Efficient Multi-Factor User Authentication Protocol with Forward Secrecy for Real-Time Data Access in WSNs

DING WANG and PING WANG, Peking University
CHENYU WANG, Beijing University of Posts and Telecommunications

It is challenging to design a secure and efficient multi-factor authentication scheme for real-time data access in wireless sensor networks. On the one hand, such real-time applications are generally security critical, and various security goals need to be met. On the other hand, sensor nodes and users' mobile devices are typically of a resource-constrained nature, and expensive cryptographic primitives cannot be used. In this work, we first revisit four foremost multi-factor authentication schemes (i.e., those of Amin et al. (JNCA'18), Srinivas et al. (IEEE TDSC'18), Li et al. (JNCA'18), and Li et al. (IEEE TII'18)) and use them as case studies to reveal the difficulties and challenges in designing a multi-factor authentication scheme for wireless sensor networks correctly. We identify the root causes for their failures in achieving truly multi-factor security and forward secrecy. We further propose a robust multi-factor authentication scheme that makes use of the imbalanced computational nature of the RSA cryptosystem, particularly suitable for scenarios where sensor nodes (but not the user's device) are the main energy bottleneck. Comparison results demonstrate the superiority of our scheme. As far as we know, it is the first two-factor authentication scheme for real-time data access in WSNs that can satisfy all 12 criteria of the state-of-the-art evaluation metric under the harshest adversary model so far.

CCS Concepts: • **Security and privacy → Security services**; **Authentication**; **Multi-factor authentication**; • **Computer systems organization → Sensor networks**;

Additional Key Words and Phrases: Cyber-physical systems, wireless sensor networks, truly multi-factor security, forward secrecy, node capture attack

ACM Reference format:
Ding Wang, Ping Wang, and Chenyu Wang. 2020. Efficient Multi-Factor User Authentication Protocol with Forward Secrecy for Real-Time Data Access in WSNs. *ACM Trans. Cyber-Phys. Syst.* 4, 3, Article 30 (March 2020), 26 pages.
https://doi.org/10.1145/3325130

1 INTRODUCTION

With the advent of the Internet of Things (IoT) era, wireless sensor networks (WSNs) have got rapid development in the past few years. In many time-critical WSN applications (e.g., battlefield monitoring [1] and health-care monitoring [2]), users need to acquire the real-time data directly

This research was supported by the National Natural Science Foundation of China under grants 61802006 and 61672059, and by the China Postdoctoral Science Foundation under grant 2018M640026.
Authors' addresses: D. Wang and P. Wang (corresponding author), Peking University, 5 Yiheyuan Road, Haidian District, Beijing, 100871, China; emails: {wangdingg, pwang}@pku.edu.cn; C. Wang, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Haidian District, Beijing, 100876, China; email: wangchenyu@bupt.edu.cn.

2378-962X/2020/03-ART30 $15.00
https://doi.org/10.1145/3325130

from the target sensor nodes without passing through the gateway node. Thus, it is crucial that such sensitive data and user behavior information are well protected against malicious adversaries, and user authentication constitutes the first line of defense. Generally, there are three basic methods for user authentication: what the user knows (e.g., passwords), what the user has (e.g., hardware tokens), and what the user is (e.g., biometrics). As each method has its own pros and cons, multi-factor authentication protocols that combine multiple methods [3, 4] are promising for security-critical applications.

In 2009, Das [5] designed the first password-based authentication scheme using a smart card to achieve mutual authentication among three participants: user, gateway node, and sensor node. However, this scheme was found to be prone to many attacks, including impersonation attack, insider attack, and offline password guessing attack, as revealed by several researchers [6–8]. Then, in 2011, Fan et al. [9] presented an improved scheme to overcome the identified weaknesses of two schemes with lightweight one-way hash operations. However, shortly after this scheme was proposed, Wang and Wang [10] found it unable to provide user anonymity but provided no solution. In 2012, Das et al. [11] proposed a new scheme that achieves dynamic sensor nodes addition. This scheme considers the practical application issues in WSNs and seems to be a promising scheme for WSNs. Unfortunately, it was shortly found subject to offline password guessing attack. In 2013, Xue et al. [12] introduced a temporal-credential-based scheme for WSNs, hoping to achieve user anonymity and resist various attacks. But this attempt fails again.

In 2018, Amin et al. [13] revealed some security flaws in the scheme of Chang and Le [14], and to mitigate the identified vulnerabilities, they designed an improved scheme via adding the biometric factor. Almost at the same time, both Srinivas et al. [15] and Li et al. [16] pointed out that the scheme of Jiang et al. [17] cannot detect unauthorized login and is inapplicable to IoT environments, designed an improved version with the biometric factor, and claimed that their new schemes are secure against various attacks. At IEEE TII 2018, Li et al. [18] demonstrated that various security drawbacks existed in the elliptic curve–based scheme of Gope and Hwang [19] and further proposed a new scheme. They stated that this scheme is secure against various attacks and has the same performance with the original scheme. In 2019, Gupta et al. [20] demonstrated that the scheme of Amin et al. [21] cannot resist several serious attacks (e.g., offline guessing attack) and further proposed an efficient authentication scheme using only lightweight hash operations.

The past 30 years of research on multi-factor authentication have demonstrated that it is difficult to design a protocol for the traditional client-server architecture correctly (see [3, 22–24]). The past 20 years of investigation have manifested that designing a multi-factor scheme for the multi-server environment is more challenging (see [25–27]), whereas the past 10 years of exploration have proved that the design of a multi-factor scheme for WSNs (Figure 1) can only be harder. Besides the difficulties in traditional networks, the design of multi-factor authentication schemes for WSNs is confronted with two additional difficulties. First, sensor nodes are resource-lightweight devices with extremely constrained computation/energy/storage capability. Second, WSNs are usually deployed for security-critical applications, and the security and privacy goals are demanding (see Table I in Wang et al. [28]): they not only withstand various traditional attacks against generic multi-factor schemes (e.g., impersonation, de-synchronization, replay, and parallel) but also resist new attacks arising in WSNs (e.g., *GWN* impersonation, *GWN* by passing, and sensor node capture).

Motivations. Although researchers have made considerable efforts to design a sound multi-factor authentication protocol for WSNs, so far no one has been successful (see Table IV in Wang et al. [28]). When attacking an existing protocol (or designing a new one), most existing literature (e.g., [13, 21, 29, 30]) mainly lists the attack steps (or the protocol procedure) but rarely explicates why

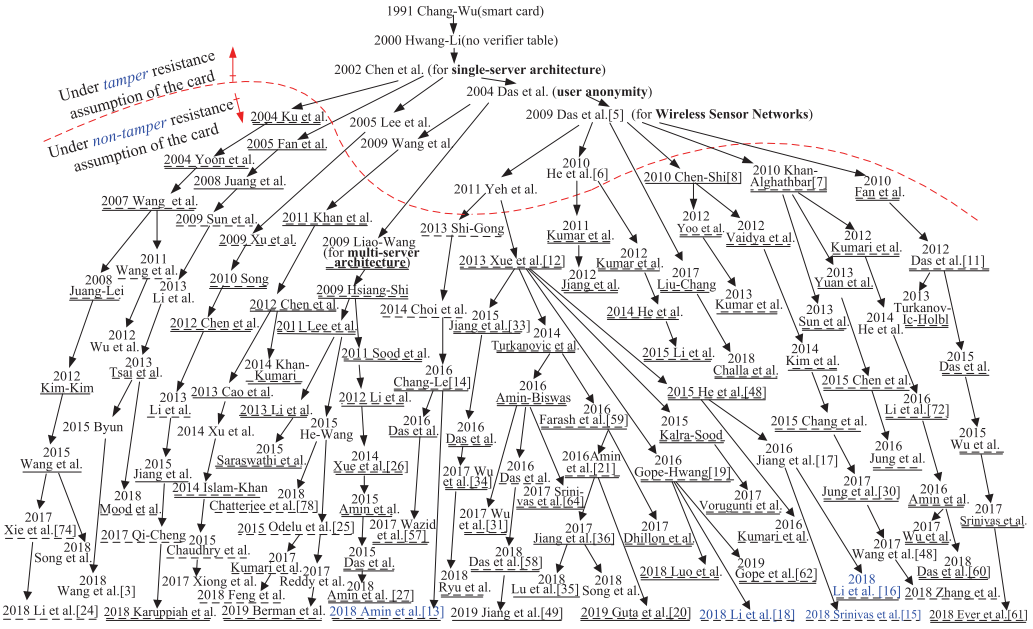

Fig. 1. A brief history of multi-factor authentication for WSNs based on Figure 1 of Wang et al. [28]. Schemes underlined with a solid line cannot achieve *forward secrecy* and schemes underlined with a dashed line fail to provide *truly multi-factor security*, whereas the remaining schemes are problematic to other security or usability issues.

the attack exists and explains what countermeasures can be used to eliminate threats. As such, it is unsurprising to see the vicious "break-fix-break-fix" cycle (see Figure 1) and even dramatically contradictory situations. For instance, several schemes (e.g., [13, 31–33]) try to only use one-way hash functions to achieve robust security, whereas other schemes (e.g., [18, 19, 34–36]) employ much more expensive public key techniques to achieve the same security goals. To mitigate this unsatisfactory situation, two crucial issues should be properly resolved: (1) understanding the inherent reasons for the failures of existing protocols and (2) identifying the real challenges and the corresponding solutions in designing a protocol for WSNs correctly.

Contributions. This work attempts to explore these two issues by revisiting four state-of-the-art multi-factor schemes and proposing a practical multi-factor scheme. In a nutshell, our contributions are summarized as follows:

(1) We investigate four foremost multi-factor authentication schemes for WSNs, namely those of Amin et al. (JNCA'18 [13]), Srinivas et al. (IEEE TDSC'18 [15]), Li et al. (JNCA'18 [16]), and Li et al. (IEEE TII'18 [18])), and find that most of them fail to provide truly multi-factor security, forward secrecy, and other important security goals such as resistance against sensor node capture attack.
(2) We point out the inherent reasons for the three identified primary attacks against multi-factor schemes for WSNs (i.e., smart card loss attack, sensor node capture attack, and forward secrecy breach attack) and summarize the corresponding solutions.
(3) We, for the first time, make use of computational imbalance of the RSA cryptosystem to construct an efficient yet secure multi-factor scheme with forward secrecy for WSNs. The security of our scheme is proved by using BAN logic, and comparison results show that our scheme overall outperforms other related schemes under the widely accepted

Table 1. Notations and Abbreviations

Symbol	Description
U_i	i^{th} user
S_j	j^{th} sensor node
GWN	Gateway node (registration authority)
$\mathcal{A}$	Adversary
x, d_x	Long-term secret key of GWN
y	Private key of GWN in the elliptic curve cryptosystem
ID_i	Identity of U_i
PW_i	Password of U_i
Bio_i	Biometrics of U_i
DID_j	Dynamic identity of U_i
SID_j	Identity of S_j
P_j	Shared secret key between GWN and S_j
$T_1, T_2 \cdots$	Current timestamp
$\oplus$	Bitwise XOR operation
$\|\|$	Concatenation operation
$h(\cdot)$	One-way hash function
$GEN(\cdot), REP(\cdot)$	Fuzzy extractor operation
$\rightarrow$	Open communication channel
$\Rightarrow$	Secure communication channel

adversary model and the most comprehensive evaluation criteria thus far [28]. Particularly, our scheme is suitable for cases where sensor nodes (but not a user's device) are the main energy bottleneck.

The remainder of this article is organized as follows. We revisit the schemes of Amin et al. [13] in Section 2, Srinivas et al. [15] in Section 3, Li et al. [16] in Section 4, and Li et al. [18] in Section 5. In Section 6, we design a new scheme for WSNs and then prove the security of our scheme in Section 7. In Section 8, we compare our scheme with others. We conclude the article in Section 9.

2 REVISITING THE SCHEME OF AMIN ET AL.

In 2018, Amin et al. [13] revealed several security flaws in the "password + smart card" two-factor scheme of Chang and Li [14]. To mitigate the identified vulnerabilities, Amin et al. [13] designed an improved scheme by adding the biometric authentication factor. They eliminated the public key operations in the scheme of Chang and Li [14] and only used a few simple symmetric key operations to save efficiency. Actually, Amin et al. [13] were not the first to make such an attempt, and many recent studies (e.g., [13, 21, 30, 34, 37]) also attempt to increase security by incorporating the biometric factor and improve efficiency by only using simple symmetric key operations. However, we now show that these attempts are all doomed to failure.

2.1 Review of the Scheme of Amin et al.

The scheme of Amin et al. [13] has six phases: registration, login, authentication, post-deployment, password recovery, and change. The first three phases of their scheme are reviewed here, whereas the last three phases are omitted. The notations used in the rest of this article are outlined in Table 1.

2.1.1 Registration Phase. The registration phase is divided into two parts: sensor node registration and user registration. Note that in the scheme of Amin et al. [13], the sensor node registration is finished in the pre-deployment phase.

