A Secure User Authentication Scheme against Smart-Card Loss Attack for Wireless Sensor Networks Using Symmetric Key Techniques

Lei Chen, Fushan Wei, and Chuangui Ma

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou Information Science and Technology Institute, No. 62, Ke-Xue Road, Zhengzhou 450001, China

Correspondence should be addressed to Lei Chen; chenleixy0730@163.com

Received 2 December 2014; Revised 10 March 2015; Accepted 21 March 2015

Academic Editor: Hairong Qi

Copyright © 2015 Lei Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

User authentication in wireless sensor networks (WSNs) is a critical security issue due to their unattended and hostile deployment in the field. In order to protect the security of real-time data query from an external user, many two-factor (password and smart-card) user authentication schemes are proposed. However, most of them are insecure against various attacks. This paper summarizes attacks and security requirements for two-factor user authentication in WSNs. Based on security requirements, a user authentication and session key establishment scheme is also proposed, and this scheme can resist smart-card loss attack merely by using symmetric key techniques. Security and performance analysis demonstrate that, compared to the existing schemes, the proposed approach is more secure and highly efficient.

1. Introduction

Wireless sensor networks (WSNs) have emerged as a field of interest in communication technologies over the years. They usually consist of a large number of autonomous sensor nodes, which are generally deployed in unattended environments. Each sensor node has limited storage capacity and computational resource and a small communication module to communicate with the outside world over an ad hoc wireless network. Generally speaking, most of the queries in WSN applications are issued by the points of base stations or gateway- (GW-) nodes of the network. However, one can foresee that there should be great needs to access the real-time data inside WSN; where real-time data from the sensor nodes may no longer be accessed through the GW-node only, instead, the data are to be accessed directly by the external party (user) when demanded. If the data in WSN were made available to the user on demand, then authentication of the user should be ensured before the users are allowed to access the data.

According to the background above and actual application demand, in 2009, Das [1] proposed an efficient two-factor user authentication scheme firstly based on password and smart-card. He claimed that his scheme is secure against many types of attacks, such as many logged-in users with the same login identity, stolen-verifier, guessing, replay, and impersonation attacks. Since then, more and more two-factor user authentication schemes have been proposed. Nyang and Lee [2] pointed out that Das's scheme was not practical and vulnerable to an offline password guessing attack by insiders and node compromise attacks, and his scheme did not meet other security issues, that is, encryption and authenticity verification of query responses. Further, they proposed an enhanced two-factor user authentication protocol for WSNs, which overcomes the security flaws of Das's scheme with some additional security services. However, their scheme [2] also does not care about mutual authentication and is without password updates. In 2010, Khan and Alghathbar [3] pointed out that the scheme of Das [1] was insecure and cannot resist many other security attacks, such as gateway-nodes bypass attacks. To overcome the security weaknesses of Das's scheme, Khan and Alghathbar [4] proposed an improved two-factor user authentication in WSNs, which provided protection against insider attacks and GW-node...
bypass attacks at the same time; it introduced a password change phase for users. There are many other cases about improvement and enhancement. For example, He et al. [5] proposed an enhanced scheme based on Das’s scheme [1]. Their scheme keeps the merits of the original protocol and can resist the security weaknesses such as vulnerabilities to insider attack and the impersonation attack. Vaidya et al. [6] showed that Das's scheme [1] and Khan-Alghathbar’s scheme [4] have flaws and remain vulnerable to various attacks such as smart-card loss attacks. In order to overcome security weaknesses of both schemes, they proposed an improved two-factor user authentication which is resilient to smart-card loss attacks as well as other common types of attacks. Fan et al. [7] proposed a simple Denial-of-Service resistant user authentication scheme for two-tiered WSN, which is efficient. Their scheme can establish a session key between the user and a master node (cluster node) in the sensor network. In the same year, Chen and Shih [8] proposed a robust protocol for WSNs and achieved mutual authentication. Later biometric-based user authentication in WSNs has drawn some research attention. A biometric-based user authentication scheme for WSNs has been proposed by Yuan [9]. However, there also exist many flaws in Yuan's scheme. To improve the performance and the security, Yeh et al. [10] proposed the first user authentication protocol for WSNs by using the elliptic curve cryptography (ECC).

However, such schemes are still used by method of “attack-improve.” Namely, even though the two-factor user authentication protocol has been proposed, it was secure until others found its vulnerabilities and improved it. This “attack-improve” method is a never-ending cycle. For the two-factor user authentication in WSNs, there is no summary on security vulnerabilities and abilities of adversary, and nobody concludes the security requirements. What is more, there is no appropriate formal model to examine the security of the user authentication scheme for WSNs. In this paper, we will classify the summary for the attacks and security vulnerabilities discussed in the previous section. Our proposed scheme is described in Section 4. Its security analysis and performance analysis are discussed in Section 5. Finally, in Section 6, we conclude the paper with a brief summary and outline our future work.

### 2. Security Vulnerabilities/Attacks Summary

In this section, we describe all security vulnerabilities in previous works, except common types of attacks such as replay attack or stolen-verifier attack. We just focus on two-factor user authentication in WSNs. Some notations and symbols used throughout this paper are summarized in Notation.

Before summarizing the attacks, it is assumed that an adversary may have full control over the network with following capabilities.

(i) An adversary may intercept all the messages at any time.

