A Provably Secure Anonymous Biometrics-Based Authentication Scheme for Wireless Sensor Networks Using Chaotic Map

FEIFEI WANG, GUOAI XU, GUOSHENG XU
Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Guosheng Xu (e-mail: guoshengxu@bupt.edu.cn).
This work was supported by the National Key Research and Development Program of China under Grant 2018YFB0803605.

ABSTRACT The rapid growth of wireless sensor networks (WSNs) has opened new doors to realize remote monitoring in various areas. However, designing authentication protocol for resource-constrained WSNs is a challenging task. Recently, Aghili et al. introduced an efficient three-factor authentication scheme for WSNs using hash function, and argued that the scheme is immune to known attacks. However, we discover that their scheme suffers from a few serious weaknesses, such as session key disclosure attack, desynchronization attack, sensor node impersonation attack, and session-specific temporary information attack, and does not provide forward secrecy. In order to overcome the deficiency of this scheme, we propose an enhanced biometric-based authentication scheme for WSNs using Chebyshev chaotic map. We demonstrate that the proposed scheme is provably secure in the random oracle model. Furthermore, Burrows–Abadi–Needham logic analysis shows that the proposed scheme achieves mutual authentication and session key agreement. The informal analysis demonstrates that the proposed scheme is resistant to various attacks and has desirable attributes such as forward secrecy and three-factor secrecy. Finally, the security and performance comparison show that the proposed scheme is more practical.

INDEX TERMS Wireless sensor networks; Chebyshev chaotic map; biometric-based authentication; provably secure

I. INTRODUCTION
A wireless sensor network (WSN) is a distributed, self-organized network with the purpose of sensing and understanding the physical world. WSNs have been widely adopted in many areas like electronic healthcare systems, safety monitoring, industrial, and precision agriculture [1-4]. They typically have numerous sensor nodes and one or more gateway nodes, along with a large number of external users [5]. The sensor nodes are equipped with limited computation capability and storage. They are deployed in unattended environments to collect valuable information and transmit the information to the gateway. The external user is able to access the gathered data with the help of gateway.

These data are transmitted via an unprotected wireless channel. Hence, an effective protective measure should be adopted to protect the data from unauthorized access, illegal eavesdropping and tampering. An authentication protocol is a commonly adopted security mechanism [6] that provides identity authentication and establishes a session key for the communication parties to realize secure data exchange.

However, the design of a practical authentication protocol for resource-constrained WSNs is not an easy task. The hash function based schemes have high efficiency, but it is hard to guarantee the security of the session key. We give the analysis of several hash function based schemes [7-12] as an illustration. The schemes of Chang et al. [7], Amin et al. [8], Aghili et al. [9] suffer from session key disclosure attack. The schemes of Amin et al. [8], Aghili et al. [9], Chang and Le [10] are susceptible to known session-specific temporary information attack. The schemes of Amin et al. [8], Aghili et al. [9], Lu et al. [11], Jung et al. [12] do not provide forward secrecy. In order to avoid these weaknesses, a number of schemes adopting the public key cryptosystem such as ElGamal cryptosystem, and elliptic curve cryptography (ECC) are introduced [13-17]. However, these schemes involve high computing cost. In addition, many authentication schemes for WSNs [7-12, 13-16], the hash function based schemes and the...
public key cryptosystem based schemes included, have design deficiencies and suffer from various security attacks, like forgery attack, replay attack, and fail to provide user anonymity, etc. It becomes a challenging task to design an authentication protocol for WSNs that can ensure the security and appropriate cost at the same time.

In recent years, Chebyshev chaotic map has drawn significant attention from cryptology experts because of its advantages. There have been some Chebyshev chaotic map based authentication protocols introduced [18-20]. The intractability of the Chebyshev chaotic Diffie–Hellman problem (CHDHP) and its semigroup property make it feasible to establish a secure session key using Chebyshev chaotic map. Furthermore, the computation overhead of a Chebyshev polynomial is approximately 1/3 of a scalar multiplication on the elliptic curve group [21]. It significantly reduces the computing overhead and energy consumption for the resource-constrained sensor node. The design of Chebyshev chaotic map based authentication protocol for WSNs is a new and attractive solution.

A. RELATED WORK

In 2009, Das [22] introduced a smart card based authentication scheme for WSNs for the first time. Afterwards, a number of user authentication schemes for WSNs [23-26] have been proposed to improve the security or efficiency. In 2012, He et al. [27] pointed out that Das’s scheme has several security vulnerabilities, such as sensor node impersonation attack and smart card lose attack. In 2014, Turkanovic et al. [28] introduced a two-factor authentication scheme for heterogeneous WSNs using hash function. In 2015, Chang et al. [13] proposed a hash function based two-factor authentication scheme preserving user anonymity. He et al. [13] proposed a smart card based dynamic identity authentication scheme employing the ElGamal cryptosystem. In 2016, Jiang et al. [15] revealed He et al.’s scheme [13] is susceptible to user impersonation attack and offline guessing attack, and introduced an enhanced scheme employing ECC. Chang and Le [10] revealed the scheme of Turkanovic cannot resist user impersonation attack, session key disclosure attack, and introduced an improved scheme. In the same year, Lu et al. [11], Jung et al. [12] presented two smart card based authentication schemes using symmetric cryptosystem. Jung et al. also pointed out that Chang at el.’s scheme suffers from session key disclosure attack in the same literature [12]. Kumari et al. [18] introduced a two-factor authentication scheme for WSNs using chaotic map. In 2017, Xiong et al. [29] pointed out the protocols of Lu et al. and Jung et al. suffer from off-line guessing attack and fail to preserve forward secrecy, and presented an improved protocol using one-time hash chain technique. In 2018, the protocol of Kumari et al. was observed to suffer from sensor node impersonation attack [30]. Amin et al. [8] demonstrated the protocol of Chang and Le suffers from smart card loss attack and session key disclosure attack, and presented a biometric-based scheme using hash function to enhance the security. In the same year, Aghili et al. [9] revealed that Amin et al.’s scheme is unable to resist desynchronization attack and proposed an improved scheme to remedy its vulnerability.

B. OUR CONTRIBUTION

In this paper, through a further analysis of Aghili et al.’s scheme, we reveal the scheme suffers from forgery attack, sensor node impersonation attack, session key disclosure attack, desynchronization attack, as well as session-specific temporary information attack, and fails to achieve forward secrecy. To eliminate these vulnerabilities, we present an enhanced biometrics-based authentication scheme for WSNs using Chebyshev chaotic map. We discuss the security of the proposed scheme using several widely-accepted security analysis methods. Burrows–Abadi–Needham (BAN) logic analysis and the formal security analysis in the random oracle model are given to prove the correctness and security of the proposed scheme. Furthermore, the informal analysis demonstrates the proposed scheme surmounts the security weaknesses of Aghili et al.’s scheme and achieves various desirable security features. In addition, the security and performance comparison show the superiority of our scheme.