(1) *Sensor node registration phase*:
Step 1. $S_j \Rightarrow GWN$: $\{SID_j\}$.
Step 2. $GWN \Rightarrow S_j$: $\{P_j\}$, where $P_j = h(SID_j||x)$.
Step 3. S_j keeps $\{P_j\}$.
(2) *User registration phase*:
Step 1. $U_i \Rightarrow GWN$: $\{ID_i, a\ unique\ credential\}$.
Step 2. $GWN \Rightarrow U_i$: smart card with $\{MI_i, f_i\}$. *GWN* first checks the availability of ID_i, then calculates $MI_i = h(ID_i||r_i)$ and $f_i = h(MI_i||x)$, where $r_i \in_R Z_q^*$.
Step 3. The smart card writes $\{C_i, E_i, A_i, \psi_i, REC, REG_i, GEN(), REP(), h(\cdot)\}$ into the card where $(\psi_i, \theta_i) = GEN(Bio_i)$, $A_i = h(ID_i||PW_i||\psi_i)$, $E_i = \theta_i \oplus h(ID_i||PW_i)$, $C_i = f_i \oplus h(PW_i||\psi_i)$, $REC = PW_i \oplus h(ID_i \oplus h(ID_i||\psi_i)$, $REG_i = h(ID_i \oplus \psi_i)$, then detects f_i.

Remark 1. Although Amin et al. [13] do not explain the reason for letting U_i only send ID_i to *GWN*, we speculate that this is due to the consideration of privileged insider attack. They want *GWN* to know as little as possible about U_i's password, but the consequence is that the key parameter f_i of the user will have to be temporarily stored in the smart card. Once f_i is not properly erased or is somehow recovered [38–40], the attacker is likely to perform an impersonation attack. However, if the commonly accepted method is used (i.e., sending the password concealed by a random number to the gateway), the scheme will not suffer from insider attacks and the security risks mentioned previously.

2.1.2 Login and Authentication Phase.

Step 1. $U_i \Rightarrow GWN$: $\{MI_i, N_i, P_i, Q_i, L_i, T_1\}$. The card computes $\theta_i = E_i \oplus h(ID_i'||PW_i')$, $\psi_i' = REP(Bio_i', \theta_i)$, $f_i = C_i \oplus h(PW_i'||\psi_i')$, $A_i^* = h(ID_i'||PW_i'||\psi_i')$. Note that in the scheme of Amin et al. [13], $psi_i' = REP(Bio_i', \theta_i)$, but this value is not used in later steps, so we believe a typo is occurring.
If $A_i^* \neq A_i$, reject U_i. Otherwise, the card computes $N_i = h(MI_i||K_i||f_i||T_1||SID_j)$, $L_i = K_i \oplus h(MI_i||f_i||T_1)$, $P_i = SID_j \oplus h(f_i||T_1)$, $Q_i = h(ID_i) \oplus h(K_i||T_1)$, where $K_i \in_R Z_q^*$.
Step 2. $GWN \Rightarrow S_j$: $\{N_j, SS_j, V_j, T_2\}$. After checking the validity of T_1, *GWN* computes $f_i' = h(MI_i||x)$, $K_i' = L_i \oplus h(MI_i||f_i'||T_1)$, $h(ID_i) = Q_i \oplus h(K_i'||T_1)$, $SID_j' = P_i \oplus h(f_i'||T_1)$, $N_i' = h(MI_i||K_i'||f_i'||T_1||SID_j')$.
If $N_i' \neq N_i$, end the session. Otherwise, *GWN* computes $P_j' = h(SID_j'||x)$, $N_j = h(h(ID_i)||P_j||T_2||K_i)$, $SS_j = h(ID_i) \oplus h(P_j'||T_2)$, $V_j = K_i \oplus h(ID_i)$.
Step 3. $S_j \Rightarrow GWN$: $\{W_j, K_{ij}, T_3\}$. After checking the validity of T_2, S_j computes $h(ID_i) = SS_j \oplus h(P_j||T_2)$, $K_i' = V_j \oplus h(ID_i)$, $N_j' = h(h(ID_i)||P_j||T_2||K_i')$.
If $N_j' \neq N_j$, end the session;.Otherwise, S_j computes $SK_j = h(h(ID_i)||SID_j||K_i'||K_j)$ $(K_j \in_R Z_q^*)$, $W_j = h(SK_j||T_3)$, $K_{ij} = K_i \oplus K_j$.
Step 4. $GWN \Rightarrow U_i$: $\{M_1, K_{ij}, T_4\}$. After checking the validity of T_3, *GWN* computes $K_j' = K_{ij} \oplus K_i$, $SK_{GWN} = h(h(ID_i)||SID_j'||K_i'||K_j')$, $W_j' = h(SK_{GWN}||T_3)$.
If $W_j' \neq W_j$, end the session. Otherwise, *GWN* computes $M_1 = h(SK_{GWN}||K_j'||T_4)$.
Step 5. $U_i \Rightarrow GWN$: $\{M_2\}$. After checking the validity of T_4, U_i computes $K_j' = K_{ij} \oplus K_i'$, $SK_i = h(h(ID_i)||SID_j||K_i||K_j')$, $M_1' = h(SK_i||K_j'||T_4)$.
If $M_1' \neq M_1$, end the session. Otherwise, U_i computes $M_2 = ID_i \oplus h(SK_i||K_i)$.

Step 6. $GWN \Rightarrow U_i$: $\{M_3, M_4, M_5\}$. GWN computes $ID_i' = M_2 \oplus h(SK_{GWN}||K_i)$, $MI_i' = h(ID_i||r_i')$, $f_i' = h(MI_i'||x)$, $M_3 = MI_i' \oplus h(ID_i)$, $M_4 = f_i' \oplus h(f_i||K_i)$, $M_5 = h(h(ID_i)||M_3||M_4)$, where $r_i' \in_R Z_q^*$ and $r_i' \neq r_i$.

Step 7. U_i calculates $M_5' = h(h(ID_i)||M_3||M_4)$. If M_5' equals M_5, U_i computes $MI_i' = M_3 \oplus h(ID_i)$, $f_i' = M_4 \oplus h(f_i||K_i)$, $C_i = f_i' \oplus h(ID_i||\psi_i)$, and finally replaces $\{MI_i, C_i\}$ with $\{MI_i', C_i^i\}$.

2.2 Cryptanalysis of the Scheme of Amin et al.

Amin et al. [13] proposed six assumptions about the adversary $\mathcal{A}$'s capacities. However, we find it cannot well capture $\mathcal{A}$'s capacities in reality:

(1) Their Assumption 3 states that the adversary cannot guess both the identity and password within polynomial time, whereas in reality, according to Wang et al. [41], user-chosen passwords follow Zipf's law. This means that "the guessing probability for an n-character password" is not $\frac{1}{26^n}$ as described in Amin et al. [13] but is $C' \cdot m^{s'}$, where $s' \in [0.15, 0.30]$ and $C' \in [0.001, 0.1]$ are constant CDF-Zipf regression parameters depending on the underlying password space (which is influenced by the user base and password creation policy) [41] and m is the guess number [42].[1] Furthermore, users' identities are usually not considered as secrets and can be acquired by the adversary easily [45]. Thus, it is more desirable to assume that guessing the password and identity simultaneously is practical, and this more realistic assumption has been followed by most of the recent studies [17, 23, 46, 47].

(2) One of the most unique attacks on WSNs is missing in Amin et al. [13]. Since sensor nodes are usually deployed in unattended or open/hostile environments, $\mathcal{A}$ is very likely to attack them and get information from them [28, 37, 48]. Thus, sensor node capture attack is a very realistic (and special) attack and has received increasing attention. However, Amin et al. [13] overlooked this kind of attack.

As pointed out by Jiang et al. [49], among all assumptions about the adversary's capabilities against two-factor schemes, those of Wang et al. [28] are "the most rigorous and practical." Therefore, we follow their assumptions and slightly adjust them to accomodate three-factor authentication schemes:

(1) $\mathcal{A}$ is able to enumerate all items in the space of identity and password simultaneously and also get a victim's identity when analyzing the scheme's security.
(2) $\mathcal{A}$ is able to control the open channel.
(3) For a two-factor authentication scheme, $\mathcal{A}$ may either acquire the victim's password (e.g., through shoulder-surfing, keylogging [50], or password reuse [43]) or parameters of the card through side-channel attacks such as power analysis and reverse engineering [51–54].
(4) $\mathcal{A}$ is able to know previous session keys.
(5) $\mathcal{A}$ is able to acquire the long-term system secret key when considering forward secrecy.
(6) $\mathcal{A}$ is able to control or obtain information of some sensor nodes.
(7) $\mathcal{A}$ may also be a legitimate user and collude with a curious gateway. Generally, this attack is considered in a multi-gateway environment.

[1] As demonstrated in the state-of-the-art password guessing algorithms [43, 44], the guessing probability for a password has little relevance with its length n but is mainly relevant to how popular the password is. Both user-chosen passwords (see [41]) and the efficacy of artificially generated password guesses (see Figure 13(k) of Wang et al. [43]) follow Zipf's law. For concrete examples of guess probabilities with regard to the guess number m, see Table 4 and Figure 4 of Wang and Wang [42].

(8) For a three-factor authentication scheme, $\mathcal{A}$ is able to break any two of the three factors [4, 49].

2.2.1 No Truly Multi-Factor Security. Multi-factor security means that the breach of two authentication factors will not endanger the remaining factor. It is the most essential goal of any multi-factor authentication scheme. However, the scheme of Amin et al. [13] fails to achieve this goal:

— *The adversary's capability*: Getting two of the three factors (the smart card with $\{C_i, E_i, A_i, \psi_i, REC, REG_i, GEN(\cdot), REP(\cdot)\}$ and the biometrics Bio_i).
— *The attack result*: Obtaining U_i's password.
— *The attack steps*:
Step 1. Guess PW_i to be PW_i', ID_i to be ID_i'. Note that, as mentioned before, $\mathcal{A}$ may also know U_i's identity ID_i or use REC to verify the correctness of the guessed value. Then the following steps show the attacking processes of the $\mathcal{A}$ who does not acquire U_i's identity and tries to offline guess $\{ID_i, PW_i\}$ using A_i.
Step 2. Compute $\theta_i = E_i \oplus h(ID_i'||PW_i')$.
Step 3. Compute $\psi_i' = REP(Bio_i', \theta_i)$.
Step 4. Compute $f_i = C_i \oplus h(PW_i'||\psi_i')$.
Step 5. Compute $A_i^* = h(ID_i'||PW_i'||\psi_i')$.
Step 6. Verify the correctness of PW_i and ID_i by checking if $(A_i^* \stackrel{?}{=} A_i)$.
Step 7. Repeat steps 1 through 6 until the correct values of PW_i and ID_i are found.
— *The time complexity*: $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (3T_H + T_B))$, where T_H is the time of hash function and T_B is the time of fuzzy extractor.

As Amin et al. [13] stated, the calculation capabilities and the power of sensor nodes are limited. Thus, it is necessary to verify U_i's legality before transmitting the access request to the gateway or sensor nodes. To authenticate U_i, some verifier A_i must be kept in the smart card. With A_i, the card can authenticate U_i by checking whether the input password (and identity) is correct. At the same time, $\mathcal{A}$ may also exploit this verifier A_i to check the correctness of her guessed password (and identity). This subtlety of verifier A_i in the authentication research domain was first observed by Wang et al. [45] in 2015. They pointed out that there is an inherent "usability vs. security" conflict between the property of local password change (i.e., C2 in Table 3), timely wrong password detection (i.e., C9 in Table 3), and the goal of resisting against offline password guessing attack.[2] The attacks on the scheme of Srinivas et al. [15] (see Section 3.2.1) and the two schemes of Li et al. [16, 18] (see Section 4.2.1 and Section 5.2.1) exploit the same vulnerability.

Fortunately, Wang and Wang [3] recently propose the combination of "fuzzy verifier" and "honeywords" to effectively address this problem. More specifically, U_i's smart card now stores $\theta_i = E_i \oplus h((ID_i||PW_i) \bmod n)$, where n defines the capacity of the (ID, PW) pair, $2^4 \leq n \leq 2^8$. Therefore, even if $\mathcal{A}$ gets E_i, A_i, she cannot figure out the right (ID, PW) by using the preceding attack, for there will be $\frac{|\mathcal{D}_{id} * \mathcal{D}_{pw}|}{n} \approx 2^{32}$ candidate (ID, PW) pairs that satisfy $A_i^* = A_i$ in step 6. To find the exactly correct (ID, PW) pair, $\mathcal{A}$ shall login with GWN, and the honeywords technique [3] can be used to detect the attack (and confine $\mathcal{A}$'s advantage to a very small value). Our new scheme (see Section 6) will adopt this method via letting the verifier A_i be $h(h(ID_i) \oplus h(PW_i)) \bmod n_0$, where $n_0 \in [2^4, 2^8]$ is an integer.

[2]Note that "offline password guessing attack" is a specific way (and also the most effective way in the literature [28, 45]) to obtain the password factor, and it can make a scheme not attain "truly multi-factor security." However, a scheme with no truly multi-factor security may not necessarily be vulnerable to offline password guessing attack.