(ii) An adversary may intercept, delete or modify, and inset any message over the public network.

(iii) An adversary may either hack passwords or steal user's smart-card and utilize secrets stored in smart-card, but not all at the same time.

(iv) An adversary may compromise the sensor node $S_n$ and extract all parameters stored in it.

Strictly speaking, almost all previous schemes are no longer safe after considering ability of adversary mentioned above, although they claimed that their schemes can resist lots of attacks. We will discuss this situation in the following paper.

#### 2.1. Privileged-Inside Attack

In registration phase, a user's password, that is, $PW_i$, is transmitted directly to the registration center in plaintext. However, in real word, it is common knowledge that many users use the same passwords in different applications or servers for their convenience of remembering long passwords and using them easily whenever required. Therefore, it would appear that the system administrator or privileged-insider of GW-node may try to use $PW_i$ to impersonate the user to access other GW-nodes where $U_i$ could be a registered user.

For insider attack, some researchers have developed various defense approaches. He et al. [5] suggested that user can transmit $h(b \oplus PW_i)$ instead of $PW_i$ to GW-node in the first time that four types of security requirements have been proposed for two-factor user authentication in WSNs. For those attacks and security requirements, we propose a secure user authentication scheme against smart-card loss attack for WSNs by using symmetric key techniques. The proposed scheme perfectly solves the open question which was pointed out in [11] and has a high efficiency.
registration phase, where $b$ is a high-entropy random number selected by $U_i$ and not revealed to the GW-node. In addition, Sun's scheme [11] suggested that $U_i$ only submits his/her ID$_i$ to the GW-node until receiving the smart-card which includes the initial password selected by GW-node; $U_i$ is able to change the initial password immediately by using the password change operation.

2.2. User Impersonation Attack. User impersonation attack can be caused by many other security flaws such as the privileged-inside attack discussed in Section 2.1, other legal users being malicious, or the parameters in the smart-card being extracted by attackers.

In registration phase, many users will receive the unique and personalized smart-card from the identical GW-node. However, the secret parameters which are generated by the GW-node and related to smart-card, such as $K$ and $x_a$, are kept unchanged for different users in some proposed scheme. In other words, different users and identical GW-nodes share the same secret parameters; namely, every entity does not have the secret parameter or key belonging to it. If the smart-card is stolen or the secret parameters are compromised, then the whole sensor network will be vulnerable to the user impersonation attack. For example, Li et al. [13] showed that an attacker $U_i$ who has registered as a user of GW-node can restore $h(K) = N_i \oplus h(ID_i \parallel PW_i)$ from his/her own password and smart-card in Da's scheme [1]. Note that $x_a$ can be extracted from $U_i$'s smart-card directly. Therefore, the attacker $U_i$ is able to impersonate another legal user or forge nonregistered user to log in GW-node legally without being noticed.

We suggest that GW-node distinguishes different users by different secret parameters. For example, GW-node can compute $h(ID_i \parallel x_a)$ instead of $x_a$ in registration phase, take a secure billing service introduced in Li's scheme [13] which stores a password/verify table, or add biometric keys in register/login phase, and so forth. However, these tricks are at the expense of anonymity. Namely, we can see from the tricks that if the user's identity ID$_i$ or biometric keys are added to register/login phase, the vulnerable schemes are robust against user impersonation attack. Therefore, to accomplish secure aims, the schemes do not have the anonymity. In order to avoid anonymity loss, it is essential to design a secure user authentication scheme with anonymity for WSNs.

2.3. Guessing Attack. Guessing attack is a crucial concern in any password-based system. The attacker can recover $h(PW_i \parallel * \ast * \ast)$ through privileged-inside attack, node compromise attack, or smart-card loss attack, where "*" is an acquired formula. Thus, the attacker can guess $U_i$'s password in a relatively small dictionary containing all the words about $U_i$. For example, in Da's scheme [1], the attacker $U_i$ who has registered as a legal user of GW-node can compute $h(ID_i \parallel PW_i) = D_{ID_i} \oplus h(ID_i \parallel PW_i)$, or an adversary can obtain any user's $h(ID_i \parallel PW_i) = h(x_a \parallel T) \oplus D_{ID_i}$ after the secret parameter $x_a$ stored in designated sensor node is compromised through node compromise attack [14]. Nyang and Lee [2] suggested that the prevention of the offline password guessing attack is using $h(ID_i \parallel PW_i \parallel x_a)$ instead of $h(ID_i \parallel PW_i)$. The other common improvement method is sending user's identity ID$_i$ and $h(b \oplus PW_i)$ to the GW-node in registration phase; then $U_i$ enters $b$ into his/her smart-card after receiving the personalized smart-card from GW-node. They claimed that their methods can resist offline password guessing attack. However, they would not achieve that goal if smart-card loss attack was taken into account (see Section 2.6).

