Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 382810, 11 pages
doi:10.1155/2012/382810

Research Article

A Security-Performance-Balanced User Authentication Scheme for Wireless Sensor Networks

Sang Guun Yoo, Keun Young Park, and Juho Kim

Department of Computer Science and Engineering, Sogang University, Seoul 121-742, Republic of Korea

Correspondence should be addressed to Juho Kim, jhkim@sogang.ac.kr

Received 3 November 2011; Revised 20 January 2012; Accepted 12 February 2012

Academic Editor: Wensheng Zhang

The uses of wireless sensor networks have increased to be applicable in many different areas, such as military applications, ecology, and health applications. These applications often include the management of confidential information, making the issue of security one of the most important aspects to consider. In this aspect, a user authentication mechanism that allows only legitimate users to access the network data becomes critical for maintaining the confidentiality and integrity of the network information. In this paper, we describe and cryptoanalyze previous works in user authentication to illustrate their vulnerabilities and security flaws. We then propose a robust user authentication scheme that solves the identified limitations. Additionally, we describe how the proposed protocol is more suitable for a secure sensor network implementation by analysis in terms of security and performance.

1. Introduction

Wireless sensor networks (WSNs) are being applied in different fields such as habitat monitoring [1], indoor sensor networks [2], military applications [3], and health monitoring [4]. Many of these applications manage confidential information, making the issue of security one of the most important points to consider. One of the fields of research in wireless sensor network security is the user authentication scheme that allows only authentic users to access the data collected by the sensor nodes.

In 2006, Wong et al. [5] proposed a dynamic user authentication scheme and discussed the implementation issues with the recommendation of using the security features of the IEEE 802.15.4 MAC sublayer. Later, in 2009, Das [6] presented his research work where he identified vulnerabilities in Wong et al.'s protocol and proposed his own authentication scheme based on the two-factor user authentication concept. After publication of Das' proposal, several works have pointed out that such a protocol was vulnerable to other attacks. Nyang and Lee [7] identified that Das' protocol was vulnerable to offline password guessing and sensor node compromising attacks. Huang et al. [8] also identified some limitations of Das' scheme, such as vulnerability from an impersonation attack. Additionally, [9] pointed out the absence of a mutual authentication feature in Das' protocol, while Khan and Alghathbar [10] pointed out more security flaws of Das' proposal, noting that it was vulnerable to privileged-insider and gateway-node bypassing attacks. References [7–10] also proposed enhanced versions of Das' protocol to eliminate detected vulnerabilities. However, those proposals still include several vulnerabilities and limitations that an adversary could take advantage of.

In this paper, we provide two specific contributions to the WSN user authentication research area: (1) first, we cryptoanalyze the aforementioned works and show how reference [7] is still vulnerable to parallel session and privileged-insider attacks and how it does not offer a password change mechanism. We also illustrate how [8] is vulnerable to parallel session and privileged-insider attacks and how it does not provide mutual authentication and password change features. Additionally, we explain how [9] is defenseless against parallel session, privileged-insider, and gateway-node bypassing attacks, does not offer a password change mechanism, and has a serious vulnerability in its mutual authentication mechanism. Furthermore, we explain

how [10] is vulnerable from parallel session attacks, only offers a partial protection against gateway-node bypassing attacks, and does not provide mutual authentication between the user and the gateway-node. (2) Later, after identifying the limitations of previously mentioned works, we propose a robust user authentication for wireless sensor networks which fixes the aforementioned weaknesses.

The rest of the paper is organized as follows. Section 2 briefly reviews the existing works and details the weaknesses and security pitfalls of such schemes. Section 3 then presents the proposed protocol which solves the vulnerabilities and limitations mentioned in Section 2. Next, Section 4 analyzes the proposed protocol in terms of security and performance. Finally, Section 5 concludes this paper.

2. Previous Works and Their Cryptanalysis

In this section, we explain briefly the proposal of Das [6]. We then describe further works [7–10] focused to solve the limitations of Das' scheme and how those enhanced proposals are still not secure and have several security vulnerabilities.

2.1. Review of the Das Scheme. The scheme proposed by Das [6] is composed of registration and authentication phases.

Registration Phase. A user U_i submits his/her identity ID_i and password PW_i to the gateway node GW using a secure channel. GW then computes $N_i = h(ID_i \| PW_i) \bigoplus h(K)$, where K is a symmetric key only known by GW, $h(\cdot)$ is a hash function, and "$\|$" is a concatenation operator. Once N_i is calculated, GW personalizes a smart card with the parameters $h(\cdot)$, ID_i, N_i, $h(PW_i)$, and x_a, where x_a is a secret parameter generated securely by GW and stored in the sensor nodes before deployment. Finally, GW delivers the smart card to U_i in a secure manner.

Authentication Phase. This phase is executed when U_i needs to access data of a sensor node of the network. The phase is composed of the Login and verification phases.

(1) *Login Phase.* U_i inserts the smart card in his/her terminal, and inputs ID_i and PW_i. The smart card verifies the validity of those values by comparing ID_i and $h(PW_i)$ with the data stored in it. If those values are correct, the smart card computes $DID_i = h(ID_i \| PW_i) \bigoplus h(x_a \| T)$ and $C_i = h(N_i \| x_a \| T)$, where T is the current timestamp of U_i's system and sends $\{DID_i, C_i, T\}$ to GW.

(2) *Verification Phase.* Upon receiving the login request at time T^*, GW validates T. If $(T^* - T) > \Delta T$, then GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW computes $h(ID_i \| PW_i)^* = DID_i \bigoplus h(x_a \| T)$ and $C_i^* = h((h(ID_i \| PW_i)^* \bigoplus h(K)) \| x_a \| T)$. If C_i^* is different to C_i, then GW rejects the login request; otherwise, GW sends a message $\{DID_i, A_i, T'\}$ to some nearest sensor node S_n to respond to the query with the data that U_i is looking for, where $A_i = h(DID_i \| S_n \| x_a \| T')$ and T' is the current timestamp of the GW's system. S_n first validates T', then computes $h(DID_i \| S_n \| x_a \| T')$, and checks whether it is equal to A_i. If those values match, then S_n responds to U_i query.

2.2. Chen-Shih's Scheme. In [9], the authors indicate that Das' scheme fails in mutual authentication and it is vulnerable to parallel session attack, and propose "A robust mutual authentication protocol for wireless sensor networks" which still has vulnerabilities and limitations.

2.2.1. Review of Chen-Shih's Scheme. The protocol proposed in [9] is composed of registration, login, verification, and mutual authentication phases.

Registration Phase. The user U_i submits his/her identity ID_i and password PW_i to the gateway node GW using a secure channel. GW then computes $N_i = h(ID_i \| PW_i) \bigoplus h(K)$, where K is a symmetric key only known by GW, $h(\cdot)$ is a hash function, and "$\|$" is a concatenation operator. Once N_i is calculated, GW personalizes a smart card with the parameters $h(\cdot)$, ID_i, N_i, $h(PW_i)$, and x_a, where x_a is a secret parameter generated by GW and stored in the sensor nodes before deployment. Finally, GW delivers the smart card to U_i in a secure manner.

Login Phase. When U_i enters his/her ID_i and PW_i, the smart card verifies the validity of ID_i and PW_i. If they are not correct, it terminates the request; otherwise, U_i's smart card generates a random nonce R_i at T_u and computes $DID_i = h(ID_i \| PW_i) \bigoplus h(x_a \| T_u \| R_i)$ and $C_i = h(N_i \| x_a \| T_u \| R_i)$, where T_u is the current timestamp of U_i's system. U_i then sends the message $\{DID_i, C_i, T_u, R_i\}$ to GW.