Besides the preceding attack, the scheme of Amin et al. [13] suffers from another kind of offline password guessing attack, as follows:

—*The adversary's capability*: Getting two authentication factors (i.e., the smart card and the biometrics); eavesdropping the message $\{MI_i, N_i, P_i, Q_i, L_i, T_1\}$ from the open channel.
—*The attack result*: getting U_i's password.
—*The attack steps*:
Step 1. Guess PW_i to be PW_i', ID_i to be ID_i'.
Step 2. Compute f_i' as as in steps 2 through 4 of the preceding attack.
Step 3. Compute $K_i' = L_i \oplus h(MI_i||f_i'||T_1)$.
Step 4. Compute $SID_j' = P_i \oplus h(f_i'||T_1)$.
Step 5. Compute $N_i' = h(MI_i||K_i'||f_i'||T_1||SID_j')$.
Step 6. Verify the correctness of PW_i and ID_i by checking if $N_i' \stackrel{?}{=} N_i$.
Step 7. Repeat steps 1 through 6 until the correct values of PW_i and ID_i are found.
—*The time complexity*: $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (5T_H + T_B))$.

The inherent reason for both attacks is similar: there is an explicit verifier for $\mathcal{A}$ to check the correctness of the guessed value. A natural way to deal with this attack is applying the public key technique to conceal the verifier (see the public key principle in Ma et al. [55]). As the formulation of the verification parameters in each scheme is generally varied, the corresponding solution will be different [56]. As for the scheme of Amin et al. [13], we can let the verifier N_i contain a random item r_i transmitted by the public key technique. Then with two uncertain items (r_i and $\{ID_i, PW_i\}$) in N_i, $\mathcal{A}$ cannot verify the correctness of the guessed $\{ID_i, PW_i\}$. Actually, several recent schemes [14, 20, 31, 57–62] are subject to exactly the same problem.

Remark 2. In their scheme, Amin et al. [13] attempt to use biometrics to overcome the identified weakness in Chang and Li [14]. With the proliferation of biotechnology (e.g., lots of smart phones are equipped with fingerprint sensors), biometrics are currently considered as a convenient and effective authentication factor. For instance, among the 515 recently proposed multi-factor schemes, 217 of them employ the biometric factor [63]. After adding a biometric to the original scheme, several protocols [13, 30–32, 64, 65] abandon the usage of necessary public key techniques. However, biometrics are not a silver bullet to deal with many security problems in the protocol, although it may be possible to improve the security of the protocol to some extent. If there has been a security flaw in the original protocol, then even if the biometrics are introduced, the scheme is still insecure. In addition, as argued by Feng et al. [66], biometrics cannot be regarded as a secret but rather as a type of user privacy information. Therefore, when designing authentication protocols, scholars should focus on how to use appropriate cryptographic principles and algorithms to harden the security of the protocol rather than simply introduce new authentication factor(s) into the protocol to address security flaws.

2.2.2 No Forward Secrecy. Forward secrecy [67] ensures that even if one protocol participant's long-term key has been leaked, the previously agreed session keys remain secure. This security feature is becoming more and more important, because information systems are becoming more and more complex and it is hardly possible to ensure that the system will not be breached and the long-term key will be free from leakage. This is particularly true for security-critical WSN applications (e.g., health monitoring, battlefield surveillance) when considering the prevailing zero-day attacks (e.g., heartbleed [68] and shellshock [69]). Actually, new security standards such as WiFi WPA3 [70] and TLS 1.3 [71] have included forward secrecy as a feature of key exchange protocols. In Amin et al.'s scheme [13], the adversary who obtains one participant's long-term secret key can compute previous agreed session keys as follows:

— *The adversary's capability*: (1) Eavesdropping $\{MI_i, P_i, T_1\}$, $\{N_j, SS_j, V_j, T_2\}$ and $\{K_{ij}\}$; (2) getting the long-term secret key x.
— *The attack result*: Getting all previous session keys. In this attack, we take the session key between U_i and S_j as an example.
— *The attack steps*:
Step 1. Compute $f_i' = h(MI_i \| x)$.
Step 2. Compute $SID_j' = P_i \oplus h(f_i' || T_1)$.
Step 3. Compute $P_j = h(SID_j' || x)$.
Step 4. Compute $h(ID_i) = SS_j \oplus h(P_j || T_2)$.
Step 5. Compute $K_i' = V_j \oplus h(ID_i)$.
Step 6. Compute $K_j' = K_{ij} \oplus K_i'$.
Step 7. Compute $SK = h(h(ID_i) || SID_j' || K_i' || K_j')$.
— *The time complexity*: $O(6T_H)$.

The preceding attack is due to a violation of the "forward secrecy principle": the public key technique is necessary to attain forward secrecy, and at least two exponential operations (or point multiplications in ECC) are needed at the sensor side. The scheme of Amin et al. [13] does not employ any public key technique at all and thus cannot provide forward secrecy. Actually, several recent schemes [14, 20, 31, 57–62] are subject to exactly the same problem.

2.2.3 *Node Capture Attack.*

— *The adversary's capability*: (1) Eavesdropping $\{N_j, SS_j, V_j, T_2\}$, $\{K_{ij}\}$ and $\{M_2\}$; (2) getting S_j's secret key P_j.
— *The attack result*: Getting all previous session keys of the captured sensor node S_j, then further computing U_i's identity ID_i to break user anonymity.
— *The attack steps*:
Step 1. Compute $h(ID_i) = SS_j \oplus h(P_j || T_2)$.
Step 2. Compute $K_i' = V_j \oplus h(ID_i)$.
Step 3. Compute $K_j' = K_{ij} \oplus K_i'$.
Step 4. Compute $SK = h(h(ID_i) || SID_j || K_i' || K_j')$.
Step 5. Compute $ID_i = M_2 \oplus h(SK || K_i)$.
— *The time complexity*: $O(4T_H)$.

Note that this attack may lead to some other severe security flaws. After getting the secret of S_j, $\mathcal{A}$ first computes S_j's session key with U_i, then exploits the vulnerability of M_2 to compute the victim's ID_i, thereby violating the user anonymity requirement [28]. In addition, $\mathcal{A}$ with ID_i can continue to perform the de-synchronization attack as follows:

Step 1. Intercept $\{M_3, M_4, M_5\}$.
Step 2. Generate an arbitrary value M_{3a} with the same length of M_3. Similarly, get M_{4a}.
Step 3. Compute $M_{5a} = h(h(ID_i) || M_{3a} || M_{4a})$.
Step 4. Send $\{M_{3a}, M_{4a}, M_{5a}\}$ to U_i.

In the preceding attack, when U_i receives $\{M_{3a}, M_{4a}, M_{5a}\}$, she computes $M_5' = h(h(ID_i) || M_{3a} || M_{4a})$. As M_5' equals M_{5a}, U_i computes $MI_{ia}' = M_{3a} \oplus h(ID_i)$, $f_{ia}' = M_{4a} \oplus h(f_i || K_i) \neq h(MI_{ia}' || x)$, $C_{ia} = f_{ia}' \oplus h(ID_i || \psi_i)$, and finally replaces $\{MI_i, C_i\}$ with $\{MI_{ia}', C_{ia}^i\}$. Then, the parameter f_{ia}' computed by U_i does not equal to f_{ia} computed by *GWN*. Then, even the legitimate user cannot login.

As revealed in Wang et al. [45], the synchronization mechanism–based method to achieve user anonymity will always raise some other security problems or increase the complexity of the protocol. Therefore, it is recommended to use some public key technique(s) to realize user anonymity.

3 REVISITING THE SCHEME OF SRINIVAS ET AL.

At IEEE TDSC 2018, Srinivas et al. [15] presented a chaotic map–based authentication scheme to meet the demand on user authentication in industrial IoT (IIoT), and showed its advantage to achieve user anonymity and forward secrecy. In addition, they proved the security of their scheme with the automated AVISP tool and the Real-Or-Random model. Unfortunately, we find that their scheme is vulnerable to the sensor node impersonation attack and offline password guessing attack. The chaotic map cryptographic primitive exploits the property of Chebyshev polynomials, and its security builds on the chaotic map–based discrete logarithm problem (DLP). It is relatively more efficient than the traditional finite-field (or elliptic curve) DLP operations.

3.1 Review of the Scheme of Srinivas et al.

3.1.1 Registration. For the sensor S_j, the gateway computes $P_j = h(SID_j \| x)$ and $Key_j = P_j \oplus x$, and delivers $\langle SID_j, P_j \rangle$ in S_j, then stores $\langle SID_j, Key_j \rangle$ to its own database. To the users, they can register to GWN as follows:

Step 1. $U_i \Rightarrow GWN$: $\{DID_i \oplus m_{i1}, DPW_i \oplus m_{i2}\}$, where $DID_i = h(ID_i \| b_i)$ and $DPW_i = h(ID_i \| PW_i)$ and b_i, m_{i1}, and m_{i2} are random numbers.

Step 2. $GWN \Rightarrow U_i$: Smart card with $\{C_i, h(\cdot)\}$. GWN first checks the availability of DID_i, then stores ID_i in the database and computes $C_i = (DID_i \oplus m_{i1}) \oplus (DPW_i \oplus m_{i2}) \oplus h(x \| h(X_{GWN-U_i}))$, where X_{GWN-U_i} is a unique secret number for U_i selected by GWN.

Step 3. U_i inputs Bio_i, computes: $(\sigma_i, \tau_i) = Gen(Bio_i)$, $L_i = b_i \oplus h(\sigma_i \| PW_i)$, $RB_i = h(ID_i \| \sigma_i \| PW_i)$, $C_i' = (C_i \oplus m_{i1} \oplus m_{i2}) \oplus h(\sigma_i \| ID_i)$, replaces C_i with C_i', then stores $RB_i, L_i, Gen(\cdot), Rep(\cdot), \tau_i$ and the fuzzy extractor threshold parameter t into the card.

3.1.2 Login and Authentication Phase.

Step 1. $U_i \Rightarrow GWN$: $M_1 = \{E_i', DID_i', V_{GWN}, G_i, SID_j', T_1\}$. U_i inputs $\{PW_i, ID_i, Bio_i\}$. The card computes $DPW_i = h(ID_i \| PW_i)$, $\sigma_i^* = Rep(Bio_i', \tau_i)$, $b_i^* = L_i \oplus h(\sigma_i^* \| PW_i)$. If $RB_i == h(ID_i \| \sigma_i^* \| PW_i)$, the card computes $C_i = C_i' \oplus h(\sigma_i^* \| ID_i)$, $DID_i = h(ID_i \| b_i^*)$, $J_i = C_i \oplus DID_i \oplus DPW_i$, $E_i = h(J_i \| h(\sigma_i^* \| PW_i) \| T_1)$, $A_g = T_{r_i}(DID_i \| SID_j \| E_i)$, $G_i = A_g \oplus h(DID_i \| J_i \| T_1)$, $V_{GWN} = h(DID_i \| A_g \| G_i \| SID_j \| T_1)$, $E_i' = E_i \oplus h(DID_i \| J_i \| T_1)$, $DID_i' = DID_i \oplus h(E_i' \| J_i \| T_1)$, and $SID_j' = SID_j \oplus h(DID_i \| T_2)$, then finally sends M_1 to GWN. Otherwise, rejects the request.

Step 2. $GWN \Rightarrow S_j$: $M_2 = \{H_j, V_{SN_j}, SID_j'', E_i'', T_2\}$. After checking T_1, GWN computes $M_i = h(x \| h(X_{GWN-U_i}))$, $DID_i = DID_i' \oplus h(E_i' \| M_i \| T_1)$, $A_g^* = G_i \oplus h(DID_i \| M_i \| T_1)$, $SID_j = SID_j' \oplus h(DID_i \| T_1)$. If $V_{GWN} \neq h(DID_i \| A_g^* \| G_i \| SID_j \| T_1)$, GWN ends the session.

Otherwise, GWN fetches P_j with SID_j, further computes $E_i = E_i' \oplus h(M_i \| DID_i \| T_1)$, $SID_j'' = h(SID_j \| P_j \| T_2) \oplus DID_i, H_j = P_j \oplus A_g^*, VSN_j = h(P_j \| SID_j \| A_g^* \| H_j \| T_2)$, $E_i'' = E_i \oplus h(P_j \| T_2)$, then transmits M_2 to S_j.

Step 3. $S_j \Rightarrow U_i$: $M_3 = \{PU_j, N_j', T_3\}$. S_j checks T_2 and computes $DID_i = h(SID_j \| P_j \| T_2) \oplus SID_j''$, $E_i = E_i'' \oplus h(P_j \| T_2)$, $A_g' = P_j \oplus H_j$. If $V_{SN_j} \neq h(P_j \| SID_j \| A_g' \| H_j \| T_3)$, rejects the session.

Otherwise, S_j selects r_j, computes $N_j = T_{r_j}(DID_i \| SID_j \| E_i)$, $SK_{ij} = h(T_{r_j}(A_g') \bmod p \| DID_i \| T_3)$, $PU_j = h(SK_{ij} \| N_j \| T_3)$, and $N_j' = N_j \oplus h(DID_i \| SID_j \| T_3)$, then sends M_3 to U_i.

Step 4. U_i checks T_3 and computes $N_j = N_j' \oplus h(DID_i \| SID_j \| T_3)$, $SK_{ij}^* = h(T_{r_i}(N_j)$ (mod p) $\| DID_i \| T_3)$. If $PU_j \neq h(SK_{ij}^* \| N_j \| T_3)$, the authentication fails. Otherwise, both U_i and S_j accept the session key ($SK_{ij}^* = SK_{ij}$).