In fact, the improvements discussed above are not a solution, which people fail to realize. Symmetric key techniques and offline validation are widely used in login phase of "smart-card-password" two-factor user authentication schemes in WSN at present. Namely, $U_i$ inserts its smart-card to a terminal in login phase and keys ID$_i$ and PW$_i$. Then the smart-card uses secret parameters and PW$_i$ to compute some formulas. The correctness of these formulas will determine whether users log in to the scheme successfully, while it is independent of GW-node. Technically, there is an unavoidable loophole; that is, if the smart-card is stolen and secret parameters such as $x_a$ and $b$ are leaked, the scheme will not be safe anymore. This problem exists in almost two-factor user authentication scheme. For example, in He's scheme [5], $H_i, V_i, b$ are stored in $U_i$'s smart-card, where $H_i = h(T_i) = h(V_i \oplus h(ID_i \parallel h(b \oplus PW_i)))$. If the attacker obtains these parameters and user's ID$_i$, he/she also can attack by guessing. Therefore, designing a genuine two-factor user authentication scheme that can defeat guessing attack is meaningful for the application.

2.4. Node Compromise Attack. Node compromise attack refers to a series of attacks caused by a malicious or captured sensor node. These attacks include guessing attack by obtaining the hash of password, impersonation of other sensor nodes by using secret parameters in captured sensor node, and GW-node bypassing attack. GW-node bypassing attack [3] means that the attacker can compute the legal messages to gain the trust of other sensor nodes by bypassing GW-node. The basic cause of above vulnerabilities is that several sensor nodes $S_j$ and GW-node share the same secret parameters such as $x_a$.

To avoid node compromise attack, Huang et al. [14] suggested using $h(x_a \parallel SID_j)$ instead of $x_a$ as a shared key between sensor node $S_j$ and GW-node. $h(x_a \parallel SID_j)$ is stored in sensor node beforehand, where $x_a$ is generated securely by the GW-node. Even if sensor node $S_j$ is captured, the attacker cannot recover the value of $x_a$ and the shared key $h(x_a \parallel SID_j)$. Therefore, sensor node that had been captured has no influence on other sensor nodes and GW-node.

2.5. GW-Node Impersonation Attack. There are at least two situations where the attack occurred. The first situation is GW-node bypassing attack, namely, adversary steals the secret shared key of GW-node from a captured sensor node to impersonate GW-node (GW-node bypassing attack can be regarded as GW-node impersonation attack). The second situation is "smart-card loss attack," that means adversary steals secret parameters from smart-card, or a malicious legitimate user recovers secret parameters from their own smart-card and impersonates GW-node. In many schemes,
smart-card would store numerous sensitive information such as \( h(PW_i), x_n \), or \( h(x_a) \) directly. The other core reason is still that different users and identical GW-nodes share the same secret parameters such as \( K \) and \( x_a \), or there is another issues like \( N_i \) is poor in design. According to Vaidya’s suggestion [6], the GW-node impersonation attack caused by smart-card loss attack can be avoided by adding various validation conditions. However, Vaidya’s suggestion cannot solve the inherent problem.

By analyzing the second situation, one way to avoid the GW-node impersonation attack is to let GW-node distinguish different users by different secret parameters. The methods are similar to user impersonation attack (see Section 2.2). However, it is a fact that all these tricks are at the sacrifice of anonymity. Therefore, to accomplish secure aims, the schemes do not have anonymity. In order to avoid anonymity loss, it is essential to design a secure user authentication scheme with anonymity for WSNs.

2.6. Smart-Card Loss Attack. Smart-card loss attack means that a variety of attacks are caused by information leakage in smart-card. If the compromised information in smart-card includes secret parameters associated with GW-node, “GW-node impersonation attack” and “GW-node bypassing attack” will happen. If it involves user’s identity and password, “password guessing attacks” will happen.

As for the appearance of guessing attack related to smart-card, adversary can recover user’s password from offline validation formulas through guessing attack. For example, in He’s scheme [5], \( H_i = h(T_i) = h(V_i \oplus h(ID_i \parallel h(b \oplus PW_i))) \) and \( H_j, V_j, b \) are all stored in smart-card (see Section 2.3). If these parameters in smart-card are compromised, adversary can easily carry out the offline password guessing attack.

For the guessing attack caused by smart-card loss attack, Sun et al. [11] pointed out that public key techniques in validation formulas of smart-card can solve this situation. However, the WSN applications cannot burden the high implementation costs of the public key algorithms. Actually, symmetric key techniques and offline validation are widely adopted in login phase of “smart-card-password” two-factor user authentication schemes in WSN at present. In the smart-card loss attack, it is still an open question whether there is a secure user authentication scheme merely by using symmetric key techniques. Therefore, designing a genuine two-factor user authentication scheme which can resist guessing attack is meaningful for the application.

2.7. Parallel Attack/No Protection against Forgery Attack. This attack works in many existing schemes. With Das’s scheme [1], for example, a malicious legal user of the system \( U^* \) can embed a synchronized Trojan virus into the other legal user \( U_i \)'s log in to the WSN as \( U_i \) at certain moment. For the detailed description of parallel attack, refer to [8, 10, 15].

In Chen-Shih’s scheme [8], the authors added the random nonce \( R \) to DID, and \( C \) to neutralize this attack. However, Yoo et al. [15] found this improvement remains vulnerable. The reason is that secret parameters used by different users are uniform. For instance, \( h(x_n) \) exists in both DID, and DID*, so it can be offset by \( "\oplus" \). In order to resolve the problems resulting from parallel attack, we can consider changing the traditional thinking mode of generating DID, or sacrificing anonymity to achieve discrepancy of parameters for different users.