Verification Phase. Once the $\{DID_i, C_i, T_u, R_i\}$ message is received at time T_g, GW verifies if $(T_g - T_u) > \Delta T$, where ΔT denotes the maximum allowed delay. If the required condition is fulfilled, GW aborts the authentication process; otherwise, GW computes $h(ID_i \| PW_i)^* = DID_i \bigoplus h(x_a \| T_u \| R_i)$ and $C_i^* = h((h(ID_i \| PW_i)^* \bigoplus h(K)) \| x_a \| T_u \| R_i)$. If C_i^* is different to C_i, GW rejects the login request; otherwise, GW accepts the request and generates a random nonce R_c and sends a message $\{DID_i, A_i, T'\}$ to some sensor node S_n, where $A_i = h(DID_i \| S_n \| x_a \| T')$ and T' is the current timestamp of GW's system. Additionally, GW also sends the $\{C_g, R_c, S_n\}$ message to U_i where $C_g = h(DID_i \| S_n \| x_a \| R_c)$. Finally, S_n, after validating T', computes $h(DID_i \| S_n \| x_a \| T')$ and checks whether it is equal to A_i. If those values match, then S_n responds to the query from U_i.

Mutual Authentication Phase. After receiving the message $\{C_g, R_c, S_n\}$, U_i only coworks with S_n if C_g is equal to $h(DID_i \| S_n \| x_a \| R_c)$.

2.2.2. Cryptoanalysis of Chen-Shih's Scheme. Here, we show how the proposal of Chen and Shih still has some critical security pitfalls and limitations.

Parallel Session Attack. In Chen-Shih's scheme, the authors include the random nonce R_i inside DID_i and C_i to neutralize this attack. However, their protocol remains vulnerable. Assume that a legal user Tom eavesdrops on the message $\{DID_i, C_i, T_1, R_{i1}\}$ between GW and U_i at timestamp T_1 to obtain $DID_{i(T1)} = h(ID_i \| PW_i) \bigoplus h(x_a \| T_1 \| R_{i1})$ and R_{i1}, where $DID_{i(T1)}$ is the DID_i value at T_1 and R_{i1} denotes the random nonce at T_1. Tom can then forge the message DID_i at timestamp T_2 $DID_{i(T2)}$ by generating $DID_{\mathrm{Tom}(T1)} = h(ID_{\mathrm{Tom}} \| PW_{\mathrm{Tom}}) \bigoplus h(x_a \| T_1 \| R_{i1})$ and $DID_{\mathrm{Tom}(T2)} = h(ID_{\mathrm{Tom}} \| PW_{\mathrm{Tom}}) \bigoplus h(x_a \| T_2 \| R_{i2})$, and computing $DID_{i(T2)} = DID_{i(T1)} \bigoplus DID_{\mathrm{Tom}(T1)} \bigoplus DID_{\mathrm{Tom}(T2)} = h(ID_i \| PW_i) \bigoplus h(x_a \| T_1 \| R_{i1}) \bigoplus h(ID_{\mathrm{Tom}} \| PW_{\mathrm{Tom}}) \bigoplus h(x_a \| T_1 \| R_{i1}) \bigoplus h(ID_{\mathrm{Tom}} \| PW_{\mathrm{Tom}}) \bigoplus h(x_a \| T_2 \| R_{i2})$, where R_{i2} is any random number selected by Tom, and ID_{Tom} and PW_{Tom} are Tom's ID and password, respectively. Once U_i's $DID_{i(T2)}$ is obtained, Tom can send a new session message $\{DID_{i(T2)}, C_i, T_2, R_{i2}\}$ at T_2 for a new login request.

Gateway Node Bypassing Attack. The Chen-Shih scheme uses the x_a value to allow S_n to verify that the A_i message originates from the authentic GW. If we assume that the adversary can extract the value of x_a stored inside of a valid smart card by using some techniques [11–13], the adversary can execute the gateway node bypassing attack. First, the attacker computes a forged $DID_f = h(ID_f \| PW_f) \bigoplus h(x_a \| T_f \| R_f)$ by using the extracted x_a, where ID_f is a forged ID, PW_f is a randomly chosen forged password, T_f is the timestamp of an adversary's terminal, and R_f is an arbitrary random nonce. The attacker then computes $A_f = h(DID_f \| S_n \| x_a \| T_f)$. Once DID_f and A_f are calculated, the adversary sends the message $\{DID_f, A_f, T_f\}$ to S_n over the public channel. Finally, S_n authenticates the adversary's message because S_n cannot recognize its invalidity because the $h(DID_f \| S_n \| x_a \| T_f)$ value computed by S_n is equal to the A_f value received from the adversary.

Privileged-Insider Attack. The system administrator or privileged-insider of GW may try to impersonate U_i by authenticating himself/herself to other servers where U_i could be registered user. This is possible because GW receives the password of U_i in plaintext, that is, PW_i, in the registration phase, and because many users use the same password to access different applications of servers.

Vulnerable Mutual Authentication between U_i and GW. Chen-Shih's scheme proposes a mutual authentication phase. However, it has vulnerability which allows an adversary to execute the GW spoofing attack. If we assume that the adversary can extract the value of x_a from a valid smart card or sensor node by using some techniques [11–13] as assumed in [8, 10], the adversary can then pretend to be a valid GW. First, the fake gateway node GW_f listens to the network and sniffs the $\{DID_i, C_i, T_u, R_i\}$ message. Once the message is received, GW_f can respond with the message $\{C_f, R_f, S_f\}$, where $C_f = h(DID_i \| S_f \| x_a \| R_f)$, where S_f is the identification of the adversary's sensor node and where R_f is a random nonce selected by the adversary. Finally, U_i authenticates the adversary's message because U_i cannot recognize its invalidity because the received C_f is equal to $h(DID_i \| S_f \| x_a \| R_f)$ computed by U_i. After authentication, the fake sensor node S_f can send false data to U_i.

Lack of Mutual Authentication between GW and the Sensor Node. The Chen-Shih scheme does not provide a mutual authentication mechanism between GW and the sensor nodes. Therefore, it is vulnerable to a sensor node spoofing attack. The adversary can place a false sensor node S_f to respond to the $\{DID_i, A_i, T\prime\}$ message with false data. GW cannot recognize the invalidity of the false data because it does not perform any verification.

Lack of a Password Change Phase. The Chen-Shih scheme does not provide a password change phase for U_i, which is a requirement for a secure system.

2.3. Khan-Alghathbar's Scheme. In [10], the authors indicate that Das' scheme is vulnerable to gateway node bypassing and privileged-insider attacks. They also point out that Das' scheme does not provide mutual authentication or a password change mechanism. As a response to such limitations, they propose improvements of Das' scheme which still has vulnerabilities and limitations.

2.3.1. Review of Khan-Alghathbar's Scheme. The protocol proposed in [10] is composed of registration and authentication phases.

Registration Phase. A user U_i submits his/her identity ID_i and password PW_i to his/her terminal. The terminal then calculates $h(PW_i)$ and sends ID_i and $h(PW_i)$ to the gateway node GW using a secure channel, where $h(\cdot)$ is a hash function. GW then computes $N_i = h(ID_i \| h(PW_i)) \bigoplus h(K)$, where K is a symmetric key only known by GW and "$\|$" is a concatenation operator. Once N_i is calculated, GW personalizes a smart card with the parameters $h(\cdot)$, ID_i, N_i, $h(PW_i)$, and x_a, where x_a is a secret parameter generated securely by GW. On the other hand, GW generates another secret parameter x_s and stores it in each sensor node before its deployment in the field.

Authentication Phase. This phase is executed when U_i needs to access the data of a sensor node of the network. The phase is composed of login and verification phases.

(1) *Login Phase.* U_i inserts the smart card in his/her terminal, and inputs ID_i and PW_i. The smart card verifies the validity of those values by comparing the data stored in it. If those values are correct, the smart card computes $DID_i = h(ID_i \| PW_i) \bigoplus h(x_a \| T)$ and $C_i = h(N_i \| x_a \| T)$, where T is the current timestamp of U_i's system and sends $\{DID_i, C_i, T\}$ to GW. Otherwise, the login request is rejected.