II. PRELIMINARIES

In this section, we discuss the concepts of fuzzy extractor and Chebyshev chaotic map in order to analyze the proposed scheme. In addition, we summarize the aggressive ability of adversary to cryptanalyze Aghili et al.’s scheme. We detail the notations used in this paper in Table 1.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>User</td>
</tr>
<tr>
<td>$S_j$</td>
<td>Sensor node</td>
</tr>
<tr>
<td>$GWN$</td>
<td>Gateway</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Malicious adversary</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>Identity of $U_i$</td>
</tr>
<tr>
<td>$did_j$</td>
<td>Dynamic identity of $U_i$</td>
</tr>
<tr>
<td>$sid_j$</td>
<td>Identity of sensor node</td>
</tr>
<tr>
<td>$PW_i$</td>
<td>Password of $U_i$</td>
</tr>
<tr>
<td>$B_i$</td>
<td>Biometric of $U_i$</td>
</tr>
<tr>
<td>$T_{1}, T_{2}, T_{3}, T_{4}$</td>
<td>Timestamp</td>
</tr>
<tr>
<td>$X_G$</td>
<td>Master key of $GWN$</td>
</tr>
<tr>
<td>$e_j$</td>
<td>Secret key of $S_j$</td>
</tr>
<tr>
<td>$sk$</td>
<td>Established session key</td>
</tr>
<tr>
<td>$|$</td>
<td>The string concatenation operation</td>
</tr>
<tr>
<td>$\bot$</td>
<td>Invalid symbol</td>
</tr>
<tr>
<td>$h()$</td>
<td>Hash function</td>
</tr>
</tbody>
</table>

A. FUZZY EXTRACTOR

The fuzzy extractor is a commonly-adopted biometric key extraction method. The fuzzy extractor consists of a probabilistic generation function $GEN$ and a deterministic reproduction function $REP$. 

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.
According to the widely adopted adversary model for WSNs

C. ADVERSARY MODEL

In this section, we outline the adversary model for WSNs

VOLUME XX, 2017  3

B. USER REGISTRATION PHASE

In this phase, \( U_i \) submits his identity and personal certificate to GWN for obtaining access permission.

Step1. \( U_i \) picks his identity \( ID_i \), and delivers his identity and personal certificate to GWN via the reliable channel.

Step2. GWN checks the uniqueness of the user identity, and calculates \( did_i = h(ID_i \parallel r_1) \). \( f_i = h(did_i \parallel X_G) \), where \( r_1 \) is a random number. GWN stores \{ \( did_i, f_i \) \} in a smart card, and issues it to \( U_i \) securely.

Step3. Upon receiving the smart card, \( U_i \) inputs his password \( PW_i \), his biometric \( B_i \). The smart card computes \( \{ \sigma_i, \theta_i = GEN(B_i) \} \), \( A_i = h(ID_i \parallel PW_i \parallel \sigma_i) \), \( E_i = \theta_i \oplus h(ID_i \parallel PW_i) \). \( C_i = f_i \oplus h(PW_i \parallel \sigma_i) \). \( \eta = PW_i \oplus h(ID_i \parallel \sigma_i) \). \( \phi = h(ID_i \parallel \sigma_i) \). The smart card stores the parameters \( \{ did_i, C_i, E_i, A_i, \eta, \phi \} \) in its memory.

D. AUTHENTICATION AND KEY AGREEMENT PHASE

\( U_i \) and \( S_j \) verify the legality of each other and establish a session key with the assistance of GWN using the following steps:

Step1. \( U_i \) attaches his smart card to a terminal, enters \( ID_i, PW_i \), and inputs his biometric \( B_i \). The smart card computes \( \{ \sigma_i, \theta_i = GEN(B_i) \} \), \( A_i = h(ID_i \parallel PW_i \parallel \sigma_i) \), \( E_i = \theta_i \oplus h(ID_i \parallel PW_i) \). \( C_i = f_i \oplus h(PW_i \parallel \sigma_i) \). \( \eta = PW_i \oplus h(ID_i \parallel \sigma_i) \). \( \phi = h(ID_i \parallel \sigma_i) \). The smart card checks if \( A_i^* = A_i \). If the equation holds, proceed to next step.

Step2. The smart card computes \( L_i = k_i \oplus h(did_i \parallel f_i \parallel T_j) \), \( P_i = sid_i \oplus h(f_i \parallel T_i) \). \( Q_i = h(ID_i \parallel \sigma_i \parallel T_i) \). \( N_i = h(did_i \parallel k_i \parallel f_i \parallel T_j \parallel sid_i \parallel Q_i) \), where \( k_i \) is a random number, \( T_j \) is the current timestamp. The smart card sends the message \{ \( did_i, N_i, P_i, Q_i, L_i, T_i \) \} to GWN via the public channel.

Step3. After getting \{ \( did_i, N_i, P_i, Q_i, L_i, T_i \) \}, GWN checks if \( T_j \) is fresh. If so, GWN computes \( f_i' = h(did_i \parallel X_G) \), \( k_i' = L_i \oplus h(did_i \parallel f_i' \parallel T_j) \), \( h(ID_i) = Q_i \oplus h(k_i' \parallel T_j) \), \( sid_i' = P_i \oplus h(f_i' \parallel T_i) \), \( N_i' = h(did_i \parallel k_i' \parallel f_i' \parallel T_i \parallel sid_i' \parallel Q_i) \), checks if \( N_i' = N_i \). If so, proceed to next step.

Step4. GWN computes \( e_j = h(sid_j \parallel X_G) \), \( D_i = h(h(ID_i) \parallel e_j \parallel T_2 \parallel k_i'') \), \( Y_i = h(h(ID_i) \parallel h(e_j \parallel T_2) \parallel V_j = k_i'' \oplus h(ID_i) \), where \( T_2 \) is the current timestamp. GWN transmits \{ \( D_i, Y_i, V_j, T_2 \) \} to \( S_j \) via the public channel.

Step5. Upon getting \{ \( D_i, Y_i, V_j, T_2 \) \}, \( S_j \) verifies whether \( T_2 \) is valid or not. \( S_j \) computes \( h(ID_i) = Y_i \oplus h(e_j \parallel T_2) \), \( k_i'' = h(ID_i) \oplus V_j \), \( D_i'' = h(h(ID_i) \parallel e_j \parallel T_2 \parallel k_i'') \), checks if \( D_i'' = D_i \). If so, proceed to next step.

III. REVIEW OF AGHILII et al.’S SCHEME

In this section, we outline the Aghili et al.’s biometrics-based authentication scheme for WSNs, which includes the following three phases:

A. SYSTEM INITIALIZATION PHASE

To initialize the system, GWN selects its master key \( X_G \). For the sensor node \( S_j \), GWN chooses its identity \( sid_j \) and computes its secret key \( e_j = h(sid_j \parallel X_G) \). GWN issues \{ \( sid_j, e_j \) \} to \( S_j \) securely.
Step 6. $S_i$ computes $sk = h(IID_i) \parallel sid_i \parallel k_1 \parallel k_2$, $W_i = H(sk \parallel T_3)$. $k_3 = k_1 \oplus k_2$, where $k_2$ is a random number generated by $S_i$, $T_3$ is the current timestamp. $S_i$ transmits $\{W_i, k_3, T_3\}$ to GWN via the public channel.

Step 7. Upon getting $\{W_i, k_3, T_3\}$, GWN verifies if $T_3$ is valid. GWN computes $k_2'' = k_3 \oplus k_1$, $sk = h(IID_i) \parallel sid_i \parallel k_1 \parallel k_2''$. $W_i' = H(sk \parallel T_3)$, checks if $W_i' = W_i$. If so, GWN computes $M_1 = h(sk \parallel k_2' \parallel T_4)$, where $T_4$ is the current timestamp. GWN transmits $\{M_1, k_3, T_3\}$ to $U_i$.