3. Security Requirements

Sastry and Wagner [16] investigated several issues regarding IEEE 802.15.4 [17], such as ACL management problems (i.e., the same key in multiple ACL entries, loss of ACL state due to power interruptions, key management problems, and insufficient integrity protection), and provided some solutions for these problems. However, the requirements for security authentication protocol in WSNs need to be considered by more appropriate methods in order to resolve the application layer issues, such as mutual authentication, impersonation, replay, parallel session, sinkhole, and wormhole attacks as well as other kinds of sensor node attacks which are described in detail in Section 2.

This section is aimed at discussing the security requirements for two-factor user authentication in WSNs. The proposed security requirements will cover all the known attacks in two-factor user authentication schemes.

3.1. Authentication Security. Authentication security includes not only the mutual authentication between user and GW-node, but also mutual authentication between GW-node and sensor node in WSNs. The main idea of authentication security may be summed up in one sentence: only by stealing all parameters of a participant can adversary impersonate the participant.

The essential idea of mutual authentication mainly has three aspects. As for the user, if the adversary wants to impersonate the user \( U_i \), he/she must obtain the \( U_i \)'s password \( PW_i \) and \( U_i \)'s smart-card simultaneously. Similarly, even if \( U_i \)'s \( PW_i \) and smart-card are compromised, it is difficult to impersonate the GW-node or sensor node. As for the GW-node, we assume that the GW-node is secure, except for privileged-inside attack. Namely, the adversary cannot obtain all parameters of the GW-node directly. However, even if all secret values of every \( U_i \) and sensor nodes are compromised, it is still difficult to impersonate the GW-node. As for the sensor node \( S_n \), it is impossible to impersonate the sensor node \( S_n \) unless \( S_n \) is captured immediately. If other sensors are captured or all secret parameters of other participants in current system are compromised, it is still difficult for attackers to impersonate \( S_n \). The three points above almost cover all attacks in Section 2, such as privileged-inside attack, user impersonation attack, node compromised attack, GW-node impersonation attack, smart-card loss attack, and parallel attack.

3.2. Session Key Security Requirement. It is necessary to establish a session key for translating sensitive data between legal users and sensor nodes. Therefore, semantic security and forward security of session key should be needed in the two-factor user authentication system. More narrowly, semantic security of session keys is guaranteed if an active attacker cannot get any information about the session keys.
shared between the legitimate parties involved. Specifically, the attacker cannot distinguish a real session key from a random one, chosen from the same key space, in the security definition; for more details, see [12]. Forward security of session key means that if long-term secret keys of all the participants are compromised, the secrecy of previously established session keys should not be affected. In the two-factor (password and smart-card) user authentication scheme for WSNs by using symmetric key techniques in this section.

To resist the attacks discussed in Section 2 and meet security requirements discussed in Section 3, we propose a secure user authentication scheme against smart-card loss attack for WSNs, long-term secret keys refer to the participants’ passwords, secret keys in smart-card, and nodes’ master keys.

3.3. Password Security Requirement. Password security requirement can divide into two aspects: offline dictionary attack security and online dictionary attack security. Offline dictionary attack security is also called guessing attack and it means that there is no successful attacker as follows: the attacker intercepts and stores the message and then chooses a candidate password of dictionary in offline mode by using the interception of messages to test whether the candidate password is correct. The process repeats until another candidate password is guessed correctly. One of online dictionary attack securities is to resist undetectable online dictionary attack, and the other is to resist computer program online dictionary attack.

This is how we define password security requirement: except if legal users’ secret values in smart-card are compromised, the external attackers and internal participants will not be able to carry out the offline dictionary attack on these legal users. Namely, it means that it can resist offline password guessing attack discussed in Section 2.3. In addition, adversary may be able to carry out the online attack; however, once the online dictionary attack occurred, the server can detect it rapidly. Namely, it means that it can resist undetectable online dictionary attack and computer program online dictionary attack.

3.4. Password Updating Security Requirement. A password-based user authentication scheme should provide users with a password updating facility so that a user can update his/her password freely.

4. Proposed Scheme

To resist the attacks discussed in Section 2 and meet security requirements discussed in Section 3, we propose a secure user authentication scheme against smart-card loss attack for WSNs by using symmetric key techniques in this section. There are three phases in our scheme: the registration phase, the password updating phase, and the authentication phase.

4.1. Registration Phase. In this phase, user $U_i$ has to submit an identity and password to the GW-node in a secure way. Then, the GW-node issues a license to $U_i$. The detailed steps are depicted as follows.

**Step-R1.** $U_i$ chooses his/her identifier $ID_i$ and password $PW_i$, generates a random number $b_i$ and computes $PW_i^* = h(PW_i || b_i)$. Then $U_i$ sends $ID_i$ and $PW_i^*$ to the GW-node over a secure channel.

**Step-R2.** Upon receiving the registration request, GW-node computes $N_i = h(ID_i || x_i) \oplus PW_i$. Then GW-node stores \{ID$_i$, $N_i$, $h(\cdot)$\} into a smart-card and sends it to the user $U_i$.

**Step-R3.** After receiving the smart-card, the user $U_i$ inputs $b_i$ into it and finishes the registration.

4.2. Password Updating Phase. The password updating phase is needed whenever user $U_i$ wants to update his old password $PW_i$. The detailed process is as follows.