3.2 Cryptanalysis of the Scheme of Srinivas et al.

Srinivas et al. [15] depicted the attack model and summarized the security requirements for IIoT. In their scheme, the most critical goal is multi-factor security. However, we now show that their scheme is not as secure as they claimed: the adversary can break the multi-factor security of their scheme.

3.2.1 *No Truly Multi-Factor Security.*

— *The adversary's capability*: (1) Breaking the smart card with $\{RB_i, L_i, Gen(\cdot), Rep(\cdot), \tau_i, t\}$; (2) getting Bio_i.
— *The attack results*: Getting U_i's password.
— *The attack steps*:
Step 1. Guess PW_i to be PW_i', ID_i to be ID_i'.
Step 2. Compute $\sigma_i^* = Rep(Bio_i', \tau_i)$.
Step 3. Compute $RB_i' = h(ID_i' \| \sigma_i^* \| PW_i')$.
Step 4. Verify the correctness of PW_i and ID_i by checking if $(RB_i' \stackrel{?}{=} RB_i)$.
Step 5. Repeat steps 1 through 6 until the equation holds.
— *The time complexity*: $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (T_H + T_B))$.

Note that the adversary can also exploit DID_i' or V_{GWN} to offline guess the value of the password once she eavesdrops on any one of the previous transcripts (i.e., $\{E_i', DID_i', V_{GWN}, G_i, SID_j', T_1\}$). Since the attack steps are similar to the preceding attack in Section 2.2.1, we omit them here.

As mentioned earlier, the preceding attack arises due to the inherent "usability vs. security" conflict. Wang and Wang [3] proposed the way of combining a "fuzzy verifier" and "honeywords" to effectively settle this conflict. More specifically, U_i's smart card now stores $RB_i = h((ID_i \| \sigma_i \| PW_i)$ mod $n)$, where $2^4 \leq n \leq 2^8$. For more rationales, see Section 2.2.1.

3.2.2 *Sensor Node Impersonation Attack.*

— *The adversary's capability*: (1) Registering as a legitimate user U_m to initiate an access request to S_j; (2) eavesdropping H_j.
— *The attack results*: Computing S_j's secret key P_j, then impersonating S_j.
— *The attack steps*:
Step 1. Initiate an access request $M_1 = \{E_m', DID_m', V_{GWN}, G_m, SID_j', T_1\}$ to GWN according to the protocol.
Step 2. Receive H_j from GWN.
Step 3. Compute $P_j = A_g \oplus H_j)$; note that A_g is known to U_m, where $A_g = T_{r_m}(DID_m || SID_j || E_m)$. Once getting S_j's private key, the adversary has the same capability with S_j and thus can impersonate S_j.
— *The time complexity*: The cost for getting a sensor node's private key is similar to a legitimate user. The cost for impersonating S_j is the same as with a legitimate sensor node.

4 REVISITING THE SCHEME OF LI ET AL.

To be self-contained, we first give a brief introduction of the scheme of Li et al. (IEEE TII'18 [18]) and then describes its pitfalls.

4.1 Review of the Scheme of Li et al.

The scheme of Li et al. [18] is built on a subgroup G of an elliptic curve E with a pair of secret/public keys $\{y, Y\}$, where $Y = yP$ and P is a base point in the elliptic curve.

4.1.1 Registration Phase. The sensor node S_j keeps $\langle SID_j, P_j\rangle$, where $P_j = h(SID_j\|x)$, and the gateway stores S_j's identity SID_j. For the users, they can register to GWN as follows:

Step 1. $U_i \Rightarrow GWN$: $\{ID_i, HPW_i, R_i\}$, where $Gen(Bio_i) = (R_i, P_i)$, $HPW_i = h(PW_i\|r_i)$ and r_i is a random number.

Step 2. $GWN \Rightarrow U_i$: $\{B_1, B_3, Y\}$.
After ensuring the validity of ID_i, GWN computes $B_1 = h(ID_i\|HPW_i\|R_i)$, $B_2 = h(ID_i\|x)$, and $B_3 = h(HPW_i\|R_i) \oplus B_2$.

Step 3. U_i stores $\{P_i, r_i, B_1, B_3, Y, SID_j, Gen(), Rep()\}$ into the mobile device.

4.1.2 Login and Authentication Phase.

Step 1. $U_i \Rightarrow GWN$: $M_1 = \{DID_i, D_1, D_3, D_4\}$.
U_i inputs $\{PW_i, ID_i, Bio_i\}$. The card computes $R_i = Rep(Bio_i', P_i)$, $B_1' = h(ID_i\|h(PW_i\|r_i)\|R_i)$. If $B_1' \stackrel{?}{=} B_1$, continues to calculate $B_2 = B_3 \oplus h(h(PW_i\|r_i)\|R_i)$, $D_1 = aP$, $D_2 = aY$, $DID_i = ID_i \oplus h(D_2)$, $D_3 = SID_j \oplus B_2 \oplus h(D_2)$, and $D_4 = h(B_2\|D_2\|SID_j)$, then finally sends M_1 to GWN. Otherwise, rejects the request.

Step 2. $GWN \Rightarrow S_j$: $M_2 = \{D_1, D_6, D_7\}$.
GWN computes $D_2' = yD_1$, $ID_i' = DID_i \oplus h(D_2')$ and checks ID_i', then continues to compute $B_2' = h(ID_i'\|x)$, $SID_j' = D_3 \oplus B_2' \oplus h(D_2')$, $D_4' = h(B_2'\|D_2'\|SID_j')$. If $D_4' \neq D_4$, ends the session.
Otherwise, GWN selects r_g, computes $D_5 = h(SID_j'\|x)$, $D_6 = D_5 \oplus r_g$, and $D_7 = h(D_1\|\ r_g\|D_5\|SID_j')$, then transmits M_2 to S_j.

Step 3. $S_j \Rightarrow GWN$: $M_3 = \{D_8, D_9, D_{10}\}$.
S_j computes $r_g' = P_j \oplus D_6$, $D_7' = h(D_1\|r_g'\|P_j\|SID_j)$. If $D_7' \neq D_7$, exits the session.
Otherwise, S_j selects b, computes $D_8 = bP$, $SK = h(D_1\|D_8\|bD_1)$, $D_9 = h(P_j\|D_8\|r_j'\|SID_j)$, and $D_{10} = h(SID_j\|SK)$, then sends M_3 to GWN.

Step 4. $GWN \Rightarrow U_i$: $M_4 = \{D_8, D_{10}, D_{11}\}$.
GWN computes $D_9' = h(D_5\|D_8\|r_g\|SID_j')$. If $D_9' \neq D_9$, GWN rejects the session. Otherwise, GWN calculates $D_{11} = h(ID_i'\|D_1\|D_8\|B_2')$ and sends M_4 to U_i.

Step 5. If $D_{11} \neq h(ID_i\|D_1\|D_8\|B_2)$, terminates the session. Otherwise, U_i computes $SK' = h(D_1\ \|D_8\|aD_8)$, $D_{10}' = h(SID_j\|SK')$. If $D_{10}' \stackrel{?}{=} D_{10}$, the authentication finishes successfully and the session key is built. Otherwise, the login request is rejected.

4.2 Cryptanalysis of the Scheme of Li et al.

4.2.1 No Truly Multi-Factor Security. To ensure the security for IIoT, Li et al. [18] were devoted to designing a secure authentication scheme and put forward an ECC-based provably secure scheme with user privacy. Although armed with a formal proof, their scheme still does not achieve truly multi-factor security, as shown next:

— *The adversary's capability*: (1) Breaking the smart card with $\{P_i, r_i, B_1, B_3, Y, SID_j, Gen(), Rep()\}$; (2) eavesdropping $\{DID_i, D_1, D_3, D_4\}$; (3) acquiring U_i's identity ID_i; (4) getting Bio_i.
— *The attack results*: Getting U_i's password.
— *The attack steps*:

Step 1. Guess PW_i to be PW_i'.
Step 2. Compute $R_i = Rep(Bio_i', P_i)$.
Step 3. Compute $B_1' = h(ID_i \| h(PW_i' \| r_i) \| R_i)$. Note that the adversary can use B_1 to verify the correctness of PW_i' here, which is simpler.
Step 4. Compute $B_2' = B_3 \oplus h(h(PW_i' \| r_i) \| R_i)$.
Step 5. Compute $D_2' = DID_i \oplus ID_i$.
Step 6. Compute $SID_j' = D_3 \oplus B_2' \oplus D_2'$.
Step 7. Compute $D_4' = h(B_2' || D_2' || SID_j')$.
Step 8. Verify the correctness of PW_i and ID_i by checking if $(D_4' \stackrel{?}{=} D_4)$.
Step 9. Repeat steps 1 through 8 until the equation holds.

—*The time complexity*: $O(|\mathcal{D}_{pw}| * (4T_H + T_B))$.

4.2.2 Poor Repairability. In the scheme of Li et al. [18], when a user U_i suspects (or realizes) that her smart card might be reverse engineered (see [51–54]) and the secret $B_2 = h(ID_u \| x)$ has been exposed. However, even if U_i has discerned this abnormality and updates her password to a new one by following the password change phase of the protocol, no countermeasures can be employed to prevent $\mathcal{M}$ from expoiting the master secret B_2 to login the sensor nodes. In other words, this scheme cannot be easily repaired [45]. In more detail, since $B_2 = h(ID_i \| x)$ is uniquely defined by U_i's identity ID_i and GWN's long-term private key x, GWN cannot update B_2 for U_i unless either ID_i or x is updated. Nevertheless, because x is generally employed for all legitimate users of the entire system but not only one user U_i, it would be unreasonable and inefficient to change x to recover the security of a single user (i.e. U_i). Furthermore, since ID_i is often bound with U_i in many application systems, it is also irrational to update ID_i to tackle the problem. In a nutshell, the repairability of the scheme of Li et al. [18] poses a realistic issue.

5 REVISITING THE SCHEME OF LI ET AL.

At JNCA 2018, Li et al. [16] pointed out that the scheme of Liang et al. [17] cannot detect unauthorized login and is inapplicable to IoT environments. Therefore, similar to Amin et al. [13], they attempted to increase the security of their protocol by adopting the biometric factor. The difference is that Li et al. [16] did not discard public key techniques to improve efficiency, and their scheme should have been more secure than that of Amin et al. [13]. However, after careful examination, we find that their scheme suffers from the same attacks as that of Amin et al. [13]: it cannot provide truly multi-factor security and forward secrecy, and it is vulnerable to sensor node capture attack.

5.1 Review of the Scheme of Li et al.

This section briefly reviews the scheme of Li et al. [16]. Since the password change phase has little relevance, it is omitted.

5.1.1 Registration. The sensor nodes registration phase is the same as that of Amin et al. [13], so we only describe the user registration phase of Li et al. [16] as follows:

Step 1. $U_i \Rightarrow GWN$: $\{ID_i, RPW_i, Bio_i\}$, where $RPW_i = h(PW_i \| a_i)$ and a_i is a random number.
Step 2. $GWN \Rightarrow U_i$: Smart card with $\{\alpha, \delta, A_i, B_i, X, REP(\cdot)\}$, where $c_i \in C$, and $\alpha = h(c_i)$, $\delta = c_i \oplus Bio_i$, $GEN(c_i, Bio_i) = (\alpha, \delta)$, $A_i = h(ID_i \| RPW_i \| c_i)$, $B_i = h(ID_i \| x) \oplus h(RPW_i \| c_i)$.
Step 3. U_i stores a_i into it.

5.1.2 Login and Authentication.

Step 1. $U_i \Rightarrow GWN$: $\{M_2, M_4, M_5, M_6, M_7\}$.
U_i inputs Bio_i', the smart card computes $c_i' = REP(\delta \oplus Bio_i') = REP(c_i \oplus (Bio_i \oplus Bio_i'))$. If $h(c_i') \stackrel{?}{=} \alpha = h(c_i)$, U_i inputs ID_i and PW_i, and computes $A_i' = h(ID_i \| h(PW_i \| a_i) \| c_i')$. If $A_i' \neq A_i$, reject the request.
Otherwise, the card $M_1 = B_i \oplus h(h(PW_i \| a_i) \| c_i')$, $M_2 = sP$, $M_3 = sX = sxP$, $M_4 = ID_i \oplus M_3$, $M_5 = M_1 \oplus r_i$, $M_6 = h(ID_i \| r_i) \oplus SID_j$, $M_7 = h(M_1 \| SID_j \| M_3 \| r_i)$, where r_i is a random number r_i and $s \in Z_n^*$.

Step 2. $GWN \Rightarrow S_j$: $\{M_8, M_9, M_{10}, M_{11}\}$.
GWN calculates $M_3' = yM_2 = ysP$, $ID_i' = M_4 \oplus M_3'$, $M_1' = h(ID_i' \| x)$, $r_i' = M_5 \oplus M_1'$, $SID_j' = M_6 \oplus h(ID_i' \| r_i')$, $M_7' = h(M_1' \| SID_j' \| M_3' \| r_i')$. If $M_7' \neq M_7$, end the session.
Otherwise, GWN computes$P_j' = h(SID_j' \| x)$, $M_8 = ID_i' \oplus P_j'$, $M_9 = r_g \oplus h(ID_i' \| P_j')$, $M_{10} = r_g \oplus r_i'$ and $M_{11} = h(ID_i' \| SID_j' \| P_j' \| r_i' \| r_g)$, where r_g is a random number.