Step 8. Upon getting $\{M_1, k_3, T_3\}$, the smart card checks if $T_4$ is fresh. Then the smart card computes $k_2''' = k_3 \oplus k_1$, $sk = h(IID_i') \parallel sid_i \parallel k_1 \parallel k_2'''$. $M_i' = h(sk \parallel k_2''' \parallel T_3)$, checks if $M_i' = M_i$. If so, $U_i$ establishes a session key with the sensor node $S_i$ successfully. The smart card computes $M_2 = ID_i^o \oplus h(sk \parallel k_1)$, sends $\{M_2\}$ to GWN.

Step 9. After getting $\{M_2\}$, GWN computes $ID_i^o = M_2 \oplus h(sk \parallel k_1)$, checks if $h(ID_i')$ is equal with $Q_i \oplus h(k_1' \parallel T_3)$. If so, GWN chooses a random number $r_3$, computes $did_i^{new} = h(ID_i \parallel r_3)$, $f_i^{new} = h(did_i^{new} \parallel X_G)$, $M_3 = did_i^{new} \oplus h(did_i \parallel k_2')$, $M_4 = f_i^{new} \oplus h(f_i \parallel k_1')$, $M_5 = h(ID_i) \parallel M_3 \parallel M_4$. GWN transmits $\{M_3, M_4, M_5\}$ to $U_i$.

Step 10. Upon getting $\{M_3, M_4, M_5\}$, the smart card computes $M_5' = h(ID_i) \parallel M_3 \parallel M_4$, checks if $M_5' = M_5$. If so, the smart card computes $did_i^{new} = M_3 \oplus h(did_i \parallel k_2'')$, $f_i^{new} = M_4 \oplus h(f_i \parallel k_1')$, $C_i^{new} = f_i^{new} \oplus h(PW_i' \parallel \sigma_i')$, replaces $did_i, C_i$ with $did_i^{new}, C_i^{new}$ in its memory.

IV. CRYPTANALYSIS OF AGHILI et al.’s SCHEME

Aghili et al. pointed out that Amin et al.’s scheme is susceptible to several security attacks, and introduced an improved scheme. But we observe that Aghili et al.’s scheme also has serious vulnerabilities. We reveal the design defects of Aghili et al.’s scheme and detail the cryptanalysis of the scheme in this section.

Before analyzing Aghili et al.’s scheme, we clarify a few ambiguities in the authentication scheme for WSNs.

- The user identity should not be viewed as a secret. In many cases, the user identity is disclosed, such as shoulder surfing attack or the login page may remember user identity by default.
- The identity of the sensor node is available in some cases. For instance, it is exposed to the registered users. The attacker is able to obtain it by gathering information.
- The random number may be leaked as the result of any one communication party using an unsecured random number generator. When evaluating the security of the session key, the session-specific temporary information attack should be considered.

A. DESIGN DEFECTS OF AGHILI et al.’s SCHEME

In this section, we reveal the design defects of Aghili et al.’s scheme. Note that these defects also exist in Amin et al.’s scheme.

- The user identity is used as a secret to transmit random number $k_1$ in the public channel. Once the attacker reveals user identity, he can easily obtain $k_1$.
- The random number $k_3$ is used as a key to transmit the random number $k_2$ in the public channel. Once one random number is compromised, the other one will inevitably be revealed. The attacker may breach the session key by exploiting this loophole.
- Aghili et al.’s scheme updates did_i and the authentication value $f_i$ in each session with the purpose of making user identity untraceable. It not only adds computation and communication overhead, but may lead to denial of service attack.
- When updating did_i and the authentication value $f_i$, the user does not verify the authenticity of GWN. In addition, there is no timestamp mechanism to protect the scheme from replay attack.

B. FORWARD SECRECY

In the case that the attacker compromises the master key of GWN, he is able to obtain the session key in the following steps.

Step 1. The attacker intercepts the messages $\{did_i, N_i, P_i, Q_i, L_i, T_1\}$ and $\{M_1, k_3, T_3\}$ from public channel.

Step 2. The attacker computes $f_1 = h(did_i \parallel X_G), sid_j = P_i \oplus h(f_1 \parallel T_1), k_1 = L_i \oplus h(did_i \parallel f_1 \parallel T_1)$, $h(ID_i) = Q_i \oplus h(k_1 \parallel T_1), k_2 = k_1 \oplus k_3$. The attacker computes the session key $sk = h(ID_i) \parallel sid_i \parallel k_1 \parallel k_2$.

C. SESSION KEY DISCLOSURE ATTACK

Step 1. The attacker intercepts $\{D_i, V_i, V_t, T_2\}$, $\{W_i, k_3, T_3\}$, $\{M_1, k_3, T_4\}$ from public channel. $\{W_i, k_3, T_3\}$ is a response to $\{D_i, V_i, V_t, T_2\}$, $\{W_i, k_3, T_3\}$ and $\{M_1, k_3, T_4\}$ contain the same $k_3$. The attacker is able to distinguish the messages $\{D_i, V_i, V_t, T_2\}$, $\{W_i, k_3, T_3\}$, and $\{M_1, k_3, T_4\}$ that belong to the same session from public channel.

Step 2. The attacker reveals the user identity $ID_i$ and the identity of sensor node sid_j.

Step 3. The attacker computes $k_3 = h(ID_i) \oplus V_i$, $k_2 = k_1 \oplus k_3$, $sk = h(h(ID_i) \parallel sid_i \parallel k_1 \parallel k_2)$. The session key is exposed to the attacker.

D. FORGERY ATTACK

The attacker intercepts $\{D_i, V_i, V_t, T_2\}$, $\{W_i, k_3, T_3\}$ and $\{M_1, k_3, T_4\}$ from public channel. In addition, the attacker reveals $ID_i$ and sid_j. The attacker tries to generate a forged message $\{M_1^*, k_3^*, T_4\}$ using the following steps:
Step 1. The attacker chooses a random number $k_3^*$, computes $k_1 = h(ID_1) \oplus V_i$, $sk = h(h(ID_1) \parallel sid_1 \parallel k_1 \parallel k_2^*)$, $M_4^* = h(sk \parallel k_2^* \parallel T_3)$, $k_3^* = k_1 + k_2^*$, where $T_3$ is the current timestamp. The attacker transmits $\{M_1^*, k_3^*, T_3\}$ to $U_i$.

Step 2. After receiving $\{M_1^*, k_3^*, T_3\}$, as $T_3$ is fresh, the smart card computes $k_2^* = k_3^* \oplus k_1$, $sk = h(h(ID_1^* \parallel sid_1 \parallel k_1^* \parallel k_2^*'))$, $M_4^* = h(sk \parallel k_2'' \parallel T_3)$. Obviously, $M_1^* = M_4^*$, $U_i$ regards the message as legitimate.

Afterwards, the attacker can use the established session key to transmit the bogus information to $U_i$.

E. SENSOR NODE IMPERSONATION ATTACK
The attacker intercepts $\{D_i, Y_i, V_i, T_2\}$, $\{W_i, k_3, T_3\}$ and $\{M_4, k_3, T_3\}$ from public channel. Besides, the attacker reveals $ID_1$ and $sid_1$. Then the attacker impersonates the sensor node $S_j$ to establish a session key with $U_i$ in the following steps:

Step 1. The attacker computes $k_1 = h(ID_1) \oplus V_i$, $sk = h(h(ID_1) \parallel sid_1 \parallel k_1 \parallel k_2^*)$, $W_i^* = h(sk \parallel T_3)$. Upon getting $\{W_i^*, k_3^*, T_3\}$ from public channel. Besides, the attacker reveals $ID_1$ and $sid_1$. Then the attacker computes $k_2 = k_3^* \oplus k_3$ or $k_4 = k_2^* \oplus k_3$, $sk = h(h(ID_1) \parallel sid_1 \parallel k_1 \parallel k_2^*)$, $W_i^* = h(sk \parallel T_3)$. As $W_i^* = W_i^*$, $GWN$ computes $M_1 = h(sk \parallel k_2' \parallel T_3)$, where $T_3$ is the current timestamp. $GWN$ transmits $\{M_1, k_3, T_3\}$ to $U_i$.