**Step-P1.** User $U_i$ inserts his/her smart-card into the terminal and enters his/her identity $ID_i$, old password $PW_i$, and new password $PW_i^*$ and requests to change the password.

**Step-P2.** $U_i$'s smart-card validates the entered parameter $ID_i$ using the stored value. If $ID_i$ is correct, then $U_i$ is allowed to change the password. Otherwise, the password change phase is terminated.

**Step-P3.** $U_i$'s smart-card calculates $PW_i^* = h(PW_i || b_i)$, and $N_i^* = N_i \oplus PW_i^* \oplus PW_i$ and then replaces the old value $N_i$ with the new value $N_i^*$.

4.3. Authentication Phase. The authentication phase is invoked when $U_i$ wants to perform some query or to access data from the network. The phase is further divided into login and verification phases. Figure 1 shows both the login phase and the verification phase.

(1) **Login Phase.** $U_i$ inserts her/his smart-card to a terminal and keys $ID_i$ and $PW_i$. The smart-card validates $ID_i$ with the stored ones. If the entered $ID_i$ is correct, the smart-card performs the following operations.

**Step-L1.** The smart-card computes $PW_i^* = h(PW_i || b_i)$ and $k = h_1((N_i \oplus PW_i) || T_1)$, where $T_1$ is the current timestamp of $U_i$’s system.

**Step-L2.** The smart-card generates a random number $R_i \in \{0, 1\}^l$ and computes $A_i = E_k(ID_i || R_i || T_1)$, where $E(\cdot)$ is a symmetric key cryptography function; $E_k(\cdot)$ means that “$*$” is encrypted by $E(\cdot)$ with the symmetric key $k$. Then it sends {ID$_i$, $A_i$, $T_1$} to the GW-node.

(2) **Verification Phase.** Upon receiving the login request {ID$_i$, $A_i$, $T_1$} at time $T_1^*$, the GW-node authenticates $U_i$ and creates a session key between $U_i$ and sensor node $S_n$ by the following steps.

**Step-V1.** Validate $T_1$. If $(T_1 - T_1^*) \leq \Delta T$ then the GW-node proceeds to the next step, or else abort, where $\Delta T$ denotes the expected time interval for the transmission delay.

**Step-V2.** The GW-node computes $k = h_1(ID_i || x_i) || T_1^*)$ and $D_k(A_i) = \{ID_i, R_i, T_1\}$, where $D_k(\cdot)$ means that “$*$” is decrypted by the symmetric key $k$. Then the GW-node checks whether the decrypted messages $ID_i$ and $T_1$ are equal to the ones received. If it is true, the GW-node establishes trust on the $U_i$ and proceeds to the next step; otherwise abort.
**Step-V3.** The GW-node computes $SK = h_1(ID_1 \parallel h(x_s) \parallel SID_n) \parallel T_2$ and $B_i = h_1(h(x_s) \parallel SID_n) \parallel SK \parallel SID_n \parallel ID_1 \parallel T_2$, where $T_2$ is the current timestamp of GW-node's system and SK is the session key between user $U_i$ and $S_n$ in information transport. Then the GW-node sends the message \{ID_1, B_i, T_2\} to the sensor node $S_n$.

**Step-V6.** Upon receiving the message from the sensor node $S_n$, the GW-node first validates $T_2$ in a similar line of Step-V1. $S_n$ computes the session key $SK = h_1(ID_1 \parallel h(x_s) \parallel SID_n) \parallel T_2$. $S_n$ computes $B_i^* = h_1(h(x_s) \parallel SID_n) \parallel SK \parallel SID_n \parallel ID_1 \parallel T_2$ and then checks whether $B_i^* = B_i$. If it is true, the sensor node $S_n$ establishes trust on the GW-node or else rejects it.

**Step-V7.** The GW-node computes $D_i = E_k(ID_1 \parallel SID_n \parallel SK \parallel R_1 \parallel T_4)$, where $T_4$ is the current timestamp of GW-node's system; then GW-node sends the message \{D_i, T_4\} to user $U_i$.

**Step-V8.** Upon receiving the message from the GW-node, $U_i$ first validates $T_4$ in a similar line of Step-V1. $U_i$ computes $D_k(D_i) = \{ID_1, SID_n, SK, R_1, T_3\}$; then $U_i$ checks whether the decrypted messages $ID_1, R_1, T_4$ are equal to the previous ones. If it is true, user $U_i$ establishes trust on the GW-node and establishes a session key SK with the sensor node $S_n$; then he/she can enjoy the data from the WSN; otherwise stop accessing to the sensor network.

### 5. Scheme Analysis

In this section, based on adversary's capabilities assumptions in Section 2, we carry out a security analysis of our scheme. Then efficiency analysis is followed (computational cost and communication cost). Finally, we give a performance comparison with existing schemes.

#### 5.1. Security Analysis

**Authentication Security.** Our scheme provides authentication security, where all entities can authenticate each other.
Further, no one can impersonate the participant unless the adversary steals all parameters of the participant. More details are as follows.

We assume that the GW-node is secure; namely, adversary cannot get the values of \( x_a \) and \( x_i \) from GW-node directly. Our scheme is passing \( PW_i = h(PW_i \parallel b) \) instead of the plain password to resist the privileged-insider attack. Furthermore, if all secret values of every \( U_j \) and sensor nodes are compromised, it is still difficult to impersonate the GW-node. Because the compromised value of \( h(x_i \parallel SID_i) \) in \( S_i \) cannot affect \( x_i \) and smart-card just stores \( N_i = h(ID_i \parallel x_i) \oplus PW_i \) about GW-node, attackers cannot recover \( x_a \) from it.

