_i \xleftrightarrow{SK} S_j)$.
- Goal 4: $S_j \models (U_i \xleftrightarrow{SK} S_j)$.

According to the proof steps in BAN logic, we re-describe our scheme into an idealized form:

- $M_1: U_i \rightarrow GW: \langle X_i, k_i, T_i, DID_i, U_i \xleftrightarrow{X} GW \rangle_{U_i \xleftrightarrow{X_{S_i}} GW}$.
- $M_2: GW \rightarrow S_j: \langle X_i, h(k_i), SID_j, T_G \rangle_{S_j \xleftrightarrow{X_{S_j}} GW}$.
- $M_3: S_j \rightarrow GW: \langle X_j, k_j, h(k_i), T_j \rangle_{S_j \xleftrightarrow{X_{S_j}} GW}$.
- $M_4: GW \rightarrow U_i: \langle X_j, k_i, X, T'_G \rangle_{U_i \xleftrightarrow{X_{S_i}} GW}$.

Then, some assumptions are defined as follows:

- $H_1: U_i \models \#(T'_G)$.
- $H_2: S_j \models \#(T_G)$.
- $H_3: GW \models \#(T_i)$.
- $H_4: GW \models \#(T_j)$.
- $H_5: GW \models S_j \xleftrightarrow{X_{S_j}} GW$.
- $H_6: S_j \models S_j \xleftrightarrow{X_{S_j}} GW$.
- $H_7: GW \models U_i \xleftrightarrow{X_{S_i}} GW$.
- $H_8: U_i \models GW \xleftrightarrow{X_{S_i}} RC$.
- $H_9: U_i \models S_j \mid\Rightarrow U_i \xleftrightarrow{SK} S_j$.

- $H_{10}: S_j \mid \equiv U_i \mid \Rightarrow U_i \xleftrightarrow{SK} S_j$.

Based on the definition above, we perform the BAN logic proof as follows:

From M_1 , it is easy to get $S_1: GW \triangleleft \langle X_i, k_i, T_i, DID_i, U_i \xleftrightarrow{X} GW \rangle_{X_{S_i}}$.

Then, according to $H_7, S_1, RULE(1)$, we get $S_2: GW \mid \equiv U_i \mid \sim \langle X_i, k_i, T_i, DID_i, U_i \xleftrightarrow{X} GW \rangle$.

According to H_3 and $RULE(4)$, we get $S_3: GW \mid \equiv \# \langle X_i, k_i, T_i, DID_i, U_i \xleftrightarrow{X} GW \rangle$.

In addition, according to S_2, S_3 and $RULE(2)$, $S_4: GW \mid \equiv U_i \mid \equiv \langle X_i, k_i, T_i, DID_i, U_i \xleftrightarrow{X} GW \rangle$

From M_2 , it is easy to get $S_5: S_j \triangleleft \langle X_i, h(k_i), SID_j, T_G \rangle_{X_{S_j}}$.

Then, according to $H_7, S_1, RULE(1)$, we get $S_6: S_j \mid \equiv GW \mid \sim \langle X_i, h(k_i), SID_j, T_G \rangle$.

According to H_3 and $RULE(4)$, we get $S_7: S_j \mid \equiv \# \langle X_i, h(k_i), SID_j, T_G \rangle$.

In addition, according to S_2, S_3 and $RULE(2)$, we get $S_8: S_j \mid \equiv GW \mid \equiv \langle X_i, h(k_i), SID_j, T_G \rangle$.

From M_3 , it is easy to get $S_9: GW \triangleleft \langle X_j, k_j, h(k_i), T_j \rangle_{X_{S_j}}$.

Then, according to $H_7, S_1, RULE(1)$, we get $S_{10}: GW \mid \equiv S_j \mid \sim \langle X_j, k_j, h(k_i), T_j \rangle$.

According to H_3 and $RULE(4)$, we get $S_{11}: GW \mid \equiv \# \langle X_j, k_j, h(k_i), T_j \rangle$.

In addition, according to S_2, S_3 and $RULE(2)$, we get $S_{12}: GW \mid \equiv S_j \mid \equiv \langle X_j, k_j, h(k_i), T_j \rangle$.

From M_4 , it is easy to get $S_{13}: U_i \triangleleft \langle X_j, k_i, X, T'_G \rangle_{X_{S_i}}$.