Step 3. $S_j \Rightarrow GWN$: $\{M_{12}, M_{13}\}$. S_j calculates $ID_i'' = M_8 \oplus P_j$, $r_g' = h(ID_i'' \| P_j) \oplus M_9$, $r_i'' = r_g' \oplus M_{10}$, $M_{11}' = h(ID_i'' \| SID_j \| P_j \| r_i'' \| r_g')$. If $M_{11}' \neq M_{11}$, end the session.
Otherwise, S_j calculates $M_{12} = r_j \oplus P_j$, $SK_j = h(ID_i'' \| SID_j \| r_i'' \| r_g' \| r_j)$, $M_{13} = h(P_j \| SK_j \| r_j)$, where r_j is a random number.

Step 4. $GWN \Rightarrow U_i$: $\{M_{14}, M_{15}, M_{16}\}$.
GWN calculates $r_j' = M_{12} \oplus P_j'$, $Sx = h(ID_i' \| SID_j' \| r_i' \| r_g \| r_j')$, $M_{13}' = h(P_j' \| Sx \| r_j')$. If $M_{13}' \neq M_{13}$, end the session.
Otherwise, GWN calculates $M_{14} = M_1' \oplus r_g$, $M_{15} = r_i' \oplus r_j'$, $M_{16} = h(ID_i' \| Sx \| r_g \| r_j')$.

Step 5. U_i calculates $r_g'' = M_{14} \oplus M_1$, $r_j'' = M_{15} \oplus r_i$, $SK_i = h(ID_i \| SID_j \| r_i \| r_g'' \| r_j'')$, $M_{16}' = h(ID_i \| SK_i \| r_g'' \| r_j'')$. If $M_{16}' \stackrel{?}{=} M_{16}$, the authentication is completed. Otherwise, the authentication fails.

5.2 Cryptanalysis of the Scheme of Li et al.

Although the protocol of Li et al. [16] has made progress in terms of the architecture and fully considers the resource limitations of sensor nodes and the insecurity of WSN environment, their scheme is still subject to offline password guessing attack and sensor node capture attack, and it fails to achieve forward secrecy.

5.2.1 No Truly Multi-Factor Security.

— *The adversary's capability*: Breaking two of the three factors—the smart card with $\{\alpha, \delta, A_i, B_i, X, a_i, REP(\cdot)\}$ and the biometrics Bio_i.
— *The attack results*: Getting U_i's password.
— *The attack steps*:
Step 1. Guess PW_i to be PW_i', ID_i to be ID_i'.
Step 2. Compute $c_i' = REP(\delta \oplus Bio_i') = REP(c_i \oplus (Bio_i \oplus Bio_i'))$.
Step 3. Compute $A_i' = h(ID_i' \| h(PW_i' \| a_i) \| c_i')$.
Step 4. Verify the correctness of PW_i and ID_i by checking if $(A_i' \stackrel{?}{=} A_i)$.
Step 5. Repeat steps 1 through 4 until the correct values of PW_i and ID_i are found.
— *The time complexity*: $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * (2T_H + T_B))$, where T_H is the time of hash-function.

Note that the scheme of Li et al. [16] is resistant to the second kind of offline password guessing attack described in Section 2.2.1, due to the use of the ECC public key technique.

5.2.2 No Forward Secrecy.

— *The adversary's capability*: (1) Obtaining the long-term secret key x; (2) eavesdropping $\{M_8, M_9, M_{10}, M_{11}\}$ and $\{M_{12}, M_{13}\}$.
— *The attack results*: Getting all session keys of the system. Note that we take U_i and S_j as an example to show the attack steps. Furthermore, if $\mathcal{A}$ has to capture the sensor node S_j to get P_j, the he can also conduct the following attacks to get all previous session keys of S_j.
— *The attack steps*:
Step 1. Compute $P_j = h(SID_j||x)$.
Step 2. Compute $ID_i'' = M_8 \oplus P_j$.
Step 3. Compute $r_g' = h(ID_i'' \parallel P_j) \oplus M_9$.
Step 4. Compute $r_i'' = r_g' \oplus M_{10}$.
Step 5. Compute $r_j' = M_{12} \oplus P_j$.
Step 6. Compute $SK = h(ID_i''||SID_j||r_i''||r_g'||r_j)$.
— *The time complexity*: $O(3T_H)$.

According to the preceding attack, besides getting the previously agreed session key, an adversary with P_j (via node capture attack) or a legitimate sensor node can also compute the user's identity, which is not recommended for user privacy. The preceding attack is due to a violation of the forward secrecy principle as discussed in Section 2.2.2.

5.2.3 Node Capture Attack.

— *The adversary's capability*: (1) Breaking a sensor node S_j to get P_j; (2) eavesdropping $\{M_8, M_9, M_{10}, M_{11}\}$ and $\{M_{14}\}$.
— *The attack results*: Computing U_i's key secret parameter M_1, then impersonating U_i.
— *The attack steps*:
Step 1. Compute $ID_i'' = M_8 \oplus P_j$.
Step 2. Compute $r_g' = h(ID_i'' \parallel P_j) \oplus M_9$.
Step 3. Compute $M_1 = M_{14} \oplus r_g'$, with M_1, $\mathcal{A}$ then can impersonate U_i.
Step 4. Select a random number r_i and $s \in Z_n^*$.$M_5 = M_1 \oplus r_i$.
Step 5. Compute $M_{3a} = sX = sxP$.
Step 6. Compute $M_{4a} = ID_i \oplus M_{3a}$.
Step 7. Compute $M_{5a} = M_1 \oplus r_i$.
Step 8. Compute $M_{6a} = h(ID_i \parallel r_i) \oplus SID_j$.
Step 9. Compute $M_{7a} = h(M_1 \parallel SID_j \parallel M_{3a} \parallel r_i)$.
Step 10. Send $\{M_{2a}, M_{4a}, M_{5a}, M_{6a}, M_{7a}\}$ to GWN.
— *The time complexity*: $O(3T_H + T_P)$, T_P is the time of scalar multiplication on the elliptic curve.

When GWN receives $\{M_{2a}, M_{4a}, M_{5a}, M_{6a}, M_{7a}\}$, it computes $M_3' = yM_{2a} = ysP$, $ID_i' = M_{4a} \oplus M_3'$, $M_1' = h(ID_i' \parallel x)$, $r_i' = M_{5a} \oplus M_1'$, $SID_j' = M_{6a} \oplus h(ID_i' \parallel r_i')$, $M_7' = h(M_1' \parallel SID_j' \parallel M_3' \parallel r_i')$. As $M_7' \stackrel{?}{=} M_{7a}$, GWN believes $\mathcal{A}$'s authenticity. Therefore, $\mathcal{A}$ impersonates U_i successfully.

From these attacks, it is obvious that node capture attack brings more challenge to the design of the user authentication scheme for WSNs. In the preceding attack, once $\mathcal{A}$ breaks a sensor node S_j, she cannot only acquire S_j's previous session key as the attack steps in forward secrecy but also impersonate U_i. The inherent reason of such an impersonation attack is the use of "$\oplus$" to conceal M_1. We should pay special attention to the bitwise XOR operation, especially in WSNs.

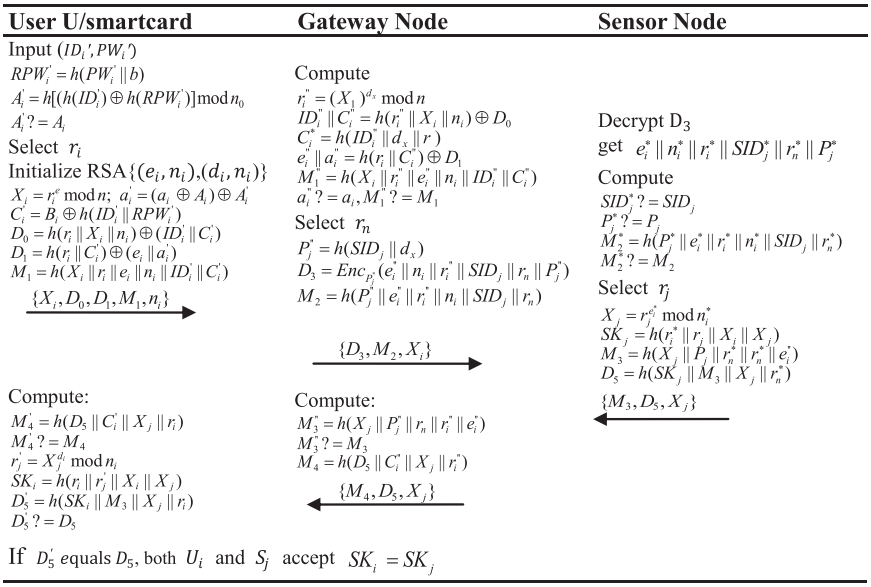

Fig. 2. Login and authentication phase of the proposed scheme.

6 OUR PROPOSED SCHEME

In this section, we first sketch our protocol design ideas and then propose a secure, simple, and efficient user authentication scheme as summarized in Figure 2.

6.1 Design Ideas

Our analysis of these four foremost protocols reveals the following facts. First, achieving truly multi-factor security is still a challenge in the design of a multi-factor authentication protocol. Second, the property of forward secrecy cannot be met by any of the four schemes, largely because of the resource-constrained nature of sensor nodes. Third, generally, the reasons for node capture attack, which is a unique type of attack in WSNs, include two aspects: (1) poor design (e.g., transmitting the secret parameters of the user or sensor node with "XOR" operation over the public channel) and (2) as a result of no forward secrecy, if the adversary $\mathcal{A}$ can breach forward secrecy by using the sensor nodes' long-term key x, then it indicates that *after capturing the sensor node* and obtaining the sensor nodes' x, $\mathcal{A}$ can also compute previously agreed session keys using the same steps as she breaches forward secrecy (see Section 2.2.2 and Section 5.2.2 for two concrete examples). Accordingly, we improve this unsatisfactory situation as follows:

- *Improvements in security*: Under the non-tamper resistance assumption of smart cards, a public key primitive is indispensable for multi-factor authentication schemes to achieve truly multi-factor security [45, 55] and preserve user anonymity [72]. Actually, the public key technique principle was first proposed by Ma et al. [55] as early as 2012 and has been widely adopted in various schemes (e.g., [2, 16, 36, 73, 74]). The key challenge lies in how to integrate a specific public key technique into the protocols for WSNs to meet the security and performance requirements. Consequently, we now show more details on how to employ the public key technique we choose:
  (1) *Employing some kind of public key technique to achieve truly multi-factor security*: In Section 2.2.1, we displayed the pivotal point in deploying a public key algorithm efficiently: add a parameter r_i in the verifier N_i, where r_i is transmitted after being encrypted by a

public key encryption algorithm. More specifically, U_i first chooses a random number r_i to construct a new secret shared parameter between U_i and S_j, and second, to transmit r_i to *GWN* securely, U_i uses *GWN*'s public key to encrypt r_i and gets the ciphertext X_i, then transmits X_i and the verification parameter M_i containing r_i and $\{ID_i, PW_i\}$ to *GWN*. By this way, only *GWN* can compute r_i correctly. To $\mathcal{A}$ who tries to conduct an offline password guessing attack, besides the guessed $\{ID_i', PW_i'\}$, there is still an uncertain r_i; thus, $\mathcal{A}$ is unable to verify the correctness of $\{ID_i', PW_i'\}$ and fails to carry out the offline password guessing attack.

(2) *Employing some kind of public key technique to achieve user anonymity*: Several schemes (e.g., those of Amin et al. [13], Wu et al. [31], and Wu et al. [34]) adopt the synchronization mechanism to achieve user anonymity, but it is not recommended. On the one hand, as Wang et al. [45] revealed, the synchronization mechanism is likely to bring the de-synchronization attack or introduce other security threats. On the other hand, since the public key algorithm is necessary to design a secure authentication scheme, we can directly use public key techniques to achieve user anonymity, it no need to use an additional synchronization mechanism.

(3) *Employing some kind of public key technique to achieve forward secrecy*: The inherent reason for $\mathcal{A}$ to compute the session key is that all items in the session key can be computed by *GWN*. How does this happen? Since U_i and S_j do not have any pre-shared secret, they have to rely on *GWN* to transmit and verify messages sent by each other. In most schemes, *GWN* verifies the items in *SK* directly but not their transformation. Thus, *GWN* can compute *SK*. However, if *GWN* just verifies the transformation of the items in *SK*, then forward secrecy can be achieved. Therefore, we can use the public key algorithm to construct such a transformation of the items in *SK*. We note that some researchers (e.g., Gope et al. [19]) adopt the synchronization mechanism to achieve forward secrecy. As stated earlier, such a synchronization mechanism brings the de-synchronization attack or introduces other security threats (see the attack of Luo et al. [75] on the scheme of Gope et al. [19]).

(4) Following the approach of Wang and Wang [3] of combining a fuzzy verifier and honeywords to settle the conflict between the goals of "detecting typo input" and "resisting against offline password guessing attack."