Step 2. After receiving $\{W_i^*, k_3^*, T_3\}$, as $T_3$ is fresh, $GWN$ computes $k_2^* = k_3^* \oplus k_1$, $sk = h(h(ID_1^* \parallel sid_1 \parallel k_1^* \parallel k_2^*)$, $W_i^* = h(sk \parallel T_3)$. As $W_i^* = W_i^*$, $GWN$ computes $M_1 = h(sk \parallel k_2' \parallel T_3)$, where $T_3$ is the current timestamp. $GWN$ transmits $\{M_1, k_3, T_3\}$ to $U_i$.

Step 3. Upon getting $\{M_4, k_3, T_3\}$, as $T_3$ is fresh, the smart card computes $k_2'' = k_3 \oplus k_1$, $sk = h(h(ID_1^* \parallel sid_1 \parallel k_1 \parallel k_2'')$, $M_4^* = h(sk \parallel k_2'' \parallel T_4)$. Obviously, $M_1^* = M_1^*$. $U_i$ regards that he establishes a session key with the sensor node $S_j$.

F. DESYNCHRONIZATION ATTACK
The attacker obtains user identity $ID_1$ by shoulder surfing, and intercepts $\{M_3, M_4, M_5\}$ from the public channel. Then he performs the following steps:

Step 1. The attacker chooses two random binary strings $M_3^*$, $M_4^*$ whose length are equal with $M_3$, $M_4$ respectively. The attacker computes $M_5^* = h(h(ID_1^* \parallel M_3^* \parallel M_4^*)$, sends $\{M_3^*, M_4^*, M_5^*\}$ to $U_i$.

Step 2. After receiving $\{M_3^*, M_4^*, M_5^*\}$, the smart card computes $M_4^* = h(h(ID_1^*)) \parallel M_3^* \parallel M_4^*)$. As $M_4^* = M_5^*$, the smart card computes $\bar{d}_{id}^{new} = M_3^* \oplus h(d_{id} \parallel k_1')$, $C_i^{new} = f_i^{new} \oplus h(PW_i' \parallel \sigma_i')$, replaces $\bar{d}_{id}, C_i$ with $\bar{d}_{id}^{new}, C_i^{new}$ in its memory.

When $U_i$ intends to access the sensor node next time, the smart card generates a login request using the forged $\bar{d}_{id}^{new}, f_i^{new}$. After receiving the message, as $N_i' \neq N_i$, $GWN$ regards the message is illegitimate and rejects it. $U_i$ is refused to access any sensor node.

G. SESSION-SPECIFIC TEMPORARY INFORMATION ATTACK
Suppose that the random number $k_1$ or $k_2$ is compromised. The attacker intercepts $\{W_i, k_3, T_3\}$ and $\{M_4, k_3, T_3\}$ from public channel. Besides, the attacker reveals $ID_1$ and $sid_1$. Then the attacker computes $k_2 = k_3^* \oplus k_3$ or $k_4 = k_2^* \oplus k_3$, $sk = h(h(ID_1) \parallel sid_1 \parallel k_1 \parallel k_2^*)$.

V. OUR PROPOSED SCHEME
In this section, we propose a three-factor authentication scheme for WSNs using chaotic map. The proposed scheme consists of the following four phases.

A. SYSTEM INITIALIZATION PHASE
$GWN$ chooses its master key $X_G$. In addition, $GWN$ picks a large prime $p$, and selects a number as $x$ from $[-\infty, +\infty]$. $GWN$ generates a random number $\varphi$, and computes $P_G = T_{\varphi}(x)$. $GWN$ keeps $X_G, \varphi$ as secret, publishes $x, p$. For sensor node $S_j$, $GWN$ chooses its identity $sid_j$ and computes its secret key $e_j = h(sid_j \parallel X_G)$. $GWN$ issues $\{sid_j, e_j\}$ to $S_j$ securely.

B. USER REGISTRATION PHASE
$U_i$ registers with $GWN$ in the following steps, as illustrated in Figure 1.

`FIGURE 1. User registration phase`
Step1. \(U_i\) chooses his identity \(ID_i\). \(U_i\) transmits \(ID_i\) and his personal voucher to \(GWN\) through the reliable channel.

Step2. After received the registration request, \(GWN\) checks if \(ID_i\) is unique. If it holds, \(GWN\) stores \(\{f_i, P_G\}\) in the memory of a smart card, and issues the smart card to \(U_i\) securely.

Step3. After receiving the smart card, \(U_i\) enters his password \(PW_i\), imprints his biometric \(B_i\). The smart card computes \(\sigma_i = \text{RE}P(B_i, \theta_i)\). \(A_i = h(ID_i \parallel PW_i \parallel \sigma_i) \mod \gamma\). \(C_i = f_i \oplus h(PW_i \parallel \sigma_i)\), where \(\gamma\) is a integer satisfying \(2^{8} \leq \gamma \leq 2^{10}\). The smart card finally stores the parameters \(<\theta_i, A_i, C_i, \gamma, P_G>\).

C. AUTHENTICATION AND KEY AGREEMENT PHASE

In this phase, \(U_i\) and \(S_j\) authenticate each other and establish a session key for secure data transmission by the aid of \(GWN\), as shown in Figure 2.

Step1. \(U_i\) attaches his smart card to a terminal, enters \(ID_i, PW_i, B_i\), and imprints his biometric \(B_i\). The smart card computes \(\sigma_i = \text{RE}P(B_i, \theta_i)\), \(A_i = h(ID_i \parallel PW_i \parallel \sigma_i) \mod \gamma\), and checks if \(A_i = A_i\). If the equation holds, proceed next step.
Step2. The smart card computes \( f_i = C_i \oplus h(PW_i^* \parallel \sigma_i^*) \), \( E_i = T_a(\alpha(x)) \), \( d_id_i = T_a(P_i) \oplus ID_i \), \( L_i = \text{sid}_i \oplus h(f_i, \parallel T_i) \), \( Q_i = h(f_i \parallel E_i) \), \( N_i = h(ID_i \parallel Q_i \parallel \text{sid}_i \parallel T_i) \), where \( \alpha \) is a random number, \( T_i \) is the current timestamp. The smart card sends \( \{d_id_i, E_i, k_i, N_i, T_i\} \) to GWN via the public channel.

Step3. After receiving \( \{d_id_i, E_i, k_i, N_i, T_i\} \), GWN checks if \( T_i \) is fresh. If it not holds, the protocol aborts. Otherwise, GWN computes \( ID_i = d_id_i \oplus T_i \). \( f_i = h(ID_i \parallel X_0) \), \( sid_i = L_i \oplus h(f_i \parallel T_i) \), \( Q_i = h(f_i \parallel E_i) \), \( N_i = h(ID_i \parallel Q_i \parallel \text{sid}_i \parallel T_i) \), checks if \( N_i \neq N_i \). If it holds, proceed next step.