Then, according to $H_7, S_1, RULE(1)$, we get $S_{14}: U_i \mid \equiv GW \mid \sim \langle X_j, k_i, X, T'_G \rangle$.

According to H_3 and $RULE(4)$, we get $S_{15}: U_i \mid \equiv \# \langle X_j, k_i, X, T'_G \rangle$.

In addition, according to S_2, S_3 and $RULE(2)$, we get $S_{16}: U_i \mid \equiv GW \mid \equiv \langle X_j, k_i, X, T'_G \rangle$.

As $SK = h(X_i \parallel X_j \parallel \alpha X_j)$, and combining S_{12}, S_{16} , we get: $S_{17}: U_i \mid \equiv S_j \mid \equiv U_i \xleftrightarrow{SK} S_j$ (**Goal 1**).

Similarly, as $SK = h(X_i \parallel X_j \parallel \beta X_i)$, with S_4, S_8 , we get: $S_{18}: S_j \mid \equiv U_i \mid \equiv U_i \xleftrightarrow{SK} S_j$ (**Goal 3**).

Finally, according to H_2, S_{17} and $RULE(3)$, we get: $S_{19}: U_i \mid \equiv (U_i \xleftrightarrow{SK} S_j)$ (**Goal 2**).

In addition, according to H_{10}, S_{18} and $RULE(3)$, we get: $S_{20}: S_j \mid \equiv (U_i \xleftrightarrow{SK} S_j)$ (**Goal 4**).

Therefore, we prove our scheme achieves Goals 1–4 successfully. In other words, our scheme promises that U_i and S_j have been authenticated mutually, and they further compute and share the same session key SK .

6.2. Informal Analysis

The heuristic way plays an important role in testing the security of the user authentication protocol. It makes up for the defects of formal proofs in some security requirements. For example, the formal proofs cannot capture user anonymity and user friendly problems. Therefore, in this section, we apply the heuristic method to prove the security of our scheme.

6.2.1. Mutual Authentication

In step 2 and step 5 of Section 5.2, the gateway node and the user authenticate each other via their shared secret parameter X_{S_i} and X . On the user side, only with the correct password, biometrics, and the corresponding smart card can U_i compute X_{S_i} , so the gateway can authenticate U_i via this parameter. On the gateway node, after receiving X_i , only the one with the long-term key x , can compute X , so the user authenticate GW via X .

In step 3 and step 4 of Section 5.2, the gateway node and the sensor node authenticate each other via X_{S_j} . If an adversary wants to compute X_{S_j} , then he/she has to guess the long-term key x , and the probability of such an event can be ignored.

Therefore, the user and the sensor node have authenticated the gateway, and the gateway has also authenticated them. Furthermore, from the authentication relationship among the three parties, equivalently, the user and the sensor node get authenticated with each other. All in all, our scheme achieves mutual authentication well.

6.2.2. User Anonymity

In our scheme, ID_i is concealed in DID_i , which is changed with X in every session. To get ID_i , \mathcal{A} has to compute X , which means that \mathcal{A} without α or x has to solve the elliptic curve discrete logarithm problem. As we introduced in Section 2.1, such a problem cannot be solved in polynomial time. Thus, in our scheme, the user identity is not only well protected, but also untraceable.

Furthermore, note that an obvious difference in user anonymity between the wireless sensor network and the distributed network is about whether the user identity can be known by other participants. In a distributed network, there are only two participants: the user and the server. In such a condition, the user identity can be known by the server to build a session key. While in the wireless sensor network, there are three participants: the user, the gateway node and the sensor node. The gateway node acts as a register center and is protected well, so it can know the user identity. While the sensor node is usually deployed in an unattended environment, it is of high possibility to be controlled by the adversary. Thus, the user identity should not be exposed to it. In addition, our scheme achieves such a goal: the user identity is not transmitted to the sensor node.

6.2.3. Forward Secrecy

The session key $SK = h(X_i || Y_j || \beta X_i) = h(X_i || Y_j || \alpha Y_j)$. The key parameter is βX_i or αY_j . If an adversary \mathcal{A} intercepts the message in an open channel, acquires the secret key x , then \mathcal{A} knows X_i and Y_j . Thus, \mathcal{A} needs to compute βX_i or αY_j . However, computing βX_i or αY_j for \mathcal{A} is equivalent to solving the Elliptic curve discrete logarithm problem, and it is bound to fail. Therefore, \mathcal{A} cannot compute SK , and our scheme achieves forward secrecy.