- *Improvements in efficiency*: As compared to schemes that only involve some symmetric key operations (e.g., Amin et al. [13] and Wu et al. [31]), schemes equipped with a public key primitive will inevitably incur more computational and storage costs. Keeping in mind these necessary increases, we try to optimize the performance of the proposed scheme. As mentioned earlier, the adoption of public key techniques instead of the synchronization mechanism to achieve user anonymity is one of our considerations on efficiency.

  For WSNs, the main restriction of the application of user authentication is the sensor nodes. In other words, the limitations of the sensor nodes itself are the main challenges in designing the authentication protocol: Both their computing capability and their battery power are limited, and it is not convenient/practical to charge the battery often [13, 16]. Therefore, if we want to design a truly practical authentication protocol, we need to fully consider the characteristics of the sensor nodes and reduce their energy consumption.

  Based on the preceding analysis, we prefer the RSA encryption algorithm as our public key technique to achieve truly multi-factor security and user anonymity. In the RSA algorithm, the public key e is recommended to be a small prime (e.g., 17 or $2^{16} + 1$ [76]), so the cryptographic operation $m^e \bmod n$ on the senor node will cost less than the modular

exponentiation operation $g^a \bmod p$ (or aP operations in ECC cryptosystems). To be secure, a needs to be of a large bit length.

6.2 Registration

GWN initializes an RSA algorithm whose public key is (e, n) and private key is (d_x, n), and stores d_x as the system long-term secret key, then chooses a medium integer n_0 ($2^4 \le n_0 \le 2^8$). In our scheme, the sensor node registration is the same as the corresponding part of Section 2.1.1, and the user registration phase is performed as follows:

Step 1. $U_i \Rightarrow GWN$: $\{ID_i, RPW_i\}$. U_i first chooses $\{PW_i, ID_i\}$ and a random number b, then computes $RPW_i = h(PW_i \| b)$, then sends the register request $\{ID_i, RPW_i\}$ to *GWN*.

Step 2. $GWN \Rightarrow U_i$: A smart card with $\{A_i, B_i, n_0, n, e, a_i \oplus A_i, h(\cdot)\}$. *GWN* first tests the availability of ID_i. If *GWN* finds such an identity in the database, then it will ask U_i to choose another ID_i. Otherwise, *GWN* selects two unique random numbers r and a_i for U_i, computes $A_i = h(h(ID_i) \oplus h(RPW_i)) \bmod n_0$, $C_i = h(ID_i \| d_x \| r)$, $B_i = h(ID_i \| RPW_i) \oplus C_i$, then stores $\{ID_i, r, a_i, \texttt{Honey\_List} = \textsc{Null}\}$ into the backend database. Note that `Honey_List` records suspicious login attempts in which the attacker can compute the correct A_i but cannot compute the correct C_i.

Step 3. U_i inputs b into the card.

6.3 Login and Authentication Phase

Step 1. $U_i \Rightarrow GWN$: Login request $\{X_i, D_0, D_1, M_1, n_i\}$. U_i enters $\{ID_i', PW_i'\}$, and the smart card computes $RPW_i' = h(PW_i' \| b)$, $A_i' = h(h(ID_i') \| h(RPW_i')) \bmod n_0$. The card verifies U_i via checking whether $A_i' \stackrel{?}{=} A_i$. If they are not equal, the card rejects the request.

Otherwise, it selects a random number r_i and initializes an RSA algorithm with public key (e_i, n_i) and private key (d_i, n_i), computes $X_i = r_i^e \bmod n$, $C_i' = B_i \oplus h(ID_i' \| RPW_i')$, $D_0 = h(r_i \| X_i \| n_i) \oplus (ID_i' \| C_i')$, $a_i' = (a_i \oplus A_i) \oplus A_i'$, $D_1 = h(r_i \| C_i') \oplus (e_i \| a_i')$, $M_1 = h(X_i \| r_i \| e_i \| \ n_i \| ID_i' \| C_i')$, and finally transmits $\{X_i, D_0, D_1, M_1, n_i\}$ to *GWN*. Note that the RSA initialization process can be pre-computed to save time.

Step 2. $GWN \Rightarrow S_j$: $\{D_3, M_2, X_i\}$. *GWN* first will authenticate U_i as follows: compute $r_i'' = X_i^{d_x} \bmod n$, $ID_i'' \| C_i'' = h(r_i'' \| X_i \| n_i \|) \oplus D_0$, $C_i^* = h(ID_i'' \| d_x \| r)$, $(e_i'' \| a_i'') = h(r_i'' \| C_i'') \oplus D_1$, $M_1'' = h(X_i \| r_i'' \| e_i'' \| n_i \| ID_i'' \| C_i'')$, and retrieve a_i. If $a_i'' \ne a_i$, the session is terminated. Otherwise, check $M_1'' \stackrel{?}{=} M_1$. If $M_1'' \ne M_1$, *GWN* now knows that $a_i'' = a_i$ but $C_i^* \ne C_i''$: there is a $1/n_0$ probability that U_i's card has been compromised. Then, *GWN* either (1) inserts C_i'' into `Honey_List` when the items in `Honey_List` are less than m_0 (e.g., $m_0 = 10$ [3]) or (2) suspends U_i's card (i.e., when $|\texttt{Honey\_List}|{=}m_0$) until U_i re-registers.

If $M_1'' \ne M_1$, *GWN* generates a random number r_n and selects a sensor node S_j to response U_i's request, and computes $P_j'' = h(SID_j \| dx)$, $D_3 = Enc_{P_j''}(e_i'' \| n_i \| r_i'' \| SID_j \| r_n \| P_j'')$, $M_2 = h(P_j'' \| e_i'' \| r_i \| n_i \| SID_j \| r_n)$, then finally sends $\{D_3, M_2, X_i\}$ to S_j.

Step 3. $S_j \Rightarrow GWN$: $\{M_3, D_5, X_j\}$. S_j will first decrypt D_3 with P_j to get $e_i^* \| n_i^* \| r_i^* \| SID_j^* \ \| r_n^* \| P_j^*$, then checks the validity of SID_j^* and P_j^*. If they are not valid, S_j will terminate the session. Otherwise, S_j will continue to calculate $M_2^* = h(P_j \| e_i^* \| r_i^* \| n_i^* \ \| SID_j \| r_n^*)$ and check whether $M_2^* \stackrel{?}{=} M_2$. If they are not equal, the session will be ended.

Otherwise, S_j believes the access request, then selects a random number r_j, computes $X_j = r_j^{e_i^*} \bmod n_i^*$, $SK_j = h(r_i^* || r_j || X_i || X_j)$, $M_3 = h(X_j \| P_j \| r_n^* \| r_i^* \| e_i^*)$, $D_5 = h(SK_j || X_j || r_i^*)$, and finally, S_j sends GWN the message $\{M_3, D_5, X_j\}$.

Step 4. $GWN \Rightarrow U_i$: $\{M_4, D_5, X_j\}$. GWN will first authenticate S_j via comparing the received M_3 to $h(X_j \| P_j'' \| r_n \| r_i'' \| e_i'')$. If the two values are not equal, GWN does not trust S_j, then terminate the session. Otherwise, GWN will forward S_j's response to U_i: compute $M_4'' = h(D_5 || C_i'' || X_j || r_i'')$, then send $\{M_4, D_5, X_j\}$ to U_i.

Step 5. On receiving $\{M_4, D_5, X_j\}$, U_i will first check its authenticity via comparing $h(D_5 || C_i' || X_j || r_i)$ to the received M_4. If the message is not authentic, U_i will terminate the session. Otherwise, U_i authenticates GWN and further computes the session key as $r_j' = X_j^{d_i} \bmod n_i$, $SK_i = h(r_i || r_j' || X_i || X_j)$. If D_5 equals $h(SK_i || X_j || r_i^*)$, U_i authenticates S_j successfully and accepts SK_i. Otherwise, the authentication phase ends in failure.

6.4 Password Change Phase

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

Step 2. The smart card authenticates U_i as in step 1 of Section 6.3. If U_i is authenticated, the password change process will be performed as in step 3; otherwise, the request will be rejected.

Step 3. The smart card computes $RPW_i^{new} = h(PW_i^{new} || b)$, $A_i^{new} = h(h(ID_i) \oplus h(RPW_i^{new})) \bmod n_0$, $B_i^{new} = B_i \oplus h(ID_i || RPW_i) \oplus h(ID_i || RPW_i^{new})$. Note that RPW_i is computed in step 2. Finally, replace $\{A_i, B_i\}$ with $\{A_i^{new}, B_i^{new}\}$.

6.5 Dynamic Node Addition Phase

For a new sensor node, it can join to the networks as follows:

Step 1. S_j sends the node addition request to GWN.

Step 2. GWN will select a unique identity for S_j, compute $P_j = h(SID_j || x)$, and finally send P_j and SID_j to the sensor node S_j.

Step 3. S_j keeps P_j as its secret private key.

7 SECURITY ANALYSIS

This section applies the BAN logic [77] to examine the design logic and security of our scheme. The BAN logic is a famous formal method in the cryptographic, and many schemes adopt this method [16, 36, 47, 48]. Its notions are shown in Table 2. First, we define the goals of our scheme:

- *Goal 1*: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- *Goal 2*: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- *Goal 3*: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- *Goal 4*: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.

Then we transform the message in the channel into an idealized form:

- $\boldsymbol{Mes_1}$: $U_i \rightarrow GWN$: $\langle r_i, ID_i, e_i, n_i, X_i \rangle_{C_i}$.
- $\boldsymbol{Mes_2}$: $GWN \rightarrow S_j$: $\{r_n, r_i, e_i, n_i, X_i\}_{P_j}$.
- $\boldsymbol{Mes_3}$: $S_j \rightarrow GWN$: $\langle r_n, X_j, \{r_i\}_{d_i} \rangle_{P_j}$.
- $\boldsymbol{Mes_4}$: $GWN \rightarrow U_i$: $\langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle_{C_i}$.

Finally, define some assumptions:

- H_1: $U_i \mid\equiv \sharp(r_j)$.
- H_2: $GWN \mid\equiv \sharp(r_i, X_j)$.

Table 2. Notations in BAN Logic

$P \mid\equiv X$	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).
$\sharp(X)$	X is fresh (i.e., X is sent in a message only at the current run of the protocol, and usually it is a timestamp or a random number).
$P \overset{K}{\longleftrightarrow} Q$	K is the shared key for P and Q.
$P \overset{Y}{\rightleftharpoons} Q$	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 usually is a secret.
$\{X\}_K$	X encrypted with K.
$\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$ or $\frac{P \mid\equiv P \overset{Y}{\rightleftharpoons} Q, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$	$RULE(1)$: the message-meaning rule. This rule will be used in the proving process.
$\frac{P \mid\equiv \sharp(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$	$RULE(2)$: the nonce-verification rule. This rule will be used in the proving process.
$\frac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$	$RULE(3)$: the jurisdiction rule. This rule will be used in the proving process.
$\frac{P \mid\equiv \sharp(X)}{P \mid\equiv \sharp(X, Y)}$	$RULE(4)$: the freshness-conjuncatenation rule. This rule will be used in the proving process.

- H_3: $S_j \mid\equiv \sharp(r_n)$.
- H_4: $U_i \mid\equiv U_i \overset{C_i}{\longleftrightarrow} GWN$.
- H_5: $GWN \mid\equiv GWN \overset{C_i}{\longleftrightarrow} U_i$.
- H_6: $GWN \mid\equiv GWN \overset{P_j}{\longleftrightarrow} S_j$.
- H_7: $S_j \mid\equiv S_j \overset{P_j}{\longleftrightarrow} GWN$.
- H_8: $U_i \mid\equiv S_j \mid\Rightarrow r_j, SK$.
- H_9: $S_j \mid\equiv U_i \mid\Rightarrow r_i, SK$.

And now we perform the BAN logic proof:
From *Mes*$_1$, we have S_1: $GWN \triangleleft \langle r_i, ID_i, e_i, n_i, X_i \rangle_{C_i}$
Then according to H_5, S_1, $RULE(1)$, we get S_2: $GWN \mid\equiv U_i \mid\sim \langle r_i, ID_i, e_i, n_i, X_i \rangle$.
According to H_2 and $RULE(4)$, we get S_3: $GWN \mid\equiv \sharp\langle r_i, ID_i, e_i, n_i, X_i \rangle$.
And according to S_2, S_3, and $RULE(2)$, we get S_4: $GWN \mid\equiv U_i \mid\equiv \langle r_i, ID_i, e_i, n_i, X_i \rangle$.
From *Mes*$_2$, we have S_5: $S_j \triangleleft \{r_n, r_i, e_i, n_i, X_i\}_{P_j}$
Then according to H_7, S_5, $RULE(1)$, we get S_6: $S_j \mid\equiv GWN \mid\sim \langle r_n, r_i, e_i, n_i, X_i \rangle$.
According to H_3 and $RULE(4)$, we get S_7: $S_j \mid\equiv \sharp\langle r_n, r_i, e_i, n_i, X_i \rangle$.
And according to S_6, S_7, and $RULE(2)$, we get S_8: $S_j \mid\equiv GWN \mid\equiv \langle r_n, r_i, e_i, n_i, X_i \rangle$.
With S_4 and S_8, we get S_9: $S_j \mid\equiv U_i \mid\equiv \langle r_n, r_i, e_i, n_i, X_i \rangle$.