(2) *Verification Phase.* Upon receiving the login request at time T^*, GW validates T. If $(T^* - T) > \Delta T$, GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW

computes $h(ID_i\|PW_i)^* = DID_i \bigoplus h(x_a\|T)$ and $C_i^* = h((h(ID_i\|PW_i)^* \bigoplus h(K))\|x_a\|T)$. If C_i^* is different to C_i, GW rejects the login request. Otherwise, GW sends a message $\{DID_i, A_i, T'\}$ to some nearest sensor node S_n, where $A_i = h(DID_i\|S_n\|x_s\|T')$ and T' is the current timestamp of GW's system. S_n first validates T', then computes $h(DID_i\|S_n\|x_s\|T')$, and checks whether it is equal to A_i. If those values match, S_n computes $B_i = h(S_n\|x_s\|T'')$, where T'' is the current timestamp of sensor node's system and sends $\{B_i, T''\}$ to GW. After receiving the mutual authentication message $\{B_i, T''\}$, GW first checks the validity of timestamp T'' and then computes $B_i^* = h(S_n\|x_s\|T'')$ and checks whether it is equal to B_i. If those values match, GW establishes trust with the sensor node; otherwise, GW alerts U_i about the possibility of a malicious sensor node in the network and sends a process-termination message.

Password Change Phase. When a user U_i wants to change his/her password PW_i to a new password PW_i^*, U_i inserts his/her smart card into the terminal and enters ID_i, PW_i, and PW_i^*. The smart card validates ID_i and PW_i. Only if those values are correct, then the smart card computes $N_i^* = N_i \bigoplus h(ID_i\|PW_i) \bigoplus h(ID_i\|PW_i^*)$ and replaces N_i and $h(PW_i)$ with N_i^* and $h(PW_i^*)$, respectively.

2.3.2. Cryptoanalysis of Khan-Alghathbar Scheme. The proposal of Khan and Alghathbar still has some critical security pitfalls and limitations as shown below.

Parallel Session Attack. Assume that a legal user Tom eavesdrops on the message $\{DID_i, C_i, T_1\}$ between GW and U_i at timestamp T_1 to obtain the DID_i at T_1 $DID_{i(T1)} = h(ID_i\|PW_i) \bigoplus h(x_a\|T_1)$. Tom then can forge the DID_i at timestamp $T_2 DID_{i(T2)}$ by generating $DID_{\text{Tom}(T1)} = h(ID_{\text{Tom}}\|PW_{\text{Tom}}) \bigoplus h(x_a\|T_1)$ and $DID_{\text{Tom}(T2)} = h(ID_{\text{Tom}}\|PW_{\text{Tom}}) \bigoplus h(x_a\|T_2)$, then computing $DID_{i(T2)} = DID_{i(T1)} \bigoplus DID_{\text{Tom}(T1)} \bigoplus DID_{\text{Tom}(T2)} = h(ID_i\|PW_i) \bigoplus h(x_a\|T_1) \bigoplus h(ID_{\text{Tom}}\|PW_{\text{Tom}}) \bigoplus h(x_a\|T_1) \bigoplus h(ID_{\text{Tom}}\|PW_{\text{Tom}}) \bigoplus h(x_a\|T_2)$, where ID_{Tom} and PW_{Tom} are Tom's ID and password, respectively. Once $DID_{i(T2)}$ is obtained, Tom can then send a new session message $\{DID_{i(T2)}, C_i, T_2\}$ at T_2 for a new login request.

Gateway Node Bypassing Attack. The secret value x_s stored and shared by sensor nodes can be extracted using similar techniques of extracting x_a from a smart card [11, 13]. If x_s is extracted, the adversary can execute the gateway node bypassing attack using $DID_f = h(ID_f\|PW_f) \bigoplus h(x_a\|T_f)$ and $A_f = h(DID_f\|S_n\|x_s\|T_f)$, where ID_f is a forged ID, PW_f is a randomly chosen forged password, and T_f is the timestamp of adversary's terminal.

Lack of Mutual Authentication between U_i and GW. First of all, the aforementioned work proposes a mutual authentication between GW and S_n. However, they omit the mutual authentication between U_i and GW. Newer sensor networks offer remote administration/query features in gateway nodes [14, 15] allowing users to access to network's data from a remote terminal. In this kind of environment, it is really important to authenticate the validity of GW from the U_i's side to avoid adversaries collecting valuable data using fake gateway nodes.

2.4. Nyang-Lee's Scheme. In [7], the authors point out that Das' scheme is vulnerable to password guessing attacks and gateway node impersonation attacks and has the limitation of a lack of protection relating to query-response. As a response to such security pitfalls, the authors propose a security-enhanced protocol.

2.4.1. Review of Nyang-Lee's Scheme. In [7], the authors propose a security-enhanced protocol composed of the registration and authentication phases.

Registration Phase. The registration phase is the same as that of Das' protocol except that N_i is computed as $N_i = h(ID_i\|PW_i\|x_a) \bigoplus h(K)$.

Authentication Phase. It starts with the submission of ID_i and PW_i by U_i. Once U_i inputs those values, then U_i's smart card authenticates ID_i and PW_i by comparing those values with the values stored in it. The smart card then computes $DID_i = h(ID_i\|PW_i\|x_a) \bigoplus h(x_a\|T)$ and $C_i = h(N_i\|x_a\|T)$ to send $\{DID_i, C_i, T\}$ to GW. Upon receiving the request from U_i, GW validates T and authenticates C_i by comparing it with $C_i^* = h((DID_i \bigoplus h(x_a\|T) \bigoplus h(K))\|x_a\|T)$. After validation, GW computes an encryption key $EK_{i,n} = h_1(DID_i\|S_n\|(DID_i \bigoplus h(x_a\|T) \bigoplus h(K))\|x_a\|T)$ and a MAC key $MK_{i,n} = h_2(DID_i\|S_n\|\ (DIDi \bigoplus h(x_a\|T) \bigoplus h(K))\|x_a\|T)$ between U_i and the sensor node S_n. To provide a secure channel for $EK_{i,n}$ and $MK_{i,n}$ between S_n and itself, GW computes the encryption key $EK_{GW,n} = h_1(GW\|S_n\|x_n\|T\prime)$ and the MAC key $MK_{GW,n} = h_2(GW\|S_n\|x_n\|T')$, respectively, where x_n is a predistributed symmetric key between GW and S_n, and T' is the current time. GW then encrypts $EK_{i,n}$ and $MK_{i,n}$ using the key $EK_{GW,n}$ computed in the previous step and produces $D_i = E_{EK_{GW},n}(EK_{i,n}, MK_{i,n})$. It also computes a MAC $A_i = h_0(DID_i\|S_n\|D_i\|T', MK_{GW,n})$ using the key $MK_{GW,n}$, and transmits $\{DID_i, D_i, T', A_i\}$ to S_n. When S_n receives those values, it first verifies T' and computes $A_i^* = h_0(DID_i\|S_n\|D_i\|T', MK_{GW,n})$ using $MK_{GW,n}$ and then checks if A_i is equal to A_i^*. After verification, S_n decrypts D_i with $EK_{GW,n}$ and recovers $EK_{i,n}$ and $MK_{i,n}$. Data sensed by nodes is encrypted with $EK_{i,n}$ as $R = E_{EK_{i,n}}(\text{Data})$ and the MAC is computed with $MK_{i,n}$ as $B_i = h_0(DID_i\|S_n\|R\|T'', MK_{i,n})$, where T'' is the current time. Once R and B_i are calculated, the $\{Sn, R, T'', B_i\}$ message is sent to U_i. U_i then verifies T'' and checks B_i by comparing it with $B_i^* = h_0(DID_i\|S_n\|R\|T'', MK_{i,n})$, where $MK_{i,n} = h_2(DID_i\|S_n\|N_i\|x_a\|T)$. If this verification is successful, the sensed data is recovered by decrypting R using $EK_{i,h} = h_1(DID_i\|S_n\|N_i\|x_a\|T)$.

2.4.2. Cryptoanalysis of Nyang-Lee's Scheme. The Nyang-Lee's proposal still has some critical security pitfalls and limitations as shown below.