6.2.4. Offline Dictionary Attack

A sound three-factor user authentication scheme should ensure that even if \mathcal{A} gets any two of the three factors, he/she cannot break the system. In our scheme, if \mathcal{A} gets the password and biometrics, he/she still cannot compute X_{s_i} to construct a valid login request; if \mathcal{A} gets the password and the smart card, he/she can neither compute X_{s_i} nor guess the biometrics, thus also fails to perform an attack; if \mathcal{A} gets the smart card and biometrics, then \mathcal{A} may conduct an offline dictionary attack by using $M_{u_i,G}$ or B_i as the verification parameter to check the correctness of the guessed value.

If \mathcal{A} uses B_i , then he/she may make the offline dictionary attack as follows: guesses ID_i and PW_i to be ID_i^* and PW_i^* , respectively, computes $R_i^* = Rep(BIO_i, P_i)$, $HPW_i^* = h(PW_i^* || R_i^*)$, then verifies ID_i^* and PW_i^* by checking $B_i \stackrel{?}{=} h(h(HPW_i^*) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0$. However, even \mathcal{A} gets a pair of $\{ID_i^*, PW_i^*\}$ such that $B_i = h(h(HPW_i^*) \oplus h(ID_i) \oplus h(P_i)) \bmod n_0$, he/she may not find the correct ID_i and PW_i , for there are $|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| \setminus n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pair (where $n_0 = 2^8$ and $|\mathcal{D}_{pw}| = |\mathcal{D}_{id}| = 2^6$) [30]. Thus, \mathcal{A} then has to test the $\{ID_i^*, PW_i^*\}$ via sending the login request to the gateway node, and once the number of login failures exceeds the preset value, the smart card will be suspended and the attack fails.

If \mathcal{A} uses $M_{u_i,G}$, then he/she can compute HPW_i^* as above, and further compute $X_{s_i}^* = A_i \oplus HPW_i^* \oplus P_i$, $k_i = h(X_{s_i}^* || T_i)$, $DID_i = ID_i^* \oplus h(X_i || X)$. However, \mathcal{A} cannot compute X , as we explained in Section 6.2.2, and thus fails to finish such an attack.

In conclusion, our scheme is resistant to dictionary attack.

6.2.5. Privileged Insider Attack

In our scheme, the user submits $\{ID_i, HPW_i, P_i\}$ to the gateway node. The password is well protected by a long-term number R_i , so GW cannot learn any useful information from it. Therefore, our scheme is secure against privileged insider attack.

6.2.6. Verifier-Stolen Attack

The verifier table stored in GW does not expose sensitive messages; even if an adversary acquires the table, he/she cannot make any attack. Thus, our scheme is resistant to verifier-stolen attack.

6.2.7. Replay Attack

The timestamp is used to prevent replay attack. On the one hand, if \mathcal{A} replays the history message directly, the corresponding communication party will find it via checking the freshness of the timestamp. On the other hand, if \mathcal{A} tries to forge the message in the open channel, such as $\{DID_i, X_i, M_{U_i,G}, T_i\}$, then he/she has to know X_{s_i} . However, to compute X_{s_i} , it is asked that \mathcal{A} has to know x or U_i 's password, biometrics and smart card, which is impossible. Similarly, \mathcal{A} also cannot replay or construct other message flows.

7. Performance Analysis

To better evaluate our scheme, we make a comparison among the related schemes for wireless sensor networks [25,26,29,44]. From Table 3, it is obvious that our scheme is more competitive than other schemes: our scheme achieves all the security requirements while others [25,26,29,44] all have some attributes that fail to satisfy more or less; the computation of our scheme is similar or slightly high to that of other schemes. Furthermore, achieving all the security requirements is more significant to an authentication scheme, and it is not advisable to sacrifice security for efficiency.

Table 3. Performance comparison among relevant schemes in wireless sensor networks.