Step4. GWN computes \( e_i = h(\text{sid}_i \parallel X_0) \), \( Y_i = h(e_i \parallel E_i) \oplus Q_i \), \( D_i = h(Q_i \parallel Y_i \parallel e_i \parallel E_i \parallel T_2) \), and transmits \( \{D_i, Y_i, E_i, T_2\} \) to \( S_j \) via the public channel.

Step5. Upon receiving \( \{D_i, Y_i, E_i, T_2\}, S_j \) verifies whether \( T_2 \) is valid, and computes \( Q_i'' = Y_i \oplus h(e_i \parallel E_i) \), \( D_i' = h(Q_i'' \parallel Y_i \parallel e_i \parallel E_i \parallel T_2) \), checks if \( D_i' = D_i \). If the equation holds, proceed next step.

Step6. \( S_j \) computes \( R_i = T_{\beta}(x), K_i = T_{\beta}(E_i) \), where \( \beta \) is a random number. \( S_j \) computes the session key \( sk = h(K_i \parallel Q_i'' \parallel \sigma_i) \), and \( G_i = h(sk \parallel R_i) \), \( W_i = h(R_i \parallel G_i \parallel e_i \parallel E_i \parallel T_2) \). \( S_j \) transmits \( \{R_i, G_i, W_i, T_2\} \) to GWN via the public channel.

Step7. Upon receiving \( \{R_i, G_i, W_i, T_3\} \), GWN verifies whether \( T_3 \) is valid, and computes \( W_i' = h(R_i \parallel G_i \parallel e_i \parallel E_i \parallel T_3) \), checks if \( W_i' = W_i \). If it holds, GWN computes \( M_i = h(R_i \parallel G_i \parallel f_i \parallel T_3) \), transmits \( \{R_i, M_i, T_4\} \) to \( U_i \) via the public channel.

Step8. Upon receiving \( \{R_i, M_i, T_4\} \), the smart card checks if \( T_4 \) is fresh. If it holds, the smart card computes \( K_i' = T_{\beta}(R_i), sk = h(K_i' \parallel Q_i), G_i' = h(sk \parallel R_i) \), computes \( M_i' = h(R_i \parallel G_i' \parallel f_i \parallel T_4) \), checks if \( M_i' = M_i \). If the equation holds, \( U_i \) believes that he establishes a session key with the sensor node successfully.

D. PASSWORD UPDATE PHASE

In this phase, \( U_i \) changes his password to a new one in the following steps.

Step1. \( U_i \) attaches his smart card to a terminal, enters \( ID_i^*, PW_i^* \), and imprints \( B_i^* \). The smart card computes \( \sigma_i^* = REP(B_i^* \parallel \theta) \), \( f_i = C_i \oplus h(PW_i^* \parallel \sigma_i^*) \), \( A_i^* = h(ID_i^* \parallel PW_i^* \parallel \sigma_i^*) \mod \gamma \), checks if \( A_i^* = A_i \). If it holds, the smart card asks the user to input a new password.

Step2. \( U_i \) enters his new password \( PW_i^{\text{new}} \). The smart card computes \( C_i^{\text{new}} = f_i \oplus h(PW_i^{\text{new}} \parallel \sigma_i^*) \), \( A_i^{\text{new}} = h(ID_i^* \parallel PW_i^{\text{new}} \parallel \sigma_i^*) \mod \gamma \), replaces \( C_i, A_i \) with \( C_i^{\text{new}}, A_i^{\text{new}} \) in its memory.

VI. SECURITY ANALYSIS

A. FORMAL SECURITY ANALYSIS

In this section, we present the formal security model of three-factor authentication for WSNs and demonstrate our scheme is provably secure in this model.

1). SECURITY MODEL

Our formal security model of three-factor authentication for WSNs is based on the security model of three-factor authentication introduced by Feng et al. [33]

Participants. The participants of authentication scheme for WSNs consist of the user \( U_i \), the gateway \( GWN \), and the sensor node \( S_j \). Each entity involves multiple instances, that is, \( U_i^a, S_j^a, \) and \( RC^a \).

Queries. The adversary can make the following queries.

\textbf{Execute (}) \( (U_i^a, S_j^a, GWN^a \). \) This query simulates the passive attack. It reveals the messages transmitted via public channel to the adversary.

\textbf{Send (}) \( (U_i^a / S_j^a / GWN^a / m) \). The adversary masquerades as the instance \( U_i^a / S_j^a / GWN^a \) to send a message \( m \). The oracle disposes \( m \) in accordance with the protocol and sends back a response to the adversary.

\textbf{Reveal (}) \( (U_i^a, S_j^a \). \) If the instance \( U_i^a \) or \( S_j^a \) has established a session key. The query retrieves the session key to the adversary. Otherwise, it returns \( \perp \).

\textbf{Corrupt (}} \( (U_i^a, z) \). The oracle reveals one or two authentication factors of user to the adversary. This query is also employed to evaluate three-factor secrecy. That is, in the case that two authentication factors are compromised, whether the adversary is able to break the other one or not.

\textbf{when z = 1, it retrieves the password.}
\textbf{When z = 2, it reveals the smart card.}
\textbf{When z = 3, it reveals the biometric.}
\textbf{Corrupt (}} \( (S_j^a, GWN^a) \). The query discloses the master key of GWN or the secret key of sensor node. It is employed to test the semantic security of session key. The adversary can make this query no more than once. In the case that the instance \( U_i^a \) or \( S_j^a \) is fresh (see below), the oracle flips a coin \( b \). If \( b = 1 \), it answers the session key. Otherwise, it answers an equal-length random string.

\textbf{Freshness.} If all the following conditions are satisfied. We say the instance is fresh.

\textbf{when z = 1, it retrieves the password.}
\textbf{2. The instance and its partner are never asked a Reveal query.}
\textbf{3. The adversary never asks a Corrupt (} \( (S_j^a, GWN^a) \) query.

\textbf{Semantic security.} The adversary makes aforementioned queries adaptively and guesses the value of \( b \) to be \( b' \). The advantage that the adversary breaks the semantic security of our scheme is defined based on the probability he correctly guesses the value of \( b \). It is described as below.

\[ Adv_{\text{adv}}^{\text{se}}(A) = 2Pr(b' = b) - 1 \]

2). FORMAL SECURITY PROOF

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Theorem 1. $P$ denotes the proposed scheme. There is an adversary $\mathcal{A}$ who tries to break the semantic security of our scheme in time $t$. Suppose that $\mathcal{A}$ can ask no more than $q_s$ Send-queries, $q_e$ Execute queries, $q_h$ Hash queries, and performs no more than $q_f$ deterministic reproduction algorithm of fuzzy extractor. Then we have

$$Adv_p^{ake}(\mathcal{A}) \leq \frac{q_h^2 + 6q_e}{2^{l_1}} + \frac{2q_s}{2^{l_2}} + \frac{(q_s + q_e)^2}{2^{l_3}} + \frac{2q_s}{\epsilon} + 2q_hAdv_p^{CDHDP}(\mathcal{A})$$

where $l_1, l_2, l_3$ are the bit length of hash output, deterministic reproduction function output, Chebyshev polynomial, respectively. And $\epsilon$ is the password dictionary space. $Adv_p^{CDHDP}(\mathcal{A})$ is the probability $\mathcal{A}$ solves CDHDP in polynomial time.

The Proof. There are a series of defined games from $G_0$ to $G_6$. $S_I$ denotes $\mathcal{A}$ correctly guesses the value of $b$ in game $G_0$. $P r[S_I]$ is the probability of $S_I$.