Parallel Session Attack. The Nang-Lee scheme is vulnerable to a parallel session attack in the same way that happens in [9, 10]. A legal user Tom can obtain the message $\{DID_i, C_i, T_1\}$ between GW and U_i at timestamp T_1 to obtain the DID_i value at $T_1 DID_{i(T1)} = h(ID_i \| PW_i \| x_a) \bigoplus h(x_a \| T_1)$. Tom then can forge the DID_i value at $T_2 DID_{i(T2)}$ by generating $DID_{\text{Tom}(T1)} = h(ID_{\text{Tom}} \| PW_{\text{Tom}} \| x_a) \bigoplus h(x_a \| T_1)$ and $DID_{\text{Tom}(T2)} = h(ID_{\text{Tom}} \| PW_{\text{Tom}} \| x_a) \bigoplus h(x_a \| T_2)$, and computing $DID_{i(T2)} = DID_{i(T1)} \bigoplus DID_{\text{Tom}(T1)} \bigoplus DID_{\text{Tom}(T2)} = h(ID_i \| PW_i \| x_a) \bigoplus h(x_a \| T_1) \bigoplus h(ID_{\text{Tom}} \| PW_{\text{Tom}} \| x_a) \bigoplus h(x_a \| T_1) \bigoplus h(ID_{\text{Tom}} \| PW_{\text{Tom}} \| x_a) \bigoplus h(x_a \| T_2)$. Once U_i's $DID_{i(T2)}$ is obtained, Tom can send a new session message $\{DID_{i(T2)}, C_i, T_2\}$ at T_2 for a new login request.

Privileged-Insider Attack. The system administrator or privileged-insider of GW may try to impersonate U_i by authenticating himself/herself to other servers where U_i could be a registered user. This is possible because GW receives the password of U_i in plaintext, that is, PW_i, in the registration phase and because many users use same password to access different applications of servers.

Lack of Password Change Phase. This scheme does not provide a password change phase for U_i, which is a requirement for a secure system.

2.5. Huang et al.'s Scheme. In [8], the authors point out that the security features of Das' scheme is based on the x_a value and its leakage can compromise the entire network. After explaining the limitations of Das' scheme, the authors propose an improved scheme which still has vulnerabilities and limitations.

2.5.1. Review of Huang et al.'s Scheme. In this scheme, GW computes and stores $h(x_a \| S_n)$ in the designated sensor node S_n before deployment. Note that each S_n is responsible for exchanging data with users. The improved scheme consists of four phases: the registration phase, login phase, verification phase, and password change phase.

Registration Phase. The user U_i submits his/her identity ID_i and password PW_i to GW using a secure channel. GW then computes $N_i = h(ID_i \| PW_i) \bigoplus h(K \| x_a)$ and issues a smart card containing N_i, $h(\cdot)$, ID_i, $h(PW_i)$, and $h(x_a)$ to U_i through a secure channel.

Login Phase. U_i inserts his/her smart card into the terminal and inputs ID_i and PW_i. The smart card then verifies ID_i and PW_i with the data stored in it. Once those values are verified, the smart card computes $DID_i = h(ID_i \| PW_i) \bigoplus h(h(x_a) \| T)$ and $C_i = h(N_i \| h(x_a) \| T)$, where T is the current timestamp, and sends $\{DID_i, C_i, T\}$ to GW.

Verification Phase. Once $\{DID_i, C_i, T\}$ has been received at time T^*, GW verifies T and authenticates C_i by comparing it with $C_i^* = h((h(ID_i \| PW_i)^* \bigoplus h(K \| x_a)) \| h(x_a) \| T)$, where $h(ID_i \| PW_i)^* = DID_i \bigoplus h(h(x_a) \| T)$. If C_i^* is different to C_i, GW rejects the login request. Otherwise, GW accepts the request and sends a message $\{DID_i, A_i, T'\}$ to some nearest sensor node S_n, where $A_i = h(DID_i \| h(x_a \| S_n) \| T')$ and T' is the current timestamp of GW's system. Finally, S_n, after validating T', computes $A_i^* = h(DID_i \| h(x_a \| S_n) \| T')$, and checks whether it is equal to A_i. If those values match, then S_n responds to the query from U_i; otherwise, the query is rejected.

Password Change Phase. When U_i wants to update his/her password, he/she inserts the smart card into a card reader and enters the original password PW_i and the new password PW_i'. The smart card computes $N_i' = N_i \bigoplus h(ID_i \| PW_i) \bigoplus h(ID_i \| PW_i') = h(ID_i \| PW_i') \bigoplus h(k \| x_a)$ and replaces the stored N_i and $h(PW_i)$ with N_i' and $h(PW_i)$, respectively.

2.5.2. Cryptoanalysis of Huang et al.'s Scheme

Parallel Session Attack. Huang et al.'s scheme is vulnerable to parallel session attacks in the same way that can happen in [7, 9, 10]. A legal user Tom can obtain the message $\{DID_i, C_i, T_1\}$ between GW and U_i at timestamp T_1 to obtain the DID_i value at T_1 $DID_{i(T1)} = h(ID_i \| PW_i) \bigoplus h(h(x_a) \| T_1)$. Tom then can forge the DID_i value at T_2 $DID_{i(T2)}$ by generating $DID_{\text{Tom}(T1)} = h(ID_{\text{Tom}} \| PW_{\text{Tom}}) \bigoplus h(h(x_a) \| T_1)$ and $DID_{\text{Tom}(T2)} = h(ID_{\text{Tom}} \| PW_{\text{Tom}}) \bigoplus h(h(x_a) \| T_2)$, and computing $DID_{i(T2)} = DID_{i(T1)} \bigoplus DID_{\text{Tom}(T1)} \bigoplus DID_{\text{Tom}(T2)} = h(ID_i \| PW_i) \bigoplus h(h(x_a) \| T_1) \bigoplus h(ID_{\text{Tom}} \| PW_{\text{Tom}}) \bigoplus h(h(x_a) \| T_1) \bigoplus h(ID_{\text{Tom}} \| PW_{\text{Tom}}) \bigoplus h(h(x_a) \| T_2)$. Once U_i's $DID_{i(T2)}$ is obtained, Tom can send a new session message $\{DID_{i(T2)}, C_i, T_2\}$ at T_2 for a new login request.

Privileged-Insider Attack. The system administrator or privileged-insider of GW may try to impersonate U_i by authenticating himself/herself to other servers where U_i could be a registered user. This is possible because GW receives the password of U_i in plaintext, that is, PW_i, in the registration phase, and because many users use same password to access different applications of servers.

Lack of Mutual Authentication. Huang et al.'s scheme does not provide a mutual authentication mechanism. Therefore, it is vulnerable to GW and Sensor node spoofing attacks. U_i does not have any mechanism to verify the validity of messages sent by GW. Therefore, the adversary can respond to the $\{DID_i, C_i, T\}$ message with false data. In the same way, the adversary can place a false sensor node S_f to respond to the $\{DID_i, A_i, T'\}$ message with false data.

3. Proposed Protocol

This section describes a proposed enhanced protocol which fixes the weaknesses of previous works. The proposed protocol is composed of three phases: Registration, authen-

tication and password change phases executed among three independent entities: users, gateway node, and sensor nodes.

3.1. Registration Phase. A user U_i chooses his/her identity ID_i and password PW_i and inputs them to the terminal. The terminal then generates a random number r_i and computes $PPW_i = h(PW_i) \bigoplus r_i$, where $h(\cdot)$ is a hash function and $\bigoplus$ is an XOR operator. Once PPW_i has been calculated, ID_i and PPW_i are sent to the Gateway node GW using a secure channel. GW then computes $M_i = h(ID_i \| PPW_i)$, $N_i = h(ID_i \| PPW_i) \bigoplus h(K \| x_a)$, and $K_i = h(x_a \| ID_i)$, where K is a symmetric key only known by GW, and where x_a is a secret parameter generated securely by GW, and "$\|$" is a concatenation operator. Once M_i, N_i, and K_i have been calculated, GW personalizes a smart card with the parameters $h(\cdot)$, M_i, N_i, and K_i. Finally, GW delivers the smart card to U_i in a secure manner and U_i stores r_i into the smart card.