	Computation Overhead		Communication Cost		The Proposed Evaluation Criteria												
	Login (ms)	Auth. (ms)	Login	Auth.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Amin et al. [44]	$8T_H \approx 0.055$	$25T_H \approx 0.017$	768 bits	1536 bits	√	×	√	×	√	√	√	×	√	√	√	√	√
Jiang et al. [29]	$T_M+6T_H \approx 1.2$	$T_M+19T_H \approx 1.2$	1408 bits	1280 bits	√	×	√	×	√	√	√	×	√	√	√	√	√
Jung et al. [25]	$5T_H \approx 0.0035$	$14T_H \approx 0.097$	512 bits	1024 bits	√	×	√	×	×	√	×	√	√	√	√	×	√
Park et al. [26]	$T_E+6T_H \approx 0.51$	$3T_E+18T_H \approx 1.5$	1536 bits	4096bits	√	×	√	×	×	×	×	×	√	√	√	√	√
Our scheme	$2T_E+8T_H \approx 1.0$	$4T_E+18T_H \approx 2.0$	1408 bits	3968 bits	√	√	√	√	√	√	√	√	√	√	√	√	√

T_M is the time of modular exponentiation operation, T_E is the time of scalar multiplication on elliptic curve, T_H is the time of hash computation, $T_M \gg T_E \gg T_H$ (according to Wang et al. [28], $T_M \approx 1.169$ ms, $T_E \approx 0.508$ ms, $T_H \approx 0.693$ μ s) and the lightweight operation such as "XOR" and "|" can be ignored. Let n_0 be 32-bit long; Let $ID_i, PW_i, h(*)$, output of symmetric encryption, timestamp, random numbers be 128-bit long; Let p, g, y be 1024-bit long. \checkmark means the property is satisfied; \times means the property is not satisfied.

8. Conclusions

In this paper, we first introduced the system architecture of wireless sensor networks. Based on this, we summarized the adversary model and security requirements in such a special environment. Secondly, we identified the security flaws in two recent three-factor authentication schemes, and analyzed the inherent reasons for those flaws. Thirdly, we proposed an enhanced scheme resistant to various attack and with many desirable attributes. Then, we proved the security of our scheme via BAN logic and the heuristic analysis. Furthermore, the comparison with other related schemes showed the great advantage of our scheme. Finally, with the development of technology, Internet of Things and Internet of Vehicles will become more and more integrated into our daily life, and the ensuing security problems will become more and more prominent. Therefore, in the future, we will focus on identifying the security requirements and designing a secure but practical protocol in the authentication of these new application scenarios.

Acknowledgments: The authors thank the anonymous reviewers and the Editor for the constructive comments and generous feedback. This research is supported by the BUPT (Beijing University of Posts and Telecommunications) Excellent Ph.D. Students Foundation; the National Natural Science Foundation of China under Grant No.61401038 and No.61702045; the National High Technology Research and Development Program Foundation of China under Grant No. 2015AA017202; the Guangdong Provincial Science and Technology Department Frontier and Key Technology Innovation Project Foundation under Grant No. 2016B010110002;

and the State Grid Corporation of China Key Technology Innovation Project Foundation under Grant No. SGRXTKJ[2017]265.

Author Contributions: Chenyu Wang analyzed the two recent schemes and proposed the enhanced scheme, and then gave the proof of it; Guoai Xu and Jing Sun improved the expressions of the paper.

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

References

1. Pecori, R.; Veltri, L. 3AKEP: Triple-authenticated key exchange protocol for peer-to-peer VoIP applications. *Comput. Commun.* **2016**, *85*, 28–40.
2. Pecori, R. A comparison analysis of trust-adaptive approaches to deliver signed public keys in P2P systems. In Proceedings of the 7th International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 27–29 July 2015.
3. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Comput. Netw.* **2014**, *73*, 41–57.
4. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of “two-factor user authentication in wireless sensor networks”. *Sensors* **2010**, *10*, 2450–2459.
5. Kumar, P.; Choudhury, J.A.; Sain, M.; Lee, S.G.; Lee, H.J. RUASN: A robust user authentication framework for wireless sensor networks. *Sensors* **2011**, *11*, 5020–5046.
6. Ling, C.; Lee, C.; Yang, C.; Hwang, M. A secure and efficient one-time password authentication scheme for WSN. *Int. J. Netw. Secur.* **2017**, *19*, 177–181.
7. Chen, C.T.; Lee, C.C. A two-factor authentication scheme with anonymity for multi-server environments. *Secur. Commun. Netw.* **2013**, *8*, 1608–1625.
8. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
9. Lee, C.; Li, C.; Chen, S. Two attacks on a two-factor user authentication in wireless sensor networks. *Parallel Process. Lett.* **2011**, *21*, 21–26.
10. Kumar, P.; Gurtov, A.; Ylianttila, M.; Lee, S.; Lee, H. A strong authentication scheme with user privacy for wireless sensor networks. *ETRI J.* **2013**, *35*, 889–899.
11. Sun, D.; Li, J.; Feng, Z.; Cao, Z.; Xu, G. On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. *Pers. Ubiquitous Comput.* **2013**, *17*, 895–905.
12. Fan, R.; He, D.P.X.P.L. An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks. *J. Zhejiang Univ. Sci. C* **2011**, *12*, 550–560.
13. Das, A.K.; Sharma, P.; Chatterjee, S.; Sing, J.K. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1646–1656.
14. Wang, D.; Wang, P. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Netw.* **2014**, *20*, 1–15.
15. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323.
16. Li, C.T.; Weng, C.Y.; Lee, C.C. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. *Sensors* **2013**, *13*, 9589–9603.
17. Li, C.T.; Lee, C.C.; Lee, C.W. An improved tTwo-factor user authentication protocol for wireless sensor networks using elliptic curve cryptography. *Sens. Lett.* **2013**, *11*, 958–965.
18. Hsiu-Lien, Y.; Chen, T.H.; Liu, P.C.; Tai-Hoo, K.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779.
19. Choi, Y.; Lee, D.; Kim, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2014**, *14*, 10081–10106.
20. Shi, W.; Gong, P. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*, 51–59.
21. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2015**, *8*, 1070–1081.
22. Wu, F.; Xu, L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 16–30.