$G_0$: It indicates the real attack. Obviously, we have

$$Adv_p^{ake}(\mathcal{A}) = 2(P r[S_0]) - 1$$

Through a conversion, yields that

$$Adv_p^{ake}(\mathcal{A}) = 2(P r[S_0] - P r[S_0]) + 2P r[S_0] - 1 = 2 \sum_{i=0}^{5}(P r[S_i] - P r[S_{i+1}]) + 2P r[S_0] - 1$$

(2)

$G_1$: The hash oracle is modeled in this game. A hash list $H_\beta$ is created. For a hash query $h(\alpha)$, if there is an item $(\alpha, \beta)$ in $H_\beta$, the oracle returns $\beta$ to the adversary. Otherwise, the oracle returns a random number $\beta$ to the adversary and adds the item $(\alpha, \beta)$ to $H_\beta$. Besides, all the oracles involved in security model are simulated in this game. Obviously, this game has no difference with $G_0$. We have

$$P r[S_I] - P r[S_0] = 0$$

(3)

$G_2$: We avoid the occurrence of some collisions in this game. $G_2$ is indistinguishable from $G_1$, unless the following conditions occur.

- A collision happens on hash queries, the probability is less than $q_h^2/2^{l_1+1}$.
- A collision happens on $H_1$ or $R_1$, the probability is less than $(q_s + q_e)^2/2^{l_3+1}$.

So we have

$$|P r[S_2] - P r[S_0]| \leq \frac{q_h^2}{2^{l_1+1}} + \frac{(q_s + q_e)^2}{2^{l_3+1}}$$

(4)

$G_3$: In this game, we avoid the situation the adversary correctly guesses $<N_1, D_1, W_1, M_1>$ without making the corresponding hash query. The probability is at most $q_s/2^{l_1}$. Thus,

$$|P r[S_3] - P r[S_2]| \leq q_s/2^{l_1}$$

(5)

$G_4$: This game averts the circumstance the adversary correctly guesses the authentication value $f_i$ directly. The probability is at most $q_s/2^{l_1}$. We get

$$|P r[S_4] - P r[S_3]| \leq q_s/2^{l_1}$$

(6)

$G_5$: In this game, we avoid the occurrence that the adversary computes the authentication value $f_i$ with the help of Corrupt $(U^a_i, x)$. The following three cases are included.

- The adversary queries Corrupt $(U^a_i, 1)$ and Corrupt $(U^a_i, 2)$. To derive $f_i$, the adversary still requires the biometric. The probability that he correctly guesses $\sigma_i$ is at most $q_s/2^{l_2}$.
- The adversary queries Corrupt $(U^a_i, 2)$ and Corrupt $(U^a_i, 3)$. The probability that he correctly guesses the password is less than $q_s/\epsilon$.
- The adversary queries Corrupt $(U^a_i, 1)$ and Corrupt $(U^a_i, 3)$. The probability that he correctly guesses the parameter $\epsilon_i$ is no more than $q_s/2^{l_1}$.

The probability that the adversary gets $f_i$ with the help of Corrupt-queries is less than $q_s \ast \left(\frac{1}{\epsilon} + \frac{1}{2^{l_1}} + \frac{1}{2^{l_2}}\right)$. We have

$$|P r[S_5] - P r[S_4]| \leq q_s \ast \left(\frac{1}{\epsilon} + \frac{1}{2^{l_1}} + \frac{1}{2^{l_2}}\right)$$

(7)

$G_6$: In this game, we use the private oracle $h'$ instead of the hash oracle $h$ to compute the session key $sk$. As the private oracle $h'$ is unknown to the adversary, We have

$$P r[S_6] = \frac{1}{2}$$

(8)

$G_6$ has no difference with $G_5$, unless the adversary makes a hash query $h(K_{i \parallel Q_1})$, we denote the event as $A_1$. We have

$$|P r[S_6] - P r[S_5]| \leq P r[A_1]$$

(9)

$G_7$: We simulate the random self-reducibility of CHDHP in this game. For $E_2 = T_2(x), R_2 = T_2(x), \text{through selecting randomly in } H_u$, we can obtain the item containing $K_1 = T_2(T_3(x))$ with the probability $\frac{1}{q_h}$. Since the event $A_1$ denotes that the adversary makes a hash query $h(K_{i \parallel Q_1})$. We have

$$P r[A_1] \leq q_h Adv_p^{ECDHHP}$$

(10)

From (1)-(10), we have

$$Adv_p^{ake}(\mathcal{A}) \leq \frac{q_h^2 + 6q_e}{2^{l_1}} + \frac{2q_s}{2^{l_2}} + \frac{(q_s + q_e)^2}{2^{l_3}} + \frac{2q_s}{\epsilon} + 2q_hAdv_p^{CDHDP}$$

B. BAN LOGIC ANALYSIS OF THE PROPOSED SCHEME

Burrows–Abadi–Needham (BAN) Logic [34] is a broadly-accepted protocol verification method. It is used to demonstrate that the proposed scheme achieves the fundamental functions of authentication protocol, such as mutual authentication and session key establishment. Table 2 details the notations and rules of BAN logic, where $P$ and $Q$ denote the principals, $X$ denotes a statement.
The proposed scheme meets the fundamental requirements of authentication protocol for WSNs.

- Goal 1: $S_j \equiv U_i \equiv (S_j \leftrightarrow U_i)$
- Goal 2: $S_j \equiv U_i \equiv (S_j \leftrightarrow U_i)$
- Goal 3: $U_i \equiv S_j \equiv (S_j \leftrightarrow U_i)$
- Goal 4: $U_i \equiv S_j \equiv (S_j \leftrightarrow U_i)$

The idealized form. We formalize the proposed scheme as follows.

M1: $U_i \rightarrow GWN < ID, sid, U, E_i \leftrightarrow S_j, U_i \equiv Q_i \equiv S_j, T_1 >_{f_i}$
M2: $GWN \rightarrow S_j < U_i \equiv U_i \equiv Q_i \equiv S_j, U_i \equiv U_i \equiv S_j, T_2 >_{e_j}$
M3: $S_j \rightarrow GWN < R_i, E_i, T_3, U_i \equiv S_j >_{e_i}$
M4: $GWN \rightarrow U_i < R_i, E_i, T_4, S_j \equiv U_i \equiv S_j, T_3 >_{f_i}$

Assumptions. The proof of the proposed scheme is based on the following initial assumptions.

A1: $GWN| \equiv U_i \leftrightarrow GWN$
A2: $GWN| \equiv \#(T_1)$
A3: $GWN| \equiv U_i \equiv U_i \leftrightarrow S_j$
A4: $GWN| \equiv U_i \equiv U_i \equiv S_j$
A5: $S_j \equiv GWN \equiv E_i \rightarrow S_j$
A6: $S_j \equiv \#(T_2)$
A7: $S_j \equiv GWN \equiv (U_i| \equiv U_i \equiv Q_i \equiv S_j)$
A8: $S_j \equiv GWN \equiv (U_i| \equiv U_i \equiv E_i \rightarrow S_j)$
A9: $S_j \equiv U_i \equiv U_i \equiv S_j$
A10. $GWN| \equiv GWN \equiv S_j$
A11: $GWN| \equiv \#(T_3)$

Proof. The proof is implemented as follows.