It is also difficult to impersonate \( S_a \) except that \( S_a \) is captured immediately because the secret values are unique in different sensor nodes by binding with sensor nodes' identity. Moreover, the compromised sensor node cannot affect other participants including the users and GW-node.

In our scheme, we solved the open question which was pointed out in [11]; namely, our scheme is secure merely by using symmetric key techniques when the smart-card is stolen. If the adversary wants to impersonate user \( U_j \), he/she must obtain the user's password \( PW_i \) and smart-card simultaneously. This is because that smart-card only stores \( \{b, N_i, h(\cdot)\} \) and uses online validation which is different from Sun's scheme [11]. It is known that offline validation with symmetric cryptography could cause guessing attack that we have discussed in Section 2.6. Moreover, offline password guessing attack also exists in online validation (Sun pointed out that his proposed scheme is insecure in smart-card loss case [11]). But in our scheme, \( N_i = h(ID_i \parallel x_i) \oplus PW_i \) stored in smart-card is used to generate symmetric cryptography key \( k = h((N_i \oplus PW_i) \parallel T_1) \), and \( A_1 = E_k(ID_i \parallel R_i \parallel T_1) \) is transmitted to GW-node, where \( R_i \in \{0, 1\}^l \). It is deserved to note that the function of the random number \( R_i \) prevents exhaust attacks, but without freshness. In this way, adversary cannot recover user's password \( PW_i \) from smart-card through offline password guessing attack. What is more, our scheme only uses symmetric key techniques. Similarly, even if \( U_j \)'s \( PW_i \) and smart-card are both compromised, it is difficult to impersonate the GW-node or sensor node.

Based on the analysis above, our scheme meets authentication security. This means that our scheme can resist all attacks in Section 2, such as user impersonation attack, node compromised attack, GW-node impersonation attack, smart-card loss attack, and parallel attack.

**Session Key Security Requirement.** Our proposed scheme establishes a session key \( SK = h_1(ID_i \parallel h(x_i \parallel SID_i) \parallel T_2) \) for translating sensitive data between legal users and sensor nodes. Obviously, the session key \( SK \) has semantic security and forward security.

**Password Security Requirement.** In accordance with the above analysis, the proposed scheme can resist privileged-inside attack. In addition, according to the analysis above, the external attackers and internal participants are unable to carry out the offline dictionary attack on legal users even if legal users' secret values in smart-card are compromised. Online dictionary attack also can be avoided. So, our scheme meets password security requirement.

**Password Updating Security Requirement.** A password updating facility is provided for users in our scheme (see Section 4.2).

From Table 1, it is easy to see that the proposed scheme has more security functionality as compared with other existing protocols for two-factor user authentication in WSNs. In Table 1, “usual” denotes the common attacks such as replay attack or stolen-verifier attack, “RA” denotes relay attack, “PIA” denotes privileged-insider attack, “ULA” denotes user impersonation attack, “GA” denotes guess attack, “NCA” denotes node compromise attack, “GIA” denotes GW-node impersonation attack, “SRA” denotes smart-card loss attack, and “PA” denotes parallel attack. “Y” denotes the scheme that can resist this attack and “N” denotes the scheme that cannot resist this attack.

### 5.2. Performance Analysis.

In this subsection, we compare our protocol with related ones in terms of computational cost in the registration/authentication phase and communication cost in the message exchange phase. Because these two phases are the main procedures of an authentication protocol, let us define “H” as performing one-way hash function, “Pub/Pri” as public/private-key computation, “Se/Sd” as symmetric key encryption/decryption, “MAC” as message authentication code (MAC) function computation, “PM/PA” as elliptic curve point multiplication/addition computation, “E” as elliptic curve polynomial computation, and “NME” as the numbers of message exchanges in authentication phase. The result is shown in Table 2.

**Computational Cost.** The computational cost for user registration is onetime job for certain period of time. The user and the GW-node in our scheme only require 1 hash operation, respectively. And computational cost for user authentication is prime concern. Our sensor node requires only 3 hash operations with ensuring security. Although its computation cost just requires one or two more hash operations than some schemes [1, 4–6, 9, 11, 14], it is highly efficient compared with other schemes [10, 18, 19]. It is also desirable that the user and the GW-node for user authentication require symmetric encryption operation Se/Sd, and our scheme does not require Pub/Pri [9], PA/PM, E [10, 18], and MAC [19], which need additional storage and computations. Sequently, all operations added to protect query responses can be implemented using a block cipher such as Advanced Encryption Standard (AES) [20]. Also, note that most of the recently developed nodes in WSNs, for example, T mote, TelosB, and Micaz, already have a built-in AES module and, thus, no additional hardware is required [21]. Therefore, our scheme is well-suited to the resource-limited sensor node. Compared with previous schemes which have too many vulnerabilities on security, our proposal achieves stronger security without loss of immense effectiveness. In conclusion, our scheme is also practical for the real world applications in enhancing the security over wireless communications.
Table 1: Functionality comparison of our scheme with existing schemes.