23. He, D.; Kumar, N.; Chilamkurthi, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf. Sci. Int. J.* **2015**, *321*, 263–277.
24. Chang, I.; Lee, T.; Lin, T.; Liu, C. Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors* **2015**, *15*, 29841–29854.
25. Jung, J.; Moon, J.; Lee, D.; Won, D. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. *Sensors* **2017**, *17*, doi:10.3390/s17030644.
26. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123, doi:10.3390/s16122123.
27. Huang, X.; Xiang, Y.; Bertino, E.; Zhou, J.; Xu, L. Robust multi-factor authentication for fragile communications. *IEEE Trans. Depend. Secur. Comput.* **2013**, *11*, 568–581.
28. Wang, D.; He, D.; Wang, P.; Chu, C. Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Trans. Depend. Secur. Comput.* **2015**, *12*, 428–442.
29. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392.
30. Wang, D.; Wang, P. Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Trans. Depend. Secur. Comput.* **2016**, doi:10.1109/TDSC.2016.2605087.
31. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Choo, K.K.R.; Shen, J. Design of a provably secure biometrics-based multi-cloud-server authentication scheme. *Futur. Gener. Comput. Syst.* **2017**, *68*, 320–330.
32. He, D.; Wang, D. Robust biometrics-based authentication scheme for multiserver environment. *IEEE Syst. J.* **2015**, *9*, 816–823.
33. Jiang, Q.; Chen, Z.; Li, B.; Shen, J.; Yang, L.; Ma, J. Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. *J. Ambient Intell. Humaniz. Comput.* **2017**, doi:10.1007/s12652-017-0516-2.
34. Lee, C.C.; Chen, C.T.; Wu, P.H.; Chen, T.Y. Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices. *IET Comput. Digit. Tech.* **2013**, *7*, 48–56.
35. He, D.; Zeadally, S.; Kummar, N.; Wu, W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2052–2064.
36. Wang, C.; Wang, D.; Xu, G.; Guo, Y. A lightweight password-based authentication protocol using smart card. *Int. J. Commun. Syst.* **2017**, doi:10.1002/dac.3336.
37. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791.
38. Kumari, S.; Khan, M.K.; Li, X.; Wu, F. Design of a user anonymous password authentication scheme without smart card. *Int. J. Commun. Syst.* **2016**, *29*, 441–458.
39. Li, X.; Qiu, W.; Zheng, D.; Chen, K.; Li, J. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. Ind. Electron.* **2010**, *57*, 793–800.
40. Li, X.; Xiong, Y.; Ma, J.; Wang, W. An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *J. Netw. Comput. Appl.* **2012**, *35*, 763–769.
41. Ma1, C.; Wang, D.; Zhao, S. Security flaws in two improved remote user authentication schemes using smart cards. *Int. J. Commun. Syst.* **2012**, *27*, 2215–2227.
42. Wang, C.; Xu, G. Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card. *Secur. Commun. Netw.* **2017**, doi:10.1155/2017/1619741.
43. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *IEEE Trans. Comput.* **1990**, *8*, 18–36.
44. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62.