From M1, it obtains that

1. $GWN \equiv ID, sid, U_i \equiv E_i \rightarrow S_j, U_i \equiv Q_i \equiv S_j, T_1 >_{f_i}$

In line with (1), A1 and the message meaning rule, it gets that

2. $GWN| \equiv U_i \equiv < U_i \equiv E_i \rightarrow S_j, U_i \equiv Q_i \equiv S_j >$

In line with (2), A2 and the nonce verification rule, it gets that

3. $GWN| \equiv U_i \equiv < U_i \equiv E_i \rightarrow S_j, U_i \equiv Q_i \equiv S_j >$

In line with (3) in line with (4), A3 and the jurisdiction rule, it gets that

4. $GWN| \equiv U_i \equiv S_j$

In line with (4), A4 and the jurisdiction rule, it gets that

5. $GWN| \equiv U_i \equiv S_j$

From M2, it obtains that

6. $S_j \equiv < U_i \equiv U_i \equiv Q_i \equiv S_j, U_i \equiv U_i \equiv E_i \rightarrow S_j, T_2 >_{e_j}$

In line with (6), A5 and the message meaning rule, it gets that

7. $S_j \equiv GWN \equiv < U_i \equiv U_i \equiv Q_i \equiv S_j, U_i \equiv U_i \equiv S_j, T_2 >_{e_j}$

In line with (7), A6, and the nonce verification rule, it gets that

8. $S_j \equiv GWN \equiv < U_i \equiv U_i \equiv Q_i \equiv S_j, U_i \equiv U_i \equiv E_i \rightarrow S_j >$

In line with (8), A7, and the jurisdiction rule, it gets that

9. $S_j \equiv U_i \equiv U_i \equiv Q_i \equiv S_j$

In line with (9), A8, and the jurisdiction rule, it gets that

10. $S_j \equiv U_i \equiv U_i \equiv S_j$

In line with (10), A9, and the jurisdiction rule, it gets that

11. $S_j \equiv U_i \equiv U_i \equiv S_j$

Goal 1

In line with (11), A9, and the jurisdiction rule, it gets that

12. $S_j \equiv U_i \equiv S_j$

Goal 2

From M3, it obtains that

13. $GWN \equiv < R_i, E_i, T_3, U_i \equiv S_j >_{e_j}$

In line with (13) and A10, and the message meaning rule, it gets that

14. $GWN| \equiv S_j \equiv < R_i, E_i, T_3, U_i \equiv S_j >$

In line with (14), A11, and the nonce verification rule, it gets that

15. $GWN| \equiv S_j \equiv U_i \equiv S_j$

In line with (15), A12, and the jurisdiction rule, it gets that

16. $GWN| \equiv U_i \equiv S_j$
The hash value guarantees that the parameters of the message are not tampered with. Besides, the authentication value $f_i$ or the secret key $e_j$ is used to verify the authenticity of sender’s identity. Hence, if the adversary modifies any parameter of the message, the receiver will detect that the message is tampered with and reject it. In addition, our scheme does not update user parameter in each session. If the adversary intercepts a message, it will terminate the current session, but will not influence the next login.

4). FORGERY ATTACK
As analyzed above, the authentication value $f_i$ or the secret key $e_j$ is employed to verify the identity of sender in every message. Without $f_i$ or $e_j$, the adversary cannot forge a valid message to defraud the $GWN$, sensor node, or the user.

5). SESSION-SPECIFIC TEMPORARY INFORMATION ATTACK
$K_i$ and $Q_i$ are required to compute the session key. In the case that the nonce $\alpha$ or $\beta$ is compromised, the adversary is able to compute $K_i = T_a(R_i) \oplus e_j = T_b(E_j)$ . However, without $f_i$, $e_j$, the adversary is incapable of getting $Q_i$. The adversary is unable to break the session key.

6). FORWARD SECRECY
With the master key of $GWN$, the adversary is able to compute $Q_i = h(f_i \parallel E_i)$. However, to get $K_i$, the adversary needs to retrieve $K_i$ from $E_i, R_i$. Because this is prevented by the intractability of CHDHP, our scheme provides forward secrecy.

7). USER ANONYMITY
The adversary intercepts \{did, $E_i, L_i, N_i, T_i$\} from the public channel, and tries to retrieve $ID_i$ from the dynamic identity $did_i$, where $did_i = T_a(P_G) \oplus ID_i$. $E_i = T_a(\alpha) \oplus P_G$ is the parameter of smart card. The adversary cannot retrieve $ID_i$, unless he gets $T_a(P_G)$. We suppose that the adversary is able to obtain $P_G$. The adversary may be a registered user, or be capable of extracting the parameters of the smart card. To compute $T_a(P_G)$, the adversary has to solve the CHDHP. It is beyond the capability of adversary. The adversary is unable to reveal the true identity of the user. In addition, the dynamic identity $did_i$ is untraceable, as it changes with the nonce $\alpha$ in each session. Compared with the scheme of Aghili et al., the proposed scheme achieves user anonymity at a smaller cost, and it is not vulnerable to desynchronization attack.

8). THREE-FACTOR SECRECY
Three-factor secrecy denotes that even if any two authentication factors are compromised, the adversary cannot break the other one or impersonate the user successfully. We demonstrate that our scheme achieves three-factor secrecy in the following three cases:

- As analyzed in off-line guessing attack, it is not possible to reveal the password of the user, even if both the smart card and biometric are compromised.

C. Informal Security Analysis
In this section, we demonstrate that the proposed scheme is secure against various attacks and provides desirable attributes such as user anonymity, forward secrecy, and three-factor secrecy.

1). OFF-LINE GUESSING ATTACK
Provided that the adversary compromises the smart card and biometric, he tries to breach the password. The adversary picks an ID$^i$ from identity dictionary space and a PW$^i$ from password dictionary space. The adversary calculates $A_i = h(ID^i \parallel PW^i \parallel \sigma^i) \mod y$, and verifies $A_i \equiv A_i$. However, our scheme adopts the fuzzy validation of the inputted authentication information. There is more than one pair of identity and password that satisfy $A_i \equiv A_i$. The adversary is incapable of distinguishing the correct one from the candidates. In our scheme, even if both the smart card and biometric are leaked, the adversary is incapable of breaching the password. Our scheme is immune to off-line guessing attack.

2). SESSION KEY DISCLOSURE ATTACK
In the proposed scheme, the session key is computed as $sk = h(K_i \parallel Q_i)$. $K_i$ and $Q_i$ are required to retrieve $sk$. $Q_i$ is computed based on $Q_i = h(f_i \parallel E_i)$, or $Q_i = h(e_j \parallel E_j) \oplus Y_i$. The adversary needs to compromise the authentication value $f_i$ or the secret key $e_j$ to get $Q_i$, $f_i, e_j$ are protected with hash function. The only way is to breach the smart card and retrieve $f_i$ from $C_i$. However, the password and biometric are required to retrieve $f_i$ from $C_i$. That is to say, the adversary needs to break all the three authentication factors at the same time. Therefore, the adversary is unable to obtain $Q_i$. On the other hand, $K_i$ is computed based on $K_i = T_a(R_i)$ or $K_i = T_b(E_i)$. To retrieve $K_i$ from $E_i, R_i$, the adversary needs to solve the CHDHP. It is beyond the capability of the adversary. Our scheme is immune to session key disclosure attack.

3). DESYNCHRONIZATION ATTACK
Our scheme employs the timestamp mechanism to protect our scheme from replay attack. In addition, we compute the hash value of the transmitted parameters with the authentication value $f_i$ or the secret key $e_j$ in every message.
• Suppose that the smart card and password of the user are compromised, the adversary tries to break the biometric. The adversary obtains the parameters $\theta_i, C_i, A_i, Y_i, F_i >$ of the smart card, where $C_i = f_i(\theta_i || PW_i || Y_i) \mod y$. However, the adversary is unable to reveal $\sigma_i$ from $A_i$ or $C_i$, as it is protected with hash function.