| Scheme/attack | Usual | PIA | UIA | GA | NCA | GIA | SRA | PA | SK |
|---------------|-------|-----|-----|----|-----|-----|-----|----|----|
| Das’s [1]     | Y     | N   | N   | N  | N   | N   | N   | N  | N  |
| Nyang-Lee’s [2]| Y     | N   | N   | N  | N   | N   | N   | N  | Y  |
| He’s [5]      | Y     | Y   | N   | N  | N   | N   | N   | N  | N  |
| Huang’s [14]  | Y     | N   | N   | N  | Y   | N   | N   | N  | N  |
| Khan and Alghathbar’s [4]| Y | N | N | N | N | N | N | N | N |
| Sun’s [11]    | Y     | Y   | Y   | N  | Y   | Y   | N   | Y  | N  |
| Yuan’s [9]    | Y     | N   | N   | N  | N   | N   | N   | N  | Y  |
| Vaida’s [6]   | Y     | N   | N   | N  | N   | N   | N   | N  | N  |
| Chen-Shih’s [8]| Y     | N   | N   | N  | N   | N   | N   | N  | N  |
| Li et al.’s [13]| Y     | N   | Y   | Y  | N   | Y   | N   | Y  | N  |
| Yoo’s [15]    | Y     | Y   | N   | N  | Y   | Y   | N   | Y  | N  |
| RUASN’s [19]  | Y     | N   | N   | N  | Y   | Y   | N   | Y  | Y  |
| Yeh’s [10]    | Y     | Y   | N   | N  | Y   | Y   | N   | Y  | Y  |
| Shi’s [18]    | Y     | Y   | N   | N  | Y   | Y   | N   | Y  | Y  |
| Ours          | Y     | Y   | Y   | Y  | Y   | Y   | Y   | Y  | Y  |

Table 2: Performance comparison among related protocols.

| Phase/scheme | Registration | Login/authentication | Sensor node | NME |
|--------------|--------------|----------------------|-------------|-----|
| User         | GW-node      | User                 | GW-node     |     |
| Das’s [1]    | —            | 3H                   | 4H          | 4H  | 2H  |
| Nyang-Lee’s [2]| —        | 3H                   | H + Sd      | 9H + Se | 2H + Se + Sd | 3 |
| He’s [5]     | H            | 5H                   | 5H          | 5H  | H   | 2 |
| Huang’s [14] | —            | 4H                   | 4H          | 6H  | H   | 2 |
| Khan and Alghathbar’s [4]| H | 2H | 4H | 5H | 2H | 3 |
| Sun’s [11]   | —            | 2H                   | 2H          | 5H  | 2H  | 5 |
| Yuan’s [9]   | —            | 5H                   | 7H + Pub    | 7H + Pri | 2H  | 4 |
| Vaida’s [6]  | H            | 4H                   | 6H          | 5H  | 2H  | 3 |
| Chen-Shih’s [8]| —        | 3H                   | 5H          | 5H  | H   | 4 |
| RUASN’s [19] | H            | 3H + Se              | 4H + Se + Sd| 4H + Se + Sd + MAC | H + Se + Sd + MAC | 4 |
| Yeh’s [10]   | H            | 3H + PM              | 4H + PA + 2PM| 4H + 2PA + 3PM + E | 3H + PA + PM + E | 3 |
| Shi’s [18]   | H            | 2H + PM              | 5H + 3PM    | 4H + PM | 3H + 2PM | 4 |
| Ours         | H            | H                    | 2H + Se + Sd| 6H + Se + Sd | 3H  | 4 |

**Communication Cost.** In WSN, without constant supply of power, sensor nodes are resource-limited. In addition, the most energy-intensive element of sensor is wireless communication module, namely, receiving, sending, and so on. Therefore, the number of message exchanges is crucial. Our scheme only needs four message exchanges to achieve all security features; however, many schemes which need less number of message exchanges have various vulnerabilities in security. In Sun’s scheme [11], it uses nonce instead of timestamp to reduce the trouble of synchronized time clocks, but it needs one additional message and also has security flaws. Obviously, compared with Sun’s scheme our scheme is well-suited to the resource-limited sensor node.

Considering computational cost and communication cost, it is clear that the proposed scheme not only is an efficient scheme with high reliability, but also is practical for real-time applications.

6. Conclusion and Future Work

In real-time, access control is an imperative requirement for WSNs to protect the data access from unauthorized parties. However, there is no appropriate formal model to examine the security of the two-factor user authentication scheme for WSNs. In addition, previous schemes failed to resist the smart-card loss attack merely by using symmetric key techniques. Therefore, designing a genuine two-factor user authentication scheme which can resist guessing attack caused by smart-card loss attack is meaningful for the application.

In this regard, we summarize the main attacks and security requirements for two-factor user authentication in WSNs in this paper. A user authentication and session key establishment scheme based on security requirements is also proposed, which can resist smart-card loss attack merely by
using symmetric key techniques. The proposed scheme solves the open question pointed out in [11]. We have shown a security analysis and performance analysis of our proposal and carried out a comparison with existing schemes. Analysis shows that the proposed scheme is more secure and highly efficient. Therefore, it is very suited to WSNs environments.