Since $SK = h(r_i||r_j||X_i||X_j)$, where r_j and X_j are generated by S_j and r_j cannot be acquired by GWN or the adversary, we have S_{10}: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ (goal 3).

According to H_9, S_8, and $RULE(3)$, we get S_{11}: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ (goal 4).

From *Mes*$_3$, we have S_{12}: $GWN \triangleleft \langle r_n, X_j, \{r_i\}_{d_i} \rangle_{P_j}$

Then according to H_6, S_{12}, and $RULE(1)$, we get S_{13}: $GWN \mid\equiv S_j \mid\sim \langle r_n, X_j, \{r_i\}_{d_i} \rangle$.

According to H_2 and $RULE(4)$, we get S_{14}: $GWN \mid\equiv \sharp\langle r_n, X_j, \{r_i\}_{d_i} \rangle$.

And according to S_{13}, S_{14}, and $RULE(2)$, we get S_{15}: $GWN \mid\equiv S_j \mid\equiv \langle r_n, X_j, \{r_i\}_{d_i} \rangle$.

From *Mes*$_4$, we have S_{16}: $U_i \triangleleft \langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle_{C_i}$

Then according to H_4, S_{16}, and $RULE(1)$, we get S_{17}: $S_j \mid\equiv U_i \mid\sim \langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle$.

According to H_1 and $RULE(4)$, we get S_{18}: $S_j \mid\equiv \sharp\langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle$.

And according to S_{17}, S_{18}, and $RULE(2)$, we get S_{19}: $S_j \mid\equiv U_i \mid\equiv \langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle$.

With S_{15} and S_{19}, we get S_{20}: $S_j \mid\equiv U_i \mid\equiv \langle r_i, \{r_i\}_{d_i}, X_i, X_j \rangle$.

Since $SK = h(r_i||r_j||X_i||X_j)$, where r_j can only be computed by U_i with secret key d_i and cannot be acquired by GWN or the adversary, we have S_{21}: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ (goal 3).

According to H_9, S_8, and $RULE(3)$, we get S_{22}: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ (goal 4).

Until now, we have demonstrated that our scheme achieves goals 1 through 4. This means that (1) U_i and S_j have been authenticated mutually, and (2) they negotiate the same session key SK securely.

8 PERFORMANCE EVALUATION

In recent years, many researchers have proposed various evaluation criteria for authentication protocols in WSNs [16, 30, 32, 37, 47]. However, these evaluation metrics are either too abstract, such as "the integrity requirement" proposed by Ali et al. [32], or cannot be practically measured, such as "the efficient use of sensor nodes" proposed by Kumari and Om [37]. Furthermore, they are not comprehensive enough and their effectiveness has not been validated by large-scale evaluation. In 2018, Wang et al. [28] proposed a comprehensive evaluation metric that is composed of 12 independent evaluation criteria, and they demonstrated its effectiveness by testing 44 representative schemes for WSNs. Therefore, we adopt this criteria set to assess our scheme and compare it with 12 related schemes [13, 15–18, 30–32, 34, 47, 64, 65].

Before we go into the pros and cons of each scheme, we examine its system (architecture) model. Each system model in Table 3 has the same definition as in Wang et al. [28], in which eight kinds of system models (i.e., a–h) are proposed to cover both the single-gateway and multi-gateway environments (see Figure 3 of Wang et al. [28]). Among the eight models, models a and g are recommended by Wang et al. [28] because other models have their inherent weaknesses. For instance, Model f, which is employed in Srinivas et al. [15], intrinsically prevents from verifying the authenticity of sensor nodes because there is lack of feedback from sensor nodes to GWN (see step 3 of Section 8).

As shown in Table 3, our scheme satisfies all 12 evaluation criteria, whereas others all have some type of security flaw. The best one that is proposed by Wang et al. [47] achieves 11 criteria. As for the computational cost, six schemes (i.e., [13, 30–32, 64, 65]) only involve symmetric key cryptographic techniques (e.g., one-way hash operation or symmetric encryption). Unsurprisingly, they cost much less computational resources than those (see [16, 17, 34, 47]) that employ public key techniques (e.g., public key encryption). However, these symmetric key–based schemes cannot ensure three important security goals: truly multi-factor security, forward secrecy, and resistance against node capture attack.

Among these schemes that employ some public key techniques, the schemes of Jiang et al. [17], Wu et al. [34], Li et al. [18], Srinivas et al. [15], and Li et al. [16] perform worse than our scheme

Table 3. Performance Comparison Among Relevant Schemes for WSNs

Related Protocols	System Model [28]	Computational Cost (ms)			Evaluation Criteria in Wang et al. [28]*										
		User	Gateway	Sensor Node	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Srinivas et al. (2018) [15]	f	T_B+2T_C+15T_H≈1002.624	10T_H≈0.007	2T_C+6T_H≈2.618	√	√	√	×	×	√	√	×	√	√	×	√
Ali et al. (2018) [32]	a	T_B+2T_S+6T_H≈1000.005	5T_S+13T_H≈0.012	T_S+5T_H≈0.004	√	√	√	×	×	√	√	×	√	√	×	×
Wazid et al. (2018) [65]	a	T_B+2T_S+13T_H≈1000.010	4T_S+5T_H≈0.006	2T_S+4T_H≈0.004	√	√	√	×	×	√	√	×	√	√	×	×
Li et al. (2018) [16]	a	T_B+2T_P+8T_H≈1001.022	T_P+9T_H≈0.514	4T_H≈0.003	√	√	√	×	×	×	√	×	√	√	×	×
Li et al. (2018) [18]	a	T_B+3T_P+7T_H≈1001.529	T_P+7T_H≈0.513	2T_P+4T_H≈1.019	√	√	√	×	√	×	√	√	√	√	√	√
Amin et al. (2018) [13]	a	T_B+17T_H≈1000.012	16T_H≈0.011	4T_H≈0.003	√	√	√	×	×	×	√	×	√	√	√	×
Jung et al. (2017) [30]	a	11T_H≈0.008	17T_H≈0.012	6T_H≈0.004	√	√	√	×	×	×	√	×	√	√	×	×
Wu et al. (2017) [34]	a	2T_P+T_S+11T_H≈1.024	2T_S+11T_H≈0.009	2T_P+T_S+4T_H≈1.020	√	√	√	×	×	×	√	√	×	√	×	√
Wang et al. (2017) [47]	a	T_B+3T_P+10T_H≈1001.531	T_P+11T_H≈0.516	2T_P+4T_H≈1.019	√	√	√	√	√	×	√	√	√	√	√	√
Srinivas et al. (2017) [64]	g	T_B+10T_H≈1000.007	13T_H≈0.009	6T_H≈0.004	√	√	√	×	×	√	√	×	√	√	×	×
Wu et al. (2017) [31]	g	8T_H≈0.006	10T_H≈0.007	3T_H≈0.002	√	√	√	×	×	×	√	×	×	√	×	√
Jiang et al. (2016) [17]	a	2T_P+8T_H≈1.022	T_P+8T_H≈0.513	6T_H≈0.004	√	√	√	√	×	×	√	×	×	√	×	×
Our scheme	a	T_{be}+T_{se}+9T_H≈9.514	T_{be}+T_S+6T_H≈9.357	T_{se}+T_S+4T_H≈0.157	√	√	√	√	√	√	√	√	√	√	√	√

T_P denotes scalar multiplication on elliptic curve, T_H denotes hash computation, T_S denotes symmetric encryption/decryption, T_C denotes Chebyshev polynomial operation, T_B denotes the time for fuzzy extracting biometric information. According to Wang et al. [45], we can get that $T_E \approx 1.169$ ms (when we set the length of modular $|n| = 512$), big-exponent modular exponentiation $T_{be} \approx 1.169^*8 = 9.352$ ms, and small-exponent modular exponentiation $T_{se} \approx \frac{17}{1024} * T_{be} = 0.156$ ms (when we set RSA $|n| = 1024$ and $e = 2^{16} + 1$ as recommended), $T_P \approx 0.508$ ms, $T_H \approx 0.693\mu s$, $T_S \approx 0.541\mu s$. According to Chatterjee et al. [78], $T_C \approx 21.02/(8.7/0.541) = 1.307$ ms; according to Dodis et al. [79] and Encinas [80], the fuzzy extractor T_B contains some hash and symmetric encryption/decryption operations, and the most time is spent on reading the biological characteristic, so we roughly estimate $T_B \approx 1$ second according to Deli 3959 fingerprint or facial recognition.
*We adopt the evaluation criteria proposed in Wang et al. [28]: C1 for no password verifier table, C2 for password friendly, C3 for password exposure, C4 for no smart card loss attack, C5 for resistance to know attacks, C6 for sound repairability, C7 for provision of key agreement, C8 for no clock synchronization, C9 for timely typo detection, C10 for mutual authentication, C11 for user anonymity, and C12 for forward secrecy. A check mark (√) means that the property is satisfied; a times sign (×) means that the property is not satisfied.

in terms of security and are more computationally expensive at the sensor node. We note that in our scheme, the gateway's computational cost is at least 18.16 times more expensive than other schemes, and this is a limitation of our scheme: it may be not suitable for a gateway with a large number of sensor nodes. Still, gateways (base stations) are generally connected to the power infrastructure (or can be recharged), and they are powerful.

We emphasize that for a user authentication scheme, security is at least as important as efficiency, and thus it is not advisable to significantly reduce security to increase marginal efficiency. The scheme of Wang et al. [47] is the only protocol that has a security level similar to our scheme, and it is more efficient at the gateway side but much more expensive at the sensor node side. As sensor nodes are generally the energy bottleneck in WSNs, our scheme is more suitable for WSNs.

9 CONCLUSION

With the proliferation of IoT, WSNs are receiving more and more attention, and how to ensure the security of WSNs has become one of the research hotspots. A large number of multi-factor authentication schemes have been proposed recently, yet most of them fail to achieve the claimed security goals. In this article, we have taken four foremost multi-factor authentication schemes as case studies to show how to effectively achieve truly multi-factor security, forward secrecy, and resistance against node capture attack under the resource-limited nature of sensor nodes (and the user's devices). We, for the first time, employ the RSA cryptosystem to design a multi-factor

authentication protocol for WSNs. RSA previously was thought to be unworkable for WSNs, but we successfully exploit it by making use of its computational imbalance at the encryption side and the decryption side. We have used BAN logic to show the security of our protocol. Finally, we compared the proposed protocol to 12 representative schemes. The evaluation results show the superiority of our new scheme. Although the article considers the framework of WSNs, the cryptanalysis results and countermeasures suggested are largely general and helpful for multi-factor authentication research in other environments.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their invaluable comments.

REFERENCES

[1] Iuliu Vasilescu, Keith Kotay, Daniela Rus, Matthew Dunbabin, and Peter Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of EWSN 2005*. 154–165.

[2] Xiaoying Jia, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo. 2019. Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wirel. Netw.* 25, 8 (2019), 4737–4750. DOI: 10.1007/s11276-018-1759-3

[3] D. Wang and P. Wang. 2018. Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Trans. Depend. Secur. Comput.* 15, 4 (2018), 708–722.

[4] Xinyi Huang, Yang Xiang, Ashley Chonka, Jianying Zhou, and Robert H. Deng. 2011. A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Trans. Parall. Distrib. Syst.* 22, 8 (2011), 1390–1397.

[5] Manik Lal Das. 2009. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* 8, 3 (2009), 1086–1090.

[6] Daojing He, Yi Gao, Sammy Chan, Chun Chen, and Jiajun Bu. 2010. An enhanced two-factor user authentication scheme in wireless sensor networks. *Ad Hoc & Sensor Wirel. Netw.* 10, 4 (2010), 361–371.

[7] M. K. Khan and K. Alghathbar. 2010. Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. *Sensors* 10, 3 (2010), 2450–2459.

[8] T. H. Chen and W. K. Shih. 2010. A robust mutual authentication protocol for wireless sensor networks. *ETRI J.* 36, 1 (2010), 316–323.

[9] R. Fan, D. He, X. Pan, and L. Ping. 2011. An efficient and DoS-resistant user authentication scheme for two-tiered wireless sensor networks. *J. Zhejinag Univ. Sci. C* 12, 7 (2011), 550–560.

[10] D. Wang and P. Wang. 2014. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Netw.* 20, 2 (2014), 1–15.

[11] A. K. Das, S. Sharma, P. Chatterjee, and J. K. Sing. 2012. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *J. Netw. Comput. Appl.* 35, 5 (2012), 1646–1656.

[12] K. Xue, C. Ma, P. Hong, and R. Ding. 2013. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* 36, 1 (2013), 316–323.

[13] Ruhul Amin, S. K. Hafizul Islam, Neeraj Kumar, and Kim Kwang Raymond Choo. 2018. An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. *J Netw. Comput. Appl.* 104 (2018), 133–144.

[14] Chin Chen Chang and Hai Duong Le. 2016. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Trans. Wireless Commun.* 15, 1 (2016), 357–366.