Meanwhile, a unique secret key $K_n = h(x_a \| S_n)$ is stored in each sensor node responsible for exchanging data with U_i, where S_n is the unique identification of the sensor node.

3.2. Authentication Phase. This phase is performed when U_i requests access to the data of a sensor node, and it is composed of login and verification phases.

Login Phase. U_i inserts the smart card and inputs his/her ID_i and PW_i. The smart card then computes $PPW_i = h(PW_i) \bigoplus r_i$ and $M_i^* = h(ID_i \| PPW_i)$ and compares M_i^* with M_i to authenticate U_i. If those values do not match, the authentication request is rejected. Otherwise, the smart card computes $DID_i = h(ID_i \| PPW_i) \bigoplus h(K_i \| T)$ and transmits $\{DID_i, T, ID_i, RN_i\}$ to GW, where T is the current timestamp of U_i's system and RN_i is a random nonce generated by U_i.

Verification Phase. Upon receiving the login request at time T^*, GW validates T. If $(T^* - T) > \Delta T$, GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW computes $K_i^* = h(x_a \| ID_i)$, $h(ID_i \| PPW_i)^* = DID_i \bigoplus h(K_i^* \| T)$, and $A_i = h(h(ID_i \| PPW_i)^* \| K_i^* \| RN_i)$. Once computed A_i, GW generates a random nonce $RN1_{GW}$ and transmits the message $\{A_i, RN1_{GW}\}$ to U_i. U_i then computes $A_i^* = h(M_i \| K_i \| RN_i)$ and checks whether it is equal to A_i. If A_i^* is different to A_i, U_i finishes the authentication process; otherwise, U_i computes $B_i = h(N_i \| K_i \| RN1_{GW})$, and sends the message $\{B_i\}$ to GW. GW then computes $B_i^* = h(h(ID_i \| PPW_i)^* \bigoplus h(K \| x_a) \| K_i^* \| RN1_{GW})$ and checks whether it is equal to B_i. GW authenticates U_i only if those values match. After a valid U_i authentication, GW generates a random nonce $RN2_{GW}$ and transmits the message $\{DID_i, T'', RN2_{GW}\}$ to some nearest sensor node S_n to respond to the query with the data that U_i is looking for, where T'' is the timestamp of GW's system when sending the message. S_n first validates T'' using similar method of T verification, then computes $C_i = h(K_n \| T'' \| RN2_{GW})$ and sends the message $\{C_i, RN_n\}$, where RN_n is a random nonce generated by S_n. GW then computes $K_n^* = h(x_a \| S_n)$ and $C_i^* = h(K_n^* \| T'' \| RN2_{GW})$ and checks whether C_i^* is equal to C_i. Only if those values match, GW responds to S_n's message by sending the message $\{D_i\}$, where $D_i = h(DID_i \| K_n^* \| RN_n)$. Finally, S_n checks the validity of D_i by comparing $D_i^* = h(DID_i \| K_n \| RN_n)$ with the received D_i. If those values match, then U_i is allowed to access S_n's data.

Session Key Establishment. A session key between U_i and GW $K_{U_i\text{-}GW} = h(RN_i \| RN1_{GW} \| K_i)$ and a session key between GW and S_n $K_{S_n\text{-}GW} = h(RN_n \| RN2_{GW} \| K_n)$ could be used if an encryption channel were required after authentication. Additionally, if a direct communication channel between U_i and S_n were required, a bilateral session key $K_{U_i\text{-}S_n}$ could be established through GW. In this case, GW would generate a random $K_{U_i\text{-}S_n}$ and send $RN_i \| K_{U_i\text{-}S_n}$ encrypted with $K_{U_i\text{-}GW}$ to U_i and $RN_n \| K_{U_i\text{-}S_n}$ encrypted with $K_{S_n\text{-}GW}$ to S_n.

3.3. Password Change Phase. U_i inputs his/her ID_i, PW_i, and new password NewPW_i to the terminal. The smart card then calculates $PPW_i = h(PW_i) \bigoplus r_i$ and $M_i^* = h(ID_i \| PPW_i)$, and verifies the validity of ID_i and PW_i by comparing M_i^* with M_i. If those values do not match, the password change request is rejected. Otherwise, the smart card computes New$M_i = h(ID_i \| \text{New}PPW_i)$, $h(K \| x_a) = N_i \bigoplus h(ID_i \| PPW_i)$, and New$N_i = h(ID_i \| \text{New}PPW_i) \bigoplus h(K \| x_a)$, where New$PPW_i = h(\text{New}PW_i) \bigoplus r_i$. Finally, the smart card replaces M_i and N_i with NewM_i and NewN_i, respectively.

4. Analysis of the Protocol

In this section, we analyze the proposed protocol in terms of security and performance.

4.1. Security Analysis. In this part, we analyze the security of the proposed protocol in terms of formal verification and analysis of aforementioned attacks. The registration and password change phases of the proposed mechanism were excluded from this analysis because they are executed in a secure environment. In the analysis of the authentication phase, the widely used Dolev-Yao [16] threat model was applied, which assumes that two communicating parties communicate over an insecure channel.

4.1.1. Formal Proof Based on BAN Logic. In this subsection, we demonstrate the security of the proposed mechanism by a well-known formal model called BAN logic [17]. BAN logic has been widely used in different works such as [18–20] to reason about their security validation.

The logical notations of BAN logic used in this paper are as described as follows.

$P \mid\equiv X$: The principal P believes that X holds. In other words, it means that P is entitled to act as though X is true.

$\#(X)$: The formula X is fresh. That is, X has not been sent before in any run of the protocol.

$P \Rightarrow X$: The principal P has jurisdiction over the statement X. That is, P is an authority on X and can be trusted on X.

$P \lhd X$: The principal P sees the statement X. That is, someone has sent a message to P containing X, and P can read and repeat X.

$P \mid\sim X$: The principal P once said the statement X. That is, P sent a message containing X sometime.

(X, Y) : The formula X or Y is one part of the formula (X, Y).

$\{X\}_K$: The formula X is encrypted under the key K

$(X)_K$: The formula X is hashed with the key K, and K may be used to prove the origin of X.

$P \overset{K}{\leftrightarrow} Q$: Principals P and Q may use the shared key K to communicate. The key K will never be discovered by any principal except P and Q.

Moreover, we describe some main logical postulates to be used in proofs.

Message-Meaning Rule. If the principal P believes that the secret key is shared with the principal Q and P sees that the statement X is encrypted under K, then the principal P believes that the principal Q once said the statement X

$$\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}. \tag{1}$$

Freshness-Conjuncatenation Rule. Provided that the principal P believes freshness of the statement X, the principal P believes freshness of the (X, Y)

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}. \tag{2}$$

Nonce-Verification Rule. Provided that the principal P believes that the statement X has never been utter before and the principal Q once said X, the principal P believes that Q believes X

$$\frac{P \mid\equiv \#(X),\ P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}. \tag{3}$$

Jurisdiction Rule. Provided that the principal P believes that the principal Q jurisdiction over the statement X, the principal P believes Q on the validity of X

$$\frac{P \mid\equiv Q \Longrightarrow X,\ P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}. \tag{4}$$

Belief Rules. A necessary property of the belief operator is that P believes a set of statements if and only if P believes each statement separately. This justifies the following rules:

$$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X, Y)},\quad \frac{P \mid\equiv (X, Y)}{P \mid\equiv X},\quad \frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X},\quad \frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X}. \tag{5}$$

In the following, we will demonstrate the security of the proposed scheme using the BAN logic. The proposed scheme will satisfy the following goals:

$$U_i \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW, \tag{G.1}$$
$$U_i \mid\equiv GW \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW, \tag{G.2}$$
$$GW \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW, \tag{G.3}$$
$$GW \mid\equiv U_i \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW, \tag{G.4}$$
$$S_n \mid\equiv S_n \overset{K_{S_n\text{-}GW}}{\longleftrightarrow} GW, \tag{G.5}$$
$$S_n \mid\equiv GW \mid\equiv S_n \overset{K_{S_n\text{-}GW}}{\longleftrightarrow} GW, \tag{G.6}$$
$$GW \mid\equiv S_n \overset{K_{S_n\text{-}GW}}{\longleftrightarrow} GW, \tag{G.7}$$
$$GW \mid\equiv S_n \mid\equiv S_n \overset{K_{S_n\text{-}GW}}{\longleftrightarrow} GW, \tag{G.8}$$
$$U_i \mid\equiv U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n, \tag{G.9}$$
$$U_i \mid\equiv S_n \mid\equiv U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n, \tag{G.10}$$
$$S_n \mid\equiv U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n, \tag{G.11}$$
$$S_n \mid\equiv U_i \mid\equiv U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n. \tag{G.12}$$

First, we transform the messages of the proposed protocol to the idealized form as follows:

$$U_i \rightarrow GW: \quad DID_i : (\{h(ID_i \| PPW_i)\}_{Ki}, (U_i \overset{K_i}{\longleftrightarrow} GW,\ T)_{Ki}) \tag{M.1}$$
$$U_i \leftarrow GW: \quad A_i : (h(ID_i \| PPW_i), U_i \overset{K_i}{\longleftrightarrow} GW, RN_i)_{K_i} \tag{M.2}$$
$$U_i \rightarrow GW: \quad B_i : (N_i, U_i \overset{K_i}{\longleftrightarrow} GW, RN1_{GW})_{K_i} \tag{M.3}$$
$$S_n \leftarrow GW: \quad DID_i : (\{h(ID_i \| PPW_i)\}_{Ki}, (U_i \overset{K_i}{\longleftrightarrow} GW, T)_{K_i}) \tag{M.4}$$
$$S_n \rightarrow GW: \quad C_i : (S_n \overset{K_n}{\longleftrightarrow} GW,\ T'', RN2_{GW})_{K_n} \tag{M.5}$$
$$S_n \leftarrow GW: \quad D_i : (DID_i, S_n \overset{K_n}{\longleftrightarrow} GW, RN_n)_{K_n} \tag{M.6}$$
$$U_i \leftarrow GW: \quad \{RN_i, U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW} \tag{M.7}$$
$$S_n \leftarrow GW: \quad \{RN_n, U_i \overset{K_{U_i\text{-}S_n}}{\longleftrightarrow} S_n\}_{KS_n\text{-}GW}. \tag{M.8}$$

Second, we make the following assumptions about the initial state of the scheme to analyze the proposed scheme:

$$GW \mid\equiv \#(T), \tag{A.1}$$
$$U_i \mid\equiv \#(RN_i), \tag{A.2}$$
$$GW \mid\equiv \#(RN1_{GW}), \tag{A.3}$$
$$S_n \mid\equiv \#(T''), \tag{A.4}$$
$$S_n \mid\equiv \#(RN_n), \tag{A.5}$$
$$GW \mid\equiv \#(RN2_{GW}), \tag{A.6}$$
$$U_i \mid\equiv U_i \overset{K_i}{\longleftrightarrow} GW, \tag{A.7}$$

$$GW \mid\equiv U_i \overset{K_i}{\longleftrightarrow} GW, \quad \text{(A.8)}$$

$$S_n \mid\equiv S_n \overset{K_n}{\longleftrightarrow} GW, \quad \text{(A.9)}$$

$$GW \mid\equiv S_n \overset{K_n}{\longleftrightarrow} GW, \quad \text{(A.10)}$$

$$U_i \mid\equiv GW \Rightarrow U_i \overset{K_{U_i \cdot S_n}}{\longleftrightarrow} S_n, \quad \text{(A.11)}$$

$$S_n \mid\equiv GW \Rightarrow U_i \overset{K_{U_i \cdot S_n}}{\longleftrightarrow} S_n. \quad \text{(A.12)}$$

Finally, we perform the proof steps to the idealized form of the proposed scheme based on the BAN logic rules and the assumptions (see Table 1).

The proposed goals were reached by (S.17)–(S.24), (S.29), (S.34), and (S.35). In summary, we have demonstrated how the proposed scheme provides mutual authentication as well as establishes a fresh session keys among U_i, GW, and S_n.

4.1.2. Security Verification from Possible Attacks. This subsection analyzes the security of the proposed solution against possible attacks. We assume that common communication channels are insecure and that there exists an attacker who can intercept all messages communicated among U_i, GW, and S_n. In addition, we assume that the attacker can obtain or steal legal user U_i's smart card. Based on these assumptions, the attacker might execute certain attacks to interfere with the proposed scheme.

Parallel Session Attack. Even though another legal user of the system, say Tom, eavesdrops on U_i's message $\{DID_i, T_1, ID_i, RN_i\}$, he cannot obtain the $DID_{i(T2)}$ as happens in previous protocols because the DID_i in our protocol is calculated as $h(ID_i \| PPW_i) \bigoplus h(K_i \| T)$which is based on U_i's unique values. The equation $PPW_i = h(PW_i) \bigoplus r_i$ contains r_i which is random and individual for each U_i, and K_i is unique for each U_i. Therefore, the resultant value of $DID_{i(T1)} \bigoplus DID_{\text{Tom}(T1)} \bigoplus hDID_{\text{Tom}(T2)}$ will be $h(ID_i \| h(PW_i) \bigoplus r_i) \bigoplus h(K_i \| T_1) \bigoplus h(ID_{\text{Tom}} \| h(PW_{\text{Tom}}) \bigoplus r_{\text{Tom}}) \bigoplus h(K_{\text{Tom}} \| T_1) \bigoplus h(ID_{\text{Tom}} \| h(PW_{\text{Tom}}) \bigoplus r_{\text{Tom}}) \bigoplus h(K_{\text{Tom}} \| T_2) = h(ID_i \| h(PW_i) \bigoplus r_i) \bigoplus h(K_i \| T_1) \bigoplus h(K_{\text{Tom}} \| T_1) \bigoplus h(K_{\text{Tom}} \| T_2)$, a totally different value from $DID_{i(T2)}$.

Privileged-Insider Attack. In the proposed solution, U_i transmits his/her pseudopassword $PPW_i = h(PW_i) \bigoplus r_i$ instead of PW_i. Therefore, GW will never know the PW_i value. This means that only U_i will know his/her secret password, thus protecting U_i in this way from a privileged-insider attack. Additionally, a random value r_i is incorporated inside PPW_i to make the discovery of PW_i harder.

Gateway Node Bypassing Attack. The reason for the possibility of a GW bypassing attack in [6, 9] is due to the sharing of secret parameter x_a with the sensor node S_n and user U_i. If the value of x_a is compromised, then the whole sensor network will become vulnerable to the gateway node bypassing attack. On the other hand, the reason for the possibility of the gateway node bypassing attack in [10] is due to the secret value x_s which is stored in the sensor nodes and can be extracted using similar method of extracting x_a from a smart card [11–13]; if x_s is extracted, the adversary can execute the GW bypassing attack using $DID_f = h(ID_f \| PW_f) \bigoplus h(x_a \| T_f)$ and $A_f = h(DID_f \| S_n \| x_s \| T_f)$.

In the proposed protocol, U_i's smart card and the sensor node S_n do not store either x_a or x_s, but instead store other individual secret values $K_i = h(x_a \| ID_i)$ and $K_n = h(x_a \| S_n)$ which are unique per smart card and sensor node. Therefore, even if the K_i or K_n value were extracted from a smart card or node, the rest of the users of the nodes will still maintain their security.

Mutual Authentication. The proposed protocol provides both mutual authentication between U_i and GW, and between GW and S_n.

(1) Mutual authentication between U_i and GW : GW verifies the authenticity of U_i by comparing B_i sent by U_i with the B_i^* value calculated by itself. B_i can only be computed by the authentic U_i because it is based on secret values such as N_i and K_i which are personal to each U_i. On the other hand, U_i verifies the authenticity of GW by comparing A_i sent by GW with the A_i^* value computed by U_i. A_i can only be computed by the authentic GW because it is based on the secret values K and x_a only known by GW.

(2) Mutual authentication between GW and S_n : S_n verifies the authenticity of GW by comparing D_i sent by GW with the D_i^* value calculated by itself. D_i can only be computed by the authentic GW because it is based on the secret value x_a. On the other hand, GW verifies the authenticity of S_n by comparing C_i sent by GW with the C_i^* value computed by GW. C_i can only be computed by the authentic S_n because it is based on the secret K_n value only known by the specific S_n.

Masquerade Attack. An adversary who wants to impersonate a valid user U_i to log into the network must calculate a valid DID_i and B_i. Since $DID_i = h(ID_i \| PPW_i) \bigoplus h(K_i \| T)$ and $B_i = h(N_i \| K_i \| RN1_{GW})$ are calculated by a one-way hash function, the adversary cannot decipher such values. Additionally, DID_i and B_i cannot be created arbitrarily because they are based on secret values such as K_i and PPW_i, Furthermore, the adversary cannot forge the GW because he/she does not know the K and x_a values.

Replay Attack. Timestamps and random nonces are used to avoid replay attacks. At the beginning of each authentication request, a timestamp mechanism is used to guarantee the freshness of the authentication request. Later, a stronger mechanism: challenge-response of codified nonces is used to respond to the authentication requests. An adversary cannot replay a valid GW's verification message $\{A_i, RN_{GW}\}$ to U_i to succeed in verification because the RN_i value required for A_i computation is regenerated in each request. In the same way, the adversary cannot replay a valid U_i's verification message $\{B_i\}$ to succeed in verification because the $RN1_{GW}$ value used in B_i is regenerated in each request. In addition, the authentication messages between GW and S_n are protected using the same method of messages between U_i and GW.

Table 1

$GW \lhd (\{h(ID_i \| PPW_i)\}_{K_i}, (U_i \overset{K_i}{\longleftrightarrow} GW, T)_{Ki})$	(S.1)	//by (M.1)
$GW \mid\equiv U_i \mid\sim h(ID_i \| PPW_i)$	(S.2)	//by (S.1), (A.8), and message-meaning rule
$GW \mid\equiv U_i \mid\sim (U_i \overset{K_i}{\longleftrightarrow} GW, T)_{Ki}$	(S.3)	//by (S.1), (A.8), and message-meaning rule
$GW \mid\equiv U_i \mid\equiv h(ID_i \| PPW_i)$	(S.4)	//by (S.2), (S.3), (A.1), freshness-conjuncatenation rule, nonce-verification rule
$U_i \lhd (h(ID_i \| PPW_i),\ U_i \overset{K_i}{\longleftrightarrow} GW, RN_i)_{Ki}$	(S.5)	//by (M.2)
$U_i \mid\equiv GW \mid\sim RN_i$	(S.6)	//by (S.5), (A.7), and message-meaning rule
$U_i \mid\equiv GW \mid\equiv RN_i$	(S.7)	//by (S.6), (A.2), and nonce-verification rule
$GW \lhd (N_i, U_i \overset{K_i}{\longleftrightarrow} GW, RN1_{GW})_{Ki}$	(S.8)	//by (M.3)
$GW \mid\equiv U_i \mid\sim\ RN1_{GW}$	(S.9)	//by (S.8), (A.8), and message-meaning rule
$GW \mid\equiv U_i \mid\equiv RN1_{GW}$	(S.10)	//by (S.9), (A.3), and nonce-verification rule
$GW \lhd (S_n \overset{K_n}{\longleftrightarrow} GW, T'', RN2_{GW})_{Kn}$	(S.11)	//by (M.5)
$GW \mid\equiv S_n \mid\sim RN2_{GW}$	(S.12)	//by (S.11), (A.10), and message-meaning rule
$GW \mid\equiv S_n \mid\equiv RN2_{GW}$	(S.13)	//by (S.12), (A.6), and nonce-verification rule
$S_n \lhd (DID_i, S_n \overset{K_n}{\longleftrightarrow} GW, RN_n)_{Kn}$	(S.14)	//by (M.6)
$S_n \mid\equiv GW \mid\sim RN_n$	(S.15)	//by (S.14), (A.9), and message-meaning rule
$S_n \mid\equiv GW \mid\equiv RN_n$	(S.16)	//by (S.15), (A.5), and nonce-verification rule
$U_i \mid\equiv \#(U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW), U_i \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW$	(S.17)	//once verified (S.7), U_i computes $K_{U_i\text{-}GW}$
$GW \mid\equiv \#(U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW), GW \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW$	(S.18)	//once verified (S.4) and (S.10), GW computes $K_{U_i\text{-}GW}$
$S_n \mid\equiv \#(S_n \overset{K_{Sn\text{-}GW}}{\longleftrightarrow} GW), S_n \mid\equiv S_n \overset{K_{Sn\text{-}GW}}{\longleftrightarrow} GW$	(S.19)	//once verified (S.16), S_n computes $K_{S_n\text{-}GW}$
$GW \mid\equiv \#(Sn \overset{K_{Sn\text{-}GW}}{\longleftrightarrow} GW), GW \mid\equiv S_n \overset{K_{Sn\text{-}GW}}{\longleftrightarrow} GW$	(S.20)	//once verified (S.13), GW computes $K_{S_n\text{-}GW}$
$U_i \mid\equiv GW \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW$	(S.21)	//by (S.17), (S.18), and $K_{U_i\text{-}GW}$ generation algorithm shared between U_i and GW
$GW \mid\equiv U_i \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW$	(S.22)	//by (S.17), (S.18), and $K_{U_i\text{-}GW}$ generation algorithm shared between U_i and GW
$S_n \mid\equiv GW \mid\equiv S_n \overset{K_{Sn\text{-}GW}}{\longleftrightarrow} GW$	(S.23)	//by (S.19), (S.20), and $K_{S_n\text{-}GW}$ generation algorithm shared between S_n and GW
$GW \mid\equiv S_n \mid\equiv S_n \overset{K_{Sn-GW}}{\longleftrightarrow} GW$	(S.24)	//by (S.19), (S.20), and $K_{S_n\text{-}GW}$ generation algorithm shared between S_n and GW
$U_i \lhd \{RN_i, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.25)	//by (M.7) and seeing rule
$U_i \mid\equiv GW \mid\sim \{RN_i, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.26)	//by (S.25), (S.17), and message-meaning rule
$U_i \mid\equiv GW \mid\equiv \{RN_i, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.27)	//by (S.26), (A.2), and nonce-verification rule
$U_i \mid\equiv GW \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n$	(S.28)	//by (S.27) and breaking conjunction rule
$U_i \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n$	(S.29)	//by (S.28), (A.11), and jurisdiction rule
$S_n \lhd \{RN_n, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.30)	//by (M.8) and seeing rule
$S_n \mid\equiv GW \mid\sim \{RN_n, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.31)	//by (S.30), (S.19), and message-meaning rule
$S_n \mid\equiv GW \mid\equiv \{RN_n, U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n\}_{KU_i\text{-}GW}$	(S.32)	//by (S.31), (A.5), and nonce-verification rule
$S_n \mid\equiv GW \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n$	(S.33)	//by (S.32) and breaking conjunction rule
$S_n \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n$	(S.34)	//by (S.33), (A.12), and jurisdiction rule
$U_i \mid\equiv S_n \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n, S_n \mid\equiv U_i \mid\equiv U_i \overset{K_{U_i-Sn}}{\longleftrightarrow} S_n$	(S.35)	//by (S.28), (S.29), (S.33), and (S.34)

Stolen-Verifier Attack. One of the features of the proposed protocol is the absence of a password/verifier table which prevents our solution from stolen-verifier attacks.

Guessing Attack. In the proposed scheme, secret values are never sent in plaintext, but encrypted inside a one-way hash function. Therefore, even if the adversary got DID_i, A_i, B_i, C_i, or D_i, he or she could not guess any secret values (PW_i, K_i, K_n, or K) because of the one-way property of the hash function.

Many Logged-IN Users with the Same Login ID. By using two-factor based authentication, the proposed scheme offers higher protection than only-password-based schemes against this attack. Assuming that the U_i's smart card is not cloned, the proposed protocol successfully prevents this threat

Table 2: List of enhanced security features of the proposed protocol.

Security Feature	Proposed	M. Das'	Nyang-Lee's	Huang et al.'s	Chen-Shih's	Khan-Alghathbar
Secure password change/update	Yes	No	No	Yes	No	Yes
Protection against insider's attack	Yes	No	No	No	No	Yes
Protection against *GW* bypassing attack	Yes	No	No	No	No	Partial
Protection against Parallel session attack	Yes	No	No	No	No	No
Mutual authentication (GW/S_n)	Yes	No	Yes	No	No	Yes
Mutual authentication (U_i/GW)	Yes	No	Yes	No	No	No
Session Key Establishment	Yes	No	Yes	No	No	No

Table 3: Performance analysis/number of operations (h: hash, se: symmetric encryption, sd: symmetric decryption).

Phase	Entity	Proposed	M. Das'	Nyang-Lee's	Huang et al.'s	Chen-Shih's	Khan-Alghathbar's
Registration phase	User	1 h	0 h	0 h	0 h	0 h	0 h
	GW	3 h	3 h	3 h	4 h	3 h	3 h
	Sensor	0 h	0 h	0 h	0 h	0 h	0 h
Login phase	User	3 h	4 h	4 h	4 h	4 h	4 h
	GW	0 h	0 h	0 h	0 h	0 h	0 h
	Sensor	0 h	0 h	0 h	0 h	0 h	0 h
Verification phase	User	2 h	0 h	1 h + 1 sd	0 h	1 h	0 h
	GW	8 h	4 h	9 h + 1 se	6 h	5 h	5 h
	Sensor	2 h	1 h	2 h + 1 se + 1 sd	1 h	1 h	2 h
Password change phase	User	5 h	—	—	4 h	—	4 h
	GW	0 h	—	—	0 h	—	0 h
	Sensor	0 h	—	—	0 h	—	0 h

because the authentication process requires computation executed inside the valid smart card.

Brute-Force Attack. An attacker can try two kinds of brute-force attacks. (1) First, the attacker can attempt to authenticate by sending random or sequential messages (DID_i/B_i or DID_i/D_i combinations) to *GW* or S_n. However, as well as explained in the replay attack, this attack becomes infeasible because each authentication process uses a different nonce. (2) On the other hand, an insider with a valid smart card can try to discover the secret values K or x_a by performing brute-force attacks. However, the determination of those values is infeasible because they are stored using a secure one-way hash functions. If higher level of protection for x_a was required, additional random numbers R_i and R_n could be added for the generation of $K_i = h(x_a \| ID_i \| R_i)$ and $K_i = h(x_a \| S_n \| R_n)$, respectively, which would be stored in secret in the *GW*. By using this additional random numbers, the number of possible combinations to decipher K_i and K_n is increased by 2^n times, where n is the size it bits of R_i and R_n.

Password Change Phase. Our proposal offers a light-weight password change phase that does not require communication with *GW*, making it secure and efficient.

Session Key Establishment. Our proposal offers a simple and practical method for session key establishment among U_i, *GW*, and S_n.

Table 2 shows the comparison of security features among different works. This demonstrates how our scheme is stronger in terms of security. Our approach provides protection against different kinds of attacks (privileged insider's attack, gateway node bypassing attack), also provides a secure password change phase, session key establishment, and achieves complete mutual authentication (mutual authentication between *GW* and S_n, and between U_i and *GW*), features that previous works do not offer or offer with limitations.

4.2. Performance Analysis. Table 3 indicates the number of hash operations required in each phase for each entity. It shows that our protocol requires a few more operations in the verification phase than some previous works. However, the majority of additional operations are executed by U_i or *GW* infrastructure which has no energy or computation power limitations. Therefore, we believe that the additional operations are not an impediment for real implementation. Additionally, we believe that the additional operations are justifiable considering that our protocol includes security features that previous works do not offer, which is indispensable for implementing a reliable and trustworthy network. It

is important to remember that a failure at the component level will often compromise the security of the entire system [21].

According [22], the energy consumed by the MIPS R4000 and MC68328 "DragonBall" processors for performing the SHA-1 hashing function are 0.0000072 mJ/bit and 0.0000410 mJ/bit, respectively. Based on the previously mentioned data, we can calculate the energy consumed by sensor nodes executing the operations of the proposed scheme. Assuming that the size of DID_i, K_n, random nonces, and timestamps are 160 bits long, the total energy consumed by sensor nodes in each authentication would be 0.008064 mJ and 0.04592 mJ for MIPS R4000 and MC68328 "DragonBall" processors, respectively. We believe that the energy consumption of sensor nodes to perform the security operations is acceptable considering the benefits of the proposed solution.

5. Conclusion

In this paper, we have analyzed previous user authentication mechanisms for wireless sensor networks and identified their vulnerabilities and limitations. We also have proposed a robust user authentication for wireless sensor networks that eliminates the identified security flaws. The proposed solution takes advantage of the two-factor authentication concept to provide a secure authentication system offering balanced features in terms of security and performance.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.

[2] J. Carlson, R. Han, S. Lao, C. Narayan, and S. Ghani, "Rapid prototyping of mobile input devices using wireless sensor nodes," in *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '03)*, pp. 21–29, October 2003.

[3] U.A.F., ARGUS Advanced Remote Ground Unattended Sensor Systems. Department of Defense, 2009, http://www.globalse-curity.org/ intell/systems/arguss.htm.

[4] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *Journal of Mobile Multimedia*, vol. 1, no. 4, pp. 307–326, 2006.

[5] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Shengwei, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 244–251, IEEE Computer Society, June 2006.

[6] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, Article ID 4801450, pp. 1086–1090, 2009.

[7] D. Nyang and M. Lee, "Improvement of das's two-factor authentication protocol in wireless sensor networks," Cryptology ePrint Archive 2009/631, http://eprint.iacr.org/2009/631.pdf.

[8] 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 Information Hiding and Multimedia Signal Processing (IIHMSP '10)*, pp. 27–30, October 2010.

[9] 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.

[10] 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.

[11] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th International Advances in Cryptology Conference (CRYPTO '99)*, pp. 388–397, 1999.

[12] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[13] B. Jack, "Exploiting embedded systems," Black Hat, Las Vegas, Nev, USA, 2006, http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Jack.pdf.

[14] M. Raluca, M. Razvan, and A. Terzis, "Gateway design for data gathering sensor networks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 296–304, June 2008.

[15] National Instruments, WSN Ethernet Gateway, http://sine.ni.com/nips/cds/view/p/lang/en/nid/206919/.

[16] D. Doley and A. C. Yao, "On the security of public-key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[17] M. Burrows, M. Abadi, and R. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[18] W. Tsaur, J. Li, and W. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.

[19] S. Wang, Q. Ma, Y. Zhang, and Y. Li, "An authentication protocol for RFID tag and its simulation," *Journal of Networks*, vol. 6, no. 3, pp. 446–453, 2011.

[20] J. Tsai, T. Wu, and K. Tsai, "New dynamic ID authentication scheme using smart cards," *International Journal of Communication Systems*, vol. 23, pp. 1449–1462, 2010.

[21] R. Ying, "Building systems using software components," *Journal of Software Technology*, vol. 9, no. 1, 2006.

[22] D. Carman, P. Kruus, and B. Matt, "Constraints and approaches for distributed sensor network security (Final)," NAI Labs Technical Report #00-010, 2000.