We highlight two areas for our future work. For the attacks and security requirements discussed in this paper, there is no appropriate formal model to examine the security of the two-factor user authentication scheme for WSNs. One direction of our future work will be to present a formal definition of the two-factor user authentication scheme under WSNs setting and construct new scheme in the given definition. Moreover, the proposed scheme cannot provide the user privacy. That is, the adversary can trace the target user by observing the user authentication session. Thus, the other future work is to devise the anonymous user authentication scheme for WSNs.

Notations and Symbols

| Symbol | Description |
|--------|-------------|
| $U_i$  | A certain user |
| $ID_i$ | $U_i$’s identifier |
| $PW_i$ | $U_i$’s password |
| DID | $U_i$’s dynamic login identifier |
| GW-node | Gateway-node of WSN |
| $K$ | The symmetric key known to only GW-node |
| $x_n$ | The secret parameter generated securely by the GW-node, which is shared with the GW-node and $U_i$ |
| $b$ | A high-entropy random number selected by user, not revealed to the GW-node |
| $S_n$ | A certain sensor node |
| SID | $S_n$’s identifier |
| $x_s$ | The shared key between the GW-node and sensor node which is generated securely by the GW-node and stored in sensor node |
| $h(x_s, SID_n)$ | The secret key instead of $x_s$ stored in sensor node, which is shared with the GW-node and sensor node, where $x_s$ is secret parameter generated securely by the GW-node |
| $T_i$ | The timestamps, such as $T_1$ and $T_2$ |
| $R_i$ | Cryptographic random numbers or the nuances, such as $R_1$ and $R_2$ |
| $h()$, $h_1()$ | The cryptographic hash functions |
| $\|$ | The concatenation operator |
| $@$ | The bitwise exclusive or or operator |

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61309016, 61379150, and 61103230), Postdoctoral Science Foundation of China (no. 2014M562493), the National Cryptology Development Project of China (no. MMJ201201004), Fundamental Research Funds for the Central Universities (no. JB140302), and the funding of Science and Technology on Information Assurance Laboratory (KJ-13-02).

References

[1] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.
[2] D. Nyang and M. K. Lee, "Improvement of Das’s two-factor authentication protocol in wireless sensor networks," IACR Cryptology ePrint Archive, 2009.
[3] M. K. Khan and K. Alghathbar, "Security analysis of ‘two-factor user authentication in wireless sensor networks,” in *Advances in Computer Science and Information Technology*, vol. 6059 of Lecture Notes in Computer Science, pp. 55–60, Springer, Berlin, Germany, 2010.
[4] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of ‘two-factor user authentication in wireless sensor networks,” *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.
[5] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.
[6] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Improved two-factor user authentication in wireless sensor networks,” in *Proceedings of the 6th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob ’10)*, pp. 600–606, October 2010.
[7] R. Fan, D.-J. He, X.-Z. Pan, and L.-D. Ping, "An efficient and DoS-resistant user authentication scheme for two-tiered wireless sensor networks,” *Journal of Zhejiang University: Science C*, vol. 12, no. 7, pp. 550–560, 2011.
[8] T.-H. Chen and W.-K. Shih, "A robust mutual authentication protocol for wireless sensor networks,” *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.
[9] J.-J. Yuan, "An enhanced two-factor user authentication in wireless sensor networks,” *Telecommunication Systems*, vol. 55, no. 1, pp. 105–113, 2014.
[10] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using Elliptic Curves Cryptography,” *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.
[11] D.-Z. Sun, J.-X. Li, Z.-Y. Feng, Z.-F. Cao, and G.-Q. Xu, "ON the security and improvement of a two-factor user authentication scheme in wireless sensor networks,” *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 895–905, 2013.
[12] M. Bellare and P. Rogaway, “Entity authentication and key distribution,” in *Advances in Cryptology—CRYPTO’93*, vol. 773 of Lecture Notes in Computer Science, pp. 232–249, Springer, Berlin, Germany, 1994.
[13] C.-T. Li, C.-C. Lee, L.-J. Wang, and C.-J. Liu, "A secure billing service with two-factor user authentication in wireless sensor networks,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 8, pp. 4821–4831, 2011.
[14] H.-F. Huang, Y.-F. Chang, and C.-H. Liu, "Enhancement of two-factor user authentication in wireless sensor networks,” in *Proceedings of the 6th International Conference on Intelligent
10

International Journal of Distributed Sensor Networks

Information Hiding and Multimedia Signal Processing (IIHMSP ’10), pp. 27–30, October 2010.

[15] S. G. Yoo, K. Y. Park, and J. Kim, “A security-performance-balanced user authentication scheme for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2012, Article ID 382810, II pages, 2012.

[16] N. Sastry and D. Wagner, “Security considerations for IEEE 802.15.4 networks,” in Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe ’04), pp. 32–42, ACM, October 2004.

[17] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, “IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks,” IEEE Network, vol. 15, no. 5, pp. 12–19, 2001.

[18] W. Shi and P. Gong, “A new user authentication protocol for wireless sensor networks using elliptic curves cryptography,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 730831, 7 pages, 2013.

[19] P. Kumar, A. J. Choudhury, M. Sain, S.-G. Lee, and H.-J. Lee, “RUASN: a robust user authentication framework for wireless sensor networks,” Sensors, vol. 11, no. 5, pp. 5020–5046, 2011.

[20] National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, 2001.

[21] Texas Instruments, CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, 2006.