• In the case that the biometric and password of the user are compromised, the adversary tries to break the parameters of smart card. As $f_i$ is unavailable, it is impossible to retrieve the crucial parameter $C_i$.

VII. SECURITY AND PERFORMANCE COMPARISON

In this section, we compare our scheme with related schemes [8, 9, 16, 17] in terms of security features, computation and communication costs. When evaluating the computation and communication costs of related schemes, we concentrate on the authentication and key agreement phase.

Table 3 summarizes the security features of related schemes. The schemes of Amin et al. [8] and Aghili et al. [9] are two hash-based schemes. These two schemes have serious security weaknesses. Specifically, they suffer from session key disclosure attack, desynchronization attack, forgery attack, sensor node impersonation attack, known session-specific temporary information attack, and fail to achieve forward secrecy. Besides, the two schemes are unable to provide three-factor secrecy. When the smart card and biometric are compromised, the attacker can retrieve the password via off-line guessing attack in the two schemes. The schemes of Ryu et al. [16] and Moon et al. [17] are two ECC-based schemes. Compared with the hash-based schemes, they achieve a higher level of security, but still have vulnerabilities such as known session-specific temporary information attack. In addition, Ryu et al.’s scheme fails to preserve three-factor secrecy. Moon et al.’s scheme is susceptible to sensor node impersonation attack. Moon et al.’s scheme is a two-factor authentication scheme using biometric and smart card. Its security analysis takes no account of three-factor secrecy. Compared with the related schemes, our scheme is resistant to various attacks and provides desirable features like forward secrecy and three-factor secrecy.

Table 4 summarizes the communication overheads of related schemes. For the sake of fairness, we presume that the user’s identity and the identity of sensor node, timestamp, random number, the outcome of hash function, the output of the user’s identity and the identity of sensor node, timestamp, related schemes. For the sake of fairness, we presume that factor secrecy.

provides desirable features like forward secrecy and three-factor schemes. Our scheme is resistant to various attacks and account of three-factor secrecy. Compared with the related using biometric and smart card. Its security analysis takes no

![FIGURE 3. Communication overhead comparisons of related schemes](image1)

![FIGURE 4. Computation cost comparisons of related schemes](image2)

Table 5 summarizes the computation costs of related schemes. $T_H$ denotes a hash function. $T_S$ is a symmetric encryption/decryption. $T_P$ denotes an elliptic curve point multiplication. $T_F$ is a deterministic reproduction function of fuzzy extractor. $T_C$ denotes computing a Chebyshev polynomial. The lightweight operation “XOR” is negligible. According to [35], the computing time of $T_H, T_S, T_F, T_P, T_C, T_F$ are $0.5$ ms, $8.7$ ms, $63.08$ ms, $21.02$ ms, $63.08$ ms, respectively. The proposed scheme requires $1T_F + 5T_C + 8T_H$ in the user end. $9T_H$ in $GWN$, and $2T_C + 5T_H$ in the sensor node. Its total computation overhead is $1T_F + 4T_C + 20T_H$. The estimated
computing time of the proposed scheme is $1 \times 63.08 + 4 \times 21.02 + 20 \times 0.5 = 157.16 \text{ ms}$, while the estimated computing time of related schemes [8, 9, 16, 17] are 80.58 ms, 81.58 ms, 325.9 ms, and 466.96 ms, respectively. In addition, Figure 4 describes the computation costs of related schemes increase with the number of users. It shows that the computation cost of the proposed scheme is inferior to the hash-based schemes [8, 9], as the extra computation load such as the computation of Chebyshev polynomial is added to ensure the security of the session key. But it is vastly superior to the ECC-based schemes [16, 17].

In summary, the hash-based schemes [8, 9] are efficient, but they have serious security weaknesses. The two schemes are vulnerable to sensor node impersonation attack, that is to say, the attacker is able to impersonate the sensor node and be authenticated by GWN and user successfully. In addition, the two schemes are unable to guarantee the security of the session key, as they are vulnerable to session key disclosure attack. In these two schemes, it is difficult to implement the fundamental functions of authentication protocols correctly. Compared with the hash-based schemes, our scheme significantly improves the security at an acceptable computation cost added. In addition, the proposed scheme remarkably reduces the communication overhead. Compared with the ECC-based schemes [16, 17], the proposed scheme has absolute advantages in terms of communication and computation costs, and security properties. Hence, the proposed scheme is preferable to the related schemes.

VIII. CONCLUSION

In this study, we cryptanalyze Aghili et al.’s scheme and reveal its vulnerabilities. In their scheme, once the attacker obtains the identities of the user and sensor node, the attacker is able to compromise the session key and impersonate the sensor node, as well as perform desynchronization attack. To overcome the weaknesses of Aghili et al.’s scheme, we present a provably secure three-factor authentication scheme for WSNs using Chebyshev chaotic map. We prove the security of the proposed scheme using several widely-accepted security analysis methods such as the formal analysis under random oracle model, BAN logic, and heuristic analysis. The proposed scheme not only achieves mutual authentication and secure session key establishment, but also provides desirable features like forward secrecy and three-factor secrecy. In addition, the security and performance analysis show that the proposed scheme is superior compared with the related schemes.

### TABLE 3. Security features of related schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User anonymity</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist off-line guessing attack</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist desynchronization attack</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist session key disclosure attack</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist forgery attack</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist user impersonation attack</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist sensor node impersonation attack</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Resist known session-specific temporary information attack</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Forward secrecy</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Three-factor secrecy</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

### TABLE 4. Communication overheads of related schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total overhead</td>
<td>2560 bits</td>
<td>2560 bits</td>
<td>2176 bits</td>
<td>2240 bits</td>
<td>2048 bits</td>
</tr>
</tbody>
</table>

### TABLE 5. Computation costs of related schemes

<table>
<thead>
<tr>
<th>Schemes</th>
<th>User</th>
<th>GWN</th>
<th>Sensor Node</th>
<th>Estimated Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amin [8]</td>
<td>$14T_H + 17T_F$</td>
<td>$17T_H$</td>
<td>$4T_H$</td>
<td>80.58 ms</td>
</tr>
<tr>
<td>Aghili [9]</td>
<td>$15T_H + 17T_F$</td>
<td>$18T_H$</td>
<td>$4T_H$</td>
<td>81.58 ms</td>
</tr>
<tr>
<td>Ryu [16]</td>
<td>$8T_H + 2T_P + 1T_F$</td>
<td>$10T_H$</td>
<td>$3T_H + 2T_P$</td>
<td>325.9 ms</td>
</tr>
<tr>
<td>Moon [17]</td>
<td>$6T_H + 3T_P + 1T_F$</td>
<td>$6T_H + 1T_P + 1T_S$</td>
<td>$4T_H + 2T_P + 1T_S$</td>
<td>466.96 ms</td>
</tr>
<tr>
<td>Our scheme</td>
<td>$1T_F + 2T_C + 8T_H$</td>
<td>$9T_H$</td>
<td>$2T_C + 5T_H$</td>
<td>157.16 ms</td>
</tr>
</tbody>
</table>

ms = milliseconds.

REFERENCES


Guosheng Xu received the Ph.D. degree in information security from Beijing University of Posts and Telecommunications, China, in 2008. He is a lecturer in school of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interest is in the areas of artificial intelligence, software security and advanced cryptography.