[15] Jangirala Srinivas, Ashok Kumar Das, Mohammad Wazid, and Neeraj Kumar. 2018. Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things. *IEEE Trans. Depend. Secur. Comput.* Epub ahead of print. July 19, 2019. DOI: 10.1109/TDSC.2018.2857811

[16] Xiong Li, Jiangwei Niu, Saru Kumari, Fan Wu, Arun Kumar Sangaiah, and Kim Kwang Raymond Choo. 2018. A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments. *J. Netw. Comput. Appl.* 103 (2018), 194–204.

[17] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang. 2016. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *J. Netw. Comput. Appl.* 76 (2016), 37–48.

[18] Xiong Li, Jianwei Niu, Md Zakirul Alam Bhuiyan, Fan Wu, Marimuthu Karuppiah, and Saru Kumari. 2018. A robust ECC based provable secure authentication protocol with privacy protection for industrial Internet of Things. *IEEE Trans. Ind. Inform.* 14, 8 (2018), 3599–3609.

[19] Prosanta Gope and Tzonelih Hwang. 2016. A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Trans. Ind. Electr.* 63, 11 (2016), 7124–7132.
[20] Ankur Gupta, Meenakshi Tripathi, Tabish Jamil Shaikh, and Aakar Sharma. 2019. A lightweight anonymous user authentication and key establishment scheme for wearable devices. *Comput. Netw.* 149 (2019), 29–42.
[21] Ruhul Amin, Sk Hafizul Islam, G. P. Biswas, Muhammad Khurram Khan, Lu Leng, and Neeraj Kumar. 2016. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* 101, C (2016), 42–62.
[22] Xinyi Huang, Xiaofeng Chen, Jin Li, Yang Xiang, and Li Xu. 2014. Further observations on smart-card-based password-authenticated key agreement in distributed systems. *IEEE Trans. Parall. Distrib. Syst.* 25, 7 (2014), 1767–1775.
[23] Ding Wang, Qianchen Gu, Haibo Cheng, and Ping Wang. The request for better measurement: A comparative evaluation of two-factor authentication schemes. In *Proceedings of ASIACCS 2016.* ACM, New York, NY, 475–486.
[24] Xiaowei Li, Dengqi Yang, Xing Zeng, Benhui Chen, and Yuqing Zhang. 2018. Comments on "Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model." *IEEE Trans. Inform. Foren. Secur.* 14, 12 (2019), 3344–3345. DOI: 10.1109/TIFS.2018.2866304
[25] Vanga Odelu, Ashok Kumar Das, and Adrijit Goswami. 2015. A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Trans. Inform. Foren. Secur.* 10, 9 (2015), 1953–1966.
[26] Kaiping Xue, Peilin Hong, and Changsha Ma. 2014. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *J. Comput. Syst. Sci.* 80, 1 (2014), 195–206.
[27] Ruhul Amin, S. K. Hafizul Islam, Prosanta Gope, Kim-Kwang Raymond Choo, and Nachiket Tapas. 2018. Anonymity preserving and lightweight multi-medical server authentication protocol for telecare medical information system. *IEEE J. Bio. Health Inform.* 23, 4 (2018), 1749–1759. DOI: 10.1109/JBHI.2018.2870319
[28] Ding Wang, Wenting Li, and Ping Wang. 2018. Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks. *IEEE Trans. Ind. Inform.* 14, 8 (2018), 4081–4092.
[29] Y. Park and Y. Park. 2016. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* 16, 12 (2016), 2123.
[30] J. Jung, J. Moon, D. Lee, and D. Won. 2017. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. *Sensors* 17, 3 (2017), 644–665.
[31] Fan Wu, Lili Xu, Saru Kumari, Xiong Li, Jian Shen, Kim Kwang Raymond Choo, Mohammad Wazid, and Ashok Kumar Das. 2017. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* 89 (2017), 72–85.
[32] Rifaqat Ali, Arup Kumar Pal, Saru Kumari, Marimuthu Karuppiah, and Mauro Conti. 2018. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Gener. Comput. Syst.* 84 (2018), 200–215.
[33] Qi Jiang, Jianfeng Ma, Xiang Lu, and Youliang Tian. 2015. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer Peer Netw. Appli.* 8, 6 (2015), 1070–1081.
[34] Fan Wu, Lili Xu, Saru Kumari, and Xiong Li. 2017. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer Peer Netw. Appli.* 10, 1 (2017), 16–30.
[35] Yanrong Lu, Guangquan Xu, Lixiang Li, and Yixian Yang. 2019. Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wirel. Netw.* 25, 4 (2019), 1461–1475. DOI: 10.1007/s11276-017-1604-0
[36] Qi Jiang, Sherali Zeadally, Jianfeng Ma, and Debiao He. 2017. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* 5 (2017), 3376–3392.
[37] Shipra Kumari and Hari Om. 2016. Authentication protocol for wireless sensor networks applications like safety monitoring in coal mines. *Comput. Netw.* 104, C (2016), 137–154.
[38] Kevin D. Fairbanks. 2012. An analysis of Ext4 for digital forensics. *Digit. Invest.* 9, 15 (2012), S118–S130.
[39] Yanbin Tang, Junbin Fang, K. P. Chow, S. M. Yiu, Jun Xu, Bo Feng, Qiong Li, and Qi Han. 2016. Recovery of heavily fragmented JPEG files. *Digit. Invest.* 18 (2016), S108–S117.
[40] H. Read, E. Thomas, I. Sutherland, K. Xynos, and M. Burgess. A forensic methodology for analyzing Nintendo 3DS devices. In *Proceedings of Digital Forensics 2016.* 127–143.
[41] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, and Gaopeng Jian. 2017. Zipf's law in passwords. *IEEE Trans. Inform. Foren. Secur.* 12, 11 (2017), 2776–2791.
[42] D. Wang and P. Wang. On the implications of Zipf's law in passwords. In *Proceedings of ESORICS 2016.* 111–131.
[43] D. Wang, Z. J. Zhang, and P. Wang. Targeted online password guessing: An underestimated threat. In *Proceedings of ACM CCS 2016.* 1242–1254.
[44] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. A study of probabilistic password models. In *Proceedings of IEEE S&P 2014.* 689–704.

[45] D. Wang, D. B. He, P. Wang, and C. H. Chu. 2015. Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Trans. Depend. Secur. Comput.* 12, 4 (2015), 428–442.
[46] Chenyu Wang, Guoai Xu, and Jing Sun. 2018. A secure and anonymous two-factor authentication protocol in multi-server environment. *Secur. Commun. Netw.* 2018 (2018), Article 9062675, 15 pages. DOI: 10.1155/2018/9062675
[47] Chenyu Wang, Guoai Xu, and Jing Sun. 2017. An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks. *Sensors* 17, 12 (2017), E2946.
[48] D. He, N. Kumar, and N. Chilamkurti. 2015. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inform. Sci.* 321, C (2015), 263–277.
[49] Qi Jiang, Yuanyuan Qian, Jianfeng Ma, Xindi Ma, Qingfeng Cheng, and Fushan Wei. 2019. User centric three-factor authentication protocol for cloud-assisted wearable devices. *Int. J. Commun. Syst.* 32, 6 (2019), e3900. DOI: 10.1002/dac.3900
[50] Fengwei Zhang, Kevin Leach, Haining Wang, and Angelos Stavrou. TrustLogin: Securing password-login on commodity operating systems. In *Proceedings of ASIACCS 2015*. 333–344.
[51] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. 2002. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* 51, 5 (2002), 541–552.
[52] Frederic Amiel, Benoit Feix, and Karine Villegas. Power analysis for secret recovering and reverse engineering of public key algorithms. In *Proceedings of SAC 2007*. 110–125.
[53] S. Mangard, E. Oswald, and F. X. Standaert. 2011. One for all-all for one: Unifying standard differential power analysis attacks. *IET Inform. Secur.* 5, 2 (2011), 100–110.
[54] Josep Balasch, Benedikt Gierlichs, Roel Verdult, Lejla Batina, and Ingrid Verbauwhede. Power analysis of Atmel CryptoMemory—Recovering keys from secure EEPROMs. In *Proceedings of CT-RSA 2012*. 19–34.
[55] C. G. Ma, D. Wang, and S. D. Zhao. 2012. Security flaws in two improved remote user authentication schemes using smart cards. *Int. J. Commun. Syst.* 27, 10 (2012), 2215–2227.
[56] Chenyu Wang and Guoai Xu. 2017. Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card. *Secur. Commun. Netw.* 2017 (2017), Article 1619741, 14 pages.
[57] Mohammad Wazid, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, and Willy Susilo. 2017. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans. Depend. Secur. Comput.* Epub ahead of print. October 18, 2017. DOI: 10.1109/TDSC.2017.2764083
[58] Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, and YoungHo Park. 2018. Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE J. Bio. Health Inform.* 22, 4 (2018), 1310–1322.
[59] Mohammad Sabzinejad Farash, Muhamed Turkanovic, Saru Kumari, and Marko Holbl. 2016. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* 36, P1 (2016), 152–176.
[60] Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Athanasios V. Vasilakos, and Joel J. P. C. Rodrigues. 2018. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment. *IEEE Internet of Things J.* 5, 6 (2018), 4900–4913. DOI: 10.1109/JIOT.2018.2877690
[61] Yoney Kirsal Ever. 2019. Secure-anonymous user authentication scheme for e-healthcare application using wireless medical sensor networks. *IEEE Syst. J.* 13, 1, (2019), 456–467. DOI: 10.1109/JSYST.2018.2866067
[62] Prosanta Gope, Ashok Kumar Das, Neeraj Kumar, and Yongqiang Cheng. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Trans. Ind. Inform.* 15, 9 (2019), 4957–4968. DOI: 10.1109/TII.2019.2895030
[63] Ignacio Velásquez, Angélica Caro, and Alfonso Rodríguez. 2018. Authentication schemes and methods: A systematic literature review. *Inform. Soft. Tech.* 94 (2018), 30–37.
[64] J. Srinivas, S. Mukhopadhyay, and D. Mishra. 2017. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. *Ad Hoc Netw.* 54, C (2017), 147–169.
[65] Mohammad Wazid, Ashok Kumar Das, Muhammad Khurram Khan, Al Dhawailie Al-Ghaiheb, Neeraj Kumar, and Athanasios Vasilakos. 2018. Design of secure user authenticated key management protocol for generic IoT networks. *IEEE Internet of Things J.* 5, 1 (2018), 269–282.
[66] Hao Feng and Dylan Clarke. Security analysis of a multi-factor authenticated key exchange protocol. In *Proceedings of ACNS 2012*. 1–11.
[67] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. 1997. Key agreement protocols and their security analysis. In *Cryptography and Coding 1997*. Lecture Notes in Computer Science, Vol. 1355. Springer, 30–45.
[68] Wei Wang. 2014. Heartbleed—OpenSSL Zero-Day Bug Leaves Millions of Websites Vulnerable. Retrieved February 10, 2020 from https://thehackernews.com/2014/04/heartbleed-openssl-zero-day-bug-leaves.html?utm_source=tuicool&utm_medium=referral.
[69] Cesar Cerrudo. 2014. Why the Shellshock Bug Is Worse Than Heartbleed. Retrieved February 10, 2020 from https://www.technologyreview.com/s/531286/why-the-shellshock-bug-is-worse-than-heartbleed/.

[70] E. Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. Retrieved February 10, 2020 from https://datatracker.ietf.org/doc/rfc8446/?include_text=1.
[71] Jacob Kastrenakes. 2018. Wi-Fi Security Is Starting to Get Its Biggest Upgrade in Over a Decade. Retrieved February 10, 2020 from https://www.theverge.com/circuitbreaker/2018/6/26/17501594/wpa3-wifi-security-certification.
[72] D. Wang and P. Wang. 2014. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Comput. Netw.* 73, C (2014), 41–57.
[73] Qi Xie, Duncan S. Wong, Guilin Wang, Xiao Tan, Kefei Chen, and Liming Fang. 2017. Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model. *IEEE Trans. Inform. Foren. Secur.* 12, 6 (2017), 1382–1392.
[74] Debiao He, Neeraj Kumar, Muhammad Khurram Khan, Lina Wang, and Jian Shen. 2018. Efficient privacy-aware authentication scheme for mobile cloud computing services. *IEEE Syst. J.* 12, 2 (2018), 1621–1631.
[75] Hanguang Luo, Guangjun Wen, and Jian Su. 2020. Lightweight three factor scheme for real-time data access in wireless sensor networks. *Wirel. Netw.* 26, 2 (2020), 955–970.
[76] W. Timothy Polk, Donna F. Dodson, William E. Burr, Hildegard Ferraiolo, and David Cooper. 2015. NIST SP 800-78-4: Cryptographic Algorithms and Key Sizes for Personal Identity Verification. Retrieved February 10, 2020 from https://csrc.nist.gov/publications/detail/sp/800-78/4/final.
[77] M. Burrows, M. Abadi, and R. Needham. 1990. A logic of authentication. *IEEE Trans. Comput.* 8, 1 (1990), 18–36.
[78] Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, and Athanasios V. Vasilakos. 2018. Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment. *IEEE Trans. Depend. Secur. Comput.* 15, 5 (2018), 824–839.
[79] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. 2006. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* 38, 1 (2006), 97–139.
[80] L. Hernandez Encinas. Biometric fuzzy extractor scheme for iris templates. In *Proceedings of SAM 2009*. 563–569.

Received February 2019; revised March 2019; accepted April 2019