A Performance and Usability Aware Secure Two-Factor User Authentication Scheme for Wireless Sensor Networks

Sang Guun Yoo, Hyukjun Lee, and Juho Kim

Department of Computer Science and Engineering, Sogang University, Seoul 121-712, Republic of Korea

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

Received 23 January 2013; Accepted 22 March 2013

Academic Editor: Muhammad Khurram Khan

Copyright © 2013 Sang Guun Yoo 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.

Recently, several user authentication schemes for wireless sensor networks based on two-factor concept using the smart card technology were proposed. However, they have serious limitations in terms of security and usability. First, even though they are enhancements of other works, they still have several security flaws, such as vulnerability against parallel session, privileged-insider, and gateway-node bypassing attacks and lack of mutual authentication between user station and gateway node. On the other hand, they also present a usability constraint, in a sense that they do not consider the use case when sensor nodes cannot communicate with gateway node. In this case, data collected by isolated sensor nodes could not be accessed until they recover such communication, which is in many times not recoverable rapidly or forever (e.g., military applications, natural disaster monitoring). Due to all these reasons, this paper proposes a robust user authentication scheme which fixes the security weaknesses of previous solutions and provides wider usability considering the use case when the sensor nodes cannot communicate with the gateway node. Once the solution is described, its security is ensured by formal proof and analysis against attacks. Additionally, performance and cost analysis are executed to determine its level of feasibility for real implementation.

1. Introduction

With the growth of wireless sensor network (WSN) application fields, the frequency of WSNs managing critical tasks and important information has also increased. However, most of real applications do not include any security mechanisms making them vulnerable to serious attacks. This fact makes evident the necessity of security solutions for WSNs such as [1]. Among different security mechanisms, a user authentication mechanism that allows only legitimate users to access the WSN’s data is considered one of the most important security mechanisms because it contributes to keeping the confidentiality and integrity of network’s data and because it is an essential primitive upon which other security mechanisms such as secure channel establishment and over-the-air (OTA) programming [2] are built on.

At this point, several research works [3–7] have been executed for providing user authentication schemes for WSNs. Those proposals consider the resource limitations of WSNs and provide lightweight schemes. However, they have serious limitations in terms of security and usability. First, previous works have several security flaws such as vulnerability against parallel session, privileged-insider, and gateway-node bypassing attacks and lack of mutual authentication between user station and the gateway node. On the other hand, they also present a usability constraint, in a sense that they do not consider the case when sensor nodes cannot communicate with the gateway node; in this case, data collected by sensor nodes could not be accessed until they recover this communication, which is in many times not recoverable rapidly or forever (e.g., military applications in war, natural disaster monitoring). Due to all these reasons, this paper proposes an enhanced user authentication scheme that solves the identified security issues and constraints.

The rest of this paper is organized as follows. Section 2 analyzes the existing works to detail their weaknesses and limitations in terms of security and usability. Section 3 then specifies the design criteria of the proposed user authentication scheme. Later, Section 4 presents the details of the proposed solution which solves the vulnerabilities and limitations mentioned in Section 2. Next, Section 5 analyzes the proposed protocol in terms of achieved security.
level, performance, and cost. Finally, Section 6 details the conclusions of this paper.

2. Analysis of Existing Works

Lately, user authentication for wireless sensor networks based on smart cards has been actively researched. The two-factor authentication approach which requires the verification of ownership of both a password and a smart card achieves effectively the purpose of authentic delivery of sensed data while minimizing the load of storing user data in the gateway node. This is because the tamper-proof smart card of the user delivers the function of secure storage of authentication data instead of the gateway node.

In 2009, Das [3] presented a research work where the author proposed an authentication scheme based on the two-factor user authentication concept using the smart card technology. Das’ proposal was considered as an efficient two-factor user authentication scheme because it only required the usage of a small number of hash function calculations [4]. However, even though Das’ proposal was adopted by different research works, its limitations and security flaws were discovered in subsequent works. Nyang and Lee [4] identified that Das’ protocol was vulnerable to offline password guessing and sensor node compromising attacks. Huang et al. [5] also identified some limitations of Das’ scheme such as vulnerability from impersonate attack. Additionally, the authors of [6] pointed out the absence of mutual authentication feature in Das’ protocol while Khan and Alghathbar [7] pointed out additional security flaws of Das’ proposal, describing that it was vulnerable to privileged-insider and gateway-node bypassing attacks.

Once they discovered different vulnerabilities and security limitations of Das’ proposal, the authors of [3–6] also proposed enhanced versions of Das’ protocol to eliminate detected vulnerabilities. However, as shown in our previous work [8], those protocols still include serious vulnerabilities and security limitations of which an attacker can take advantage, exposing WSNs to serious risks. Table 1 summarizes the security analysis of [8] showing that the protocols in [3–7] are still vulnerable to different attacks and have several security limitations.

In addition, all of the aforementioned approaches only focus on their analysis on the security and performance aspects, and neglect the usability aspect. From this point of view, we can say that the previous user authentication schemes also present a serious usability constraint because they do not consider the case when important sensor nodes are isolated. Isolation of sensor nodes could occur because of network link failures between the gateway node and sensor nodes (see Figure 1) or because of disconnections between critical routing sensor nodes (see Figure 2). Sensor nodes isolation is considered problematic for several reasons. First, the isolated sensor nodes frequently store critical information which requires to be transmitted opportune to the user for decision making. However, using the previous user authentication mechanisms, users could not authenticate to the isolated sensor nodes until sensor nodes recover the communication with the gateway node, which is in many times not recoverable rapidly or forever. Additionally, if the link between the gateway node and isolated sensor nodes is not reestablished rapidly, historical data of those nodes could be eliminated because of their limited storage memory capacity. In this regard, this situation demands for a new user authentication scheme with an offline user authentication mechanism which allows users to authenticate directly to

| Security feature                          | Das’ [3] | Nyang-Lee’s [4] | Huang et al.’s [5] | Chen-Shih’s [6] | Khan-Alghathbar’s [7] |
|------------------------------------------|----------|-----------------|-------------------|----------------|----------------------|
| Secure password change/update            | No       | No              | Yes               | No             | Yes                  |
| Protection against insider’s attack      | No       | No              | No                | No             | Yes                  |
| Protection against gateway bypassing attack | No     | No              | No                | No             | Partial              |
| Protection against parallel session attack | No     | Yes             | No                | No             | No                   |
| Mutual authentication (gateway-sensor node) | No     | Yes             | No                | No             | Yes                  |
| Mutual authentication (user station-gateway) | No     | Yes             | No                | No             | No                   |
| Session key establishment                | No       | Yes             | No                | No             | No                   |

Figure 1: Sensor nodes isolated because of broken link between gateway and sensor nodes.

Figure 2: Sensor nodes isolated because of broken link between sensor nodes.

Table 1: Summary of cryptanalysis of previous works based on smart cards.
isolated sensor nodes for getting the critical information in an opportune way.

Even though many people may think sensor node isolations are not common, they are common in particular applications. Here we describe some applications where sensor nodes could be isolated and why offline user authentication (mobile user station authenticating directly to the sensor node) is important. First, let us consider a volcano monitoring sensor network [9] gathering seismic and infrasonic signals. In such systems, there are several events that could provoke sensor nodes’ isolation. One of the possibilities is when an explosion occurs in a side vent (see Figure 3); in such case, the sensor nodes located in the superior region of the side vent can lose the connection with the inferior region because the sensor nodes nearby the side vent are destroyed or buried. Another case is when critical routing sensor nodes are buried or damaged because of lahars or seismic activities. Data of Tungurahua volcano in Ecuador (see Figure 4) published by Instituto Geofísico de la Escuela Politécnica Nacional (http://www.igepn.edu.ec/) illustrates how frequent this situation could occur in an active volcano. The report indicates that 29 lahars and 8400 long-period seismic events were detected only in 2004. In those cases, it is frequent that the isolated nodes store important data that could help to forecast the future behavior of the volcano. Therefore, it is important to provide a mechanism that allows a rapid and opportune access to such information while maintaining the confidentiality and integrity features.

Another type of applications where critical sensor nodes could be isolated is the military one. Consider that a battlefield application has lost some intermediate nodes because they have been destroyed by the enemy, but the nodes deployed in the enemy’s territory have accumulated important data. In this case, an automaton user station could be sent to the enemy’s territory to gather such information. An easy way to authenticate to the sensor nodes is that the automaton carries the secret keys. However, carrying the secret keys in hostile environment opens wide possibilities of leakage of the secret values which could compromise the security of the whole network. Therefore, the user authentication scheme must provide a way to maintain the security of the sensor network even if the automaton is taken by the adversary.

In conclusion, as described previously, previous works present several limitations in terms of security and usability, and this situation creates the need for designing an enhanced user authentication scheme which overcomes such constraints.

3. User Authentication Scheme Design Criteria

We believe that several limitations of previous works were produced because of the absence of a concrete and clear requirement elicitation process. To avoid falling into the same mistake, this paper has decided to describe the design criteria of the proposed user authentication scheme.
3.1. General Considerations and Assumptions. The proposed user authentication scheme must be able to be implemented in a scenario with the following considerations and assumptions.

(i) The network is composed of traditional elements, that is, sensor nodes, gateway node(s), and user station with time synchronization.

(ii) The network has at least one gateway node which has a stable link with the user station.

(iii) The network implements a routing protocol which provides the path between sensor nodes and gateway node(s).

(iv) The hierarchies of sensor nodes and the topology of the network are taken care of by the routing protocol.

(v) Each sensor node can store a set of predefined data.

3.2. Security Requirements. The most important aspect of the proposed user authentication scheme is its security. Therefore, it is vital to precisely define the security requirements that the proposed scheme must satisfy. This section defines the threat model and then describes the list of security requirements considered in this paper.

3.2.1. Threat Model. In the analysis of the proposed protocol, the widely used Doley and Yao [10] threat model will be used, which assumes that two communicating parties communicate over an insecure channel. This means that the attacker is able to eavesdrop and manipulate the messages sent over the air. In other words, the attacker can read, modify, and delay the messages sent by the different entities that participate in WSNs. Additionally, it is assumed that the attacker may know the algorithm of the security mechanisms that are deployed in the network. The main aim of the attacker is the falsification of an authentication. It is assumed that the attacker aims for a forgery in the proposed message exchange scenario where a user authenticates to the sensor network. The attack is considered successful if the sensor network (whether sensor node or gateway node) accepts a fake message that was not sent by an authentic user or if the user accepts messages coming from a fake sensor node or gateway node. All kinds of nonauthentic messages such as random new messages, replayed messages, and modified messages sent by the attacker are considered as fake messages.

This paper only takes care of the user authentication problem; other security issues of other layers of the protocol are taken care of by security solutions of other levels. In addition, it is assumed that gateway node(s) is managed by a trusted infrastructure. Therefore, the gateway node(s) is considered as secure and its security is not considered as part of this work. Additionally, not every aspect of physical attacks is considered in this paper; however, the unauthorized extraction of the secret values of a sensor node or smart card using techniques such as shown in [11–13] must not have effect on the security of the rest of nodes and users.

3.2.2. Basic Requirements

Data Confidentiality. Data confidentiality is the most important issue in network security. The proposed security solution must provide concealment of private information making it infeasible for an unauthorized user to understand the confidential data.

Data Integrity. With the implementation of confidentiality, an adversary may be unable to read the information. However, this does not mean the data is safe. The adversary can change the data to produce disorder in the sensor network. Therefore, it is important that different entities of the network can detect modification of messages transmitted over the network.

Data Freshness. Even if confidentiality and data integrity are assured, it is necessary to ensure the freshness of each message. Data freshness suggests that the data is recent and it ensures that no old messages have been replayed.

3.2.3. Attack List. There are several attacks that have been considered common in user authentication for WSNs such as privileged-insider, stolen-verifier, replay, parallel session, guessing, brute force, impersonation, and gateway-node bypassing attacks [3–7]. The proposed solution must also demonstrate its security against such attacks.

3.2.4. Other Security Requirements

Mutual Authentication. Some of previous works [4, 7] provide mutual authentication between the gateway node and sensor nodes but do not provide mutual authentication between user and gateway node. This situation can compromise the security of the whole network because newer sensor network implementations offer remote administration/query features in their 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 the gateway node from the user’s side to avoid adversaries collecting valuable data using fake gateway nodes.

Secure Registration, Authentication, and Password Change Processes. It is important to remember that the proposed user authentication mechanism must offer user registration, authentication, and password change processes and they must be executed in a secure manner.

Session Key Establishment. After authentication, the scheme must provide a simple session key establishment algorithm to provide a secure channel between entities after authentication.

3.3. Performance Requirements. WSNs own special characteristics which must also be considered in the designing of the security mechanisms. The most important resource constraints of WSN are the resource related ones [16], namely, limited storage space, limited computation power, and low energy capacity. Since the use of security algorithms reduces the lifetime of nodes, it is critical that the user authentication algorithm uses low quantity of energy. The aspects to be considered as performance requirements are as follows.
Type of Cryptographic Algorithm. Traditionally, there are two different types of cryptography algorithms, namely, public-key cryptography and symmetric-key cryptography. The first one has the characteristic of using two separate keys: one to encrypt the plaintext and another to decrypt the ciphertext. The most representative algorithms in public-key cryptography are Rivest-Shamir-Adleman ((RSA) the creators of the algorithm) and elliptic curve cryptography (ECC). On the other hand, the symmetric-key cryptography uses a common key for encryption and decryption which is shared among the communicating parties. One of the most representative algorithms of symmetric-key cryptography is advanced encryption standard (AES). Additionally, there is another type of encryption/decryption mechanism based on hash functions and exclusive-or operations which is considered as part of symmetric-key cryptography because it uses a common key between the communicating entities. One of the most representative hash functions used in such mechanisms is the secure hash algorithm (SHA). The importance of selecting a correct cryptographic algorithm lies in its complexity. A different Diffie-Hellman algorithm mean different complexities which is reflected in the required computation power and energy usage. This issue is very important because the common sensor nodes have very limited computation power and energy capacity. Several research works [17–20] have applied the asymmetric-key algorithms in WSNs. The results of such works reveal that despite the use of energy efficient techniques, such as ECC or dedicated cryptography coprocessors, asymmetric-key algorithms consume more energy than symmetric-key algorithms. For this reason, many researchers believe that the processing time and power consumption make it undesirable for public key algorithm techniques to be employed in sensor networks. Based on this criterion, the public cryptography was discarded for the proposed solution.

On the other hand, between the traditional symmetric key cryptography and encryption using hash functions, this paper has considered to use the last one because of its benefits in terms of energy usage as shown in previous works such as [21, 22]. In case of [21], the authors explain how AES-128 consumes more than double of SHA-1 function showing that the last one consumes around 154 $\mu$J while AES-128 consumes 339 $\mu$J on CrossBow nodes. The work described in [22] shows how for payloads of 17 bytes or above SHA-1 requires considerably less iterations than AES and therefore a shorter running time and less energy. In another earlier publication [16], the authors estimated that the energy per bit consumed by MIPS R4400 and MC68328 “DragonBall” processors for performing AES encryption/decryption operations is 9 nJ/bit and 101 nJ/bit, respectively, while for SHA-1 hashing function, the same processors consume 7.2 nJ/bit and 41 nJ/bit, respectively.

Number of Cryptographic Operations. The number of cryptographic operations used in sensor nodes must be minimal to extend their lifetime. The number of cryptographic operations executed by the user station and gateway node is not considered as important because they own superior resource capacities (i.e., computation power, storage, and energy capacity).

Number of Messages. One of main the operations that consume more energy is the transmission of messages. Therefore, the number of messages sent by sensor nodes must also be minimal to reduce their energy spent in wireless communication.

Energy Consumption Analysis. Energy consumption analysis using realistic data sizes will help to understand the effects of the proposed user authentication scheme over WSN.

3.4. Usability Requirements. The proposed scheme must also consider the different use cases of the authentication process according to the state of the network to offer extensive usability. This paper considers two specific use cases.

3.4.1. Online User Authentication. Online user authentication refers to the user authentication process executed when the sensor node has network connection with the gateway node (see Figure 5). This is considered the most traditional because sensor nodes are commonly monitored from the fixed network infrastructure installed in a safe place which includes the gateway node.

3.4.2. Offline or Gateway-Less User Authentication. This case is when the user authenticates directly to the sensor node because the network connection between the sensor node and gateway node cannot be established. In this case, the user must approach to the sensor node to authenticate with it but without the authorization of the gateway node (see Figure 6).

Important. Although this use case is very important for different types of WSNs such as military operation and natural disaster monitoring applications (as explained in Section 2), none of the previous works have considered this use case.
4. Proposed User Authentication Scheme

The proposed solution is composed of three protocols: user registration, user authentication, and password change protocols which are executed among three independent entities, that is, users, gateway node, and sensor nodes.

4.1. User Registration Protocol. The user registration protocol is executed when a new user needs to be authorized to access the sensor network. The steps executed in this protocol are as follows (see Figure 7 and Table 2). 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 $PPW_i = h(PW_i) \oplus r_i$, where $h(\cdot)$ is a hash function and $\oplus$ is an XOR operator. Once $PPW_i$ has been calculated, $ID_i$ and $PPW_i$ are sent to the gateway node $GW$. $GW$ then computes $N_i = h(ID_i \parallel PPW_i) \oplus h(ID_i \parallel K)$, where $K$ is a symmetric key only known by $GW$ and "$\parallel"$ is a concatenation operator. Once $N_i$ has been calculated, $GW$ personalizes a smart card with the parameters $h(\cdot)$ and $N_i$. Finally, $GW$ delivers the smart card to $U_i$ in a secure manner.

On the other hand, a unique secret key $K_{sn} = h(S_n \parallel K)$ is stored in each sensor node responsible for exchanging data with $U_i$, where $S_n$ is the unique identification of the sensor node.

The proposed user registration protocol includes several enhancements compared to the previous solutions. First, each smart card and sensor node includes unique secret values: $N_i = h(ID_i \parallel PPW_i) \oplus (ID_i \parallel K)$ and $K_{sn} = h(S_n \parallel K)$, respectively, which limits the effect of the physical attack to the attacked unit, maintaining the security of the rest of the network. Additionally, the usage of $PPW_i$ (instead of $PW_i$) provides protection against the privileged-insider...
attack. Finally, the usage of \( h(ID_i \| K) \) inside \( N_i \) delivers protection against parallel session attacks.

4.2. User Authentication Protocol. This protocol is performed when \( U_i \) needs to access the data gathered by a sensor node. The process of user authentication is differentiated depending on the use case as described in Section 3.

4.2.1. Online User Authentication Protocol. This subsection describes the authentication protocol when the sensor node is connected to the gateway node (see Figure 5). In this case, the gateway node works as a verifier to validate the authenticity of the user and sensor node. The steps executed in this protocol are as follows (see Figure 8 and Table 2).

**Authentication Phase.** \( U_i \) inserts the smart card and inputs his/her ID and PW. The smart card then computes \( M_i^* = h(ID_i \| PW_i \| r_i) \) 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 \( PW_i = h(PW) \oplus r_i \) and \( K_i = N_i \oplus h(ID_i \| PW_i) \), and \( A_i = h(ID_i \| K_i \| T_i) \) and transmits \( \{ID_i, A_i, T_i\} \) to GW, where \( T_i \) is the current timestamp of U_i’s system. Upon receiving the login request at time \( T_i^* \), GW validates \( T_i \). If \( (T_i^* - T_i) > \Delta T \), GW aborts the authentication process, where \( \Delta T \) denotes the maximum allowed communication delay. Otherwise, GW computes \( K_i^* = h(ID_i \| K) \) and \( A_i^* = h(ID_i \| K_i^* \| T_i) \). Once \( A_i^* \) is computed, GW checks whether it is equal to \( A_i \). If \( A_i^* \) is different to \( A_i \), GW finishes the authentication process; otherwise, GW computes \( B_i^* = h(ID_i \| K_i^* \| T_2) \) and transmits the message \( \{ID_i, B_i, T_2\} \) to some nearest sensor node \( S_n \) to respond to the query with the data that \( U_i \) is looking for, where \( T_2 \) is the timestamp of GW’s system when sending the message. \( S_n \) first validates \( T_2 \) using similar method of \( T_1 \) verification then computes \( B_i^* = h(ID_i \| K_i^* \| T_2) \) and checks whether \( B_i^* \) is equal to \( B_i \). Only if those values match, \( S_n \) responds to GW’s message by sending the message \( \{C_i, T_3\} \), where \( C_i = h(ID_i \| K_i \| T_3) \). Once GW receives the message, validates \( T_3 \) using similar method of \( T_1 \) and \( T_2 \) verification and then checks the validity of \( C_i \) by comparing \( C_i^* = h(ID_i \| K_i^* \| T_3) \) with the received \( C_i \). Once the sensor node is authenticated, GW computes \( D_i = h(ID_i \| K_i^* \| T_4) \), where \( T_4 \) is the current timestamp of GW and sends the message \( \{D_i, T_4\} \) to \( U_i \). Finally, \( U_i \) verifies the message \( T_4 \) and computes \( D_i^* = h(ID_i \| K_i \| T_4) \) and checks whether it is equal to \( D_i \).

**Session Key Establishment Phase.** A session key between \( U_i \) and GW \( K_{G,GW} = h(T_1 \| T_3 \| K) \) and a session key between GW and \( S_n \) \( K_{S,GW} = h(T_2 \| T_3 \| K_n) \) could be used.
if an encryption channel was required after authentication. Additionally, if a direct communication channel between \( U_i \) and \( S_n \) was required, a bilateral session key \( K_{U_i,S_n} \) could be established through GW. In this case, GW would generate a random \( K_{U_i,GW} \) and send \( T_5 \parallel K_{U_i,GW} \) encrypted with \( K_{U_i,GW} \) to \( U_i \) and \( T_5 \parallel K_{U_i,S_n} \) encrypted with \( K_{S_n,GW} \) to \( S_n \), where \( T_5 \) is the timestamp of GW when sending the session key.

**Note.** Although the process of the online user authentication protocol is similar to that proposed by the existing works, it includes several enhancements. First, it delivers mutual authentication among all entities (\( U_i \), \( GW \), and \( S_n \)). Additionally, the proposed protocol includes a simple session key establishment phase which was not provided in most of solutions.

4.2.2. Offline or Gateway-Less User Authentication Protocol
In this use case, \( S_n \) is disconnected from the gateway node; therefore, it is not possible to receive the authorization from GW to provide access to the collected data to \( U_i \) (see Figure 6). In this situation, another form of authentication is required, where \( S_n \) must be sure that the \( U_i \) is an authenticated user authorized by GW without dealing directly with \( U_i \)'s key; on the other hand, \( U_i \) also must be sure that \( S_n \) is an authentic sensor node, but without dealing with \( S_n \)'s key. On that point, a special data called *Permit* is proposed to be used instead of the secret keys. *Permit* contains the authorization of GW to access the data of a sensor node.

The offline or gateway-less user authentication protocol is composed of two subprotocols, namely, permit issue and User Authentication. The first one allows \( U_i \) to request the permit for accessing \( S_n \) (\( Permit_{in} \)) to the gateway node, while the second one allows \( U_i \) to use the issued permit to authenticate to \( S_n \).

(A) Permit Issue Subprotocol. After receiving the smart card from GW, \( U_i \) can use it to authenticate himself or herself to GW and receive the \( Permit_{in} \) which will allow \( U_i \) to authenticate to the sensor node \( S_n \) when it is disconnected from GW. \( Permit_{in} \) is a secret value issued by GW, which certifies that \( U_i \) has permission to access the data collected by \( S_n \). The permit issue subprotocol is executed as follows (see Figure 9 and Table 2).

\( U_i \) inserts the smart card and inputs his/her \( ID_i \) and \( PW_i \). The smart card then computes \( M_i^* = h(ID_i \parallel PW_i \parallel K_i \parallel T_1) \).
r_j) and compares M_j with M_j to authenticate U_j. If those values do not match, the authentication request is rejected. Otherwise, the smart card computes PPW_j = h(PW_j) r_j, K_j = N_j h(ID_j PPW_j), and A_j = h(ID_j K_j T_j) and transmits {ID_j, A_j, T_j} to GW, where T_j is the current timestamp of U_j’s system. Upon receiving the login request at time T_j, GW validates T_j. If (T_j - T_j) > ΔT, GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW computes K_j = h(ID_j || K) and A_j = h(ID_j || K_j) T_j). Once A_j is computed, GW checks whether it is equal to A_j. If A_j is different to A_j, GW finishes the permit issue process; otherwise, GW computes K_j = h(S_n || K), Permit_in = h(ID_j || K_j || T_permit) K_j, K_U-GW = h(T_j || T_permit || K_j), and encrypted permit EP_j = E_kU-GW(Permit_in || T_permit) and transmits {EP_j, T_permit} to U_j, where T_permit is the timestamp of GW’s system when generating Permit_in and E_secret_key(string) is a symmetric encryption of a string using the key secret_key. U_j first validates T_permit received via a similar method of T_j verification and then computes K_U-GW = h(T_j || T_permit || K_j) to obtain Permit_in || T_permit = D_kU-GW(EP_j), where D_secret_key(string) is a symmetric decryption of a string using the key secret_key. Once T_permit is obtained, U_j compares it with the T_permit received in plaintext to validate once more the validity of EP_j. Finally, U_j stores Permit_in and T_permit into the smart card.

The criterion for checking the validity of ΔT_permit is the same for both cases, that is, a period of time (e.g., one day, one week, etc.). The first case provides the advantage that the user can mobilize to a location near sensor nodes but without network connection with the GW. However, in this case, the user must be prudent in updating the Permit before its expiration. The second case can be used when the user stays in a location with stable network connection with the GW. We believe that obtaining the Permit from the GW when the need arises is not a critical problem because the time required in executing the permit issue sub-protocol is small. In conclusion, taking into account the previous analysis, the first case (obtaining the Permit immediately) could be a better solution, if the response time to the events was critical and the user was not in a place connected to the GW. On the other hand, the second case (obtaining the Permit from the GW when the need arises) could be better when the user is located in a place with stable network connection with the GW.

(B) User Authentication Subprotocol. Once the Permit received, U_j can go to the field to authenticate directly to the sensor node. This protocol is composed of two phases, namely authentication and session key establishment (see Figure 10 and Table 2).

Authentication Phase. Once the Permit is issued from GW, U_j can use it to authenticate himself or herself to S_n and access S_n’s data. The authentication process is executed as follows. U_j inserts the smart card and inputs his/her ID_j and PW_j. The smart card then computes M_j = h(ID_j || PW_j |
delay and \( \Delta T \) by comparing to the smart card. The smart card then calculates \( r_i \) 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 \( PPW_i = h(PW_i) \oplus r_i \), \( K_i = N_i \oplus h(ID_i \parallel PPW_i) \), and \( P_i = h(Permit_{pn} \oplus K_i \parallel T_i) \) and transmits \( \{ID_i, P_i, T_i, T_{perm} \} \) to \( S_n \), where \( T_{perm} \) is the current timestamp of \( U_i \)'s system. Upon receiving the login request at time \( T_i^* \), \( S_n \) validates \( T_i \) and \( T_{perm} \). If \( (T_i^* - T_i) > \Delta T \) or \( (T_i^* - T_{perm}) > \Delta T_{perm} \), \( S_n \) aborts the authentication process, where \( \Delta T \) denotes the maximum allowed communication delay and \( \Delta T_{perm} \) denotes the validity of a permit. Otherwise, \( S_n \) computes \( P_i^* = h(h(ID_i \parallel K_n \parallel T_{perm}) \parallel T_i) \). Once \( P_i^* \) is computed, \( S_n \) checks whether it is equal to \( P_i \). If \( P_i^* \) is different to \( P_i \), \( S_n \) finishes the authentication process; otherwise, \( S_n \) computes \( Q_i = h(h(ID_i \parallel K_n \parallel T_{perm}) \parallel T_2) \) and transmits \( \{Q_i, T_2 \} \) to \( U_i \), where \( T_2 \) is the actual timestamp of \( S_n \)'s system. \( U_i \) first validates \( T_2 \) using similar method of \( T_1 \) verification then checks the validity of \( Q_i \) by comparing \( Q_i^* = h(Permit_{pn} \oplus K_i \parallel T_2) \) with the received \( D_2 \). If those values match, then \( U_i \) trusts \( S_n \) and accesses its data.

Session Key Establishment. A session key between \( U_i \) and \( S_n \), \( K_{U_iS_n} = h(T_1 \parallel T_2 \parallel h(ID_i \parallel K_n \parallel T_{perm})) \) could be used if an encryption channel was required after authentication.

4.3. Password Change Protocol. One of the requirements of a secure protocol is the delivery of a mechanism to users so that they can freely change their passwords. The proposed scheme provides a simple and efficient password change protocol which does not require communication with the gateway node. The proposed password change protocol is executed as follows (see Figure 11 and Table 2).

\( U_i \) inputs his/her \( ID_i, PW_i \), and new password \( NewPW_i \) to the smart card. The smart card then calculates \( M_i^* = h(ID_i \parallel PW_i \parallel r_i) \) 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 \( PPW_i = h(PW_i) \oplus r_i \), \( h(ID_i \parallel K) = N_i \oplus h(ID_i \parallel PPW_i) \), \( NewM_i = h(ID_i \parallel NewPW_i \parallel r_i) \), and \( NewN_i = h(ID_i \parallel NewPPW_i) \oplus h(ID_i \parallel K) \), where \( NewPPW_i = h(NewPW_i) \oplus r_i \). Finally, the smart card replaces \( M_i \) and \( N_i \) with \( NewM_i \) and \( NewN_i \), respectively.

4.4. Implementation Issue. This paper does not consider the details related to the real implementation of the proposed scheme. However, we would like to share several ideas required on implementing the proposed scheme. First, the user interface of the application to be installed in the user station (\( U_i \)) managing the execution of protocols must include the option to choose the protocol to be executed by the user. Second, as the content of the messages sent from \( U_i \) to \( GW \) in online user authentication protocol and offline permit issue sub-protocol is the same (\( \{ID_i, A_i, T_i\} \)), it must include a header indicating which protocol \( GW \) must execute; one possible solution could be the usage of 1 bit: “0” for online user authentication and “1” for offline permit issue sub-protocol. Finally, the requests \( \{ID_i, B_i, T_i\} \) and \( \{ID_i, P_i, T_i, T_{perm} \} \) used in online and offline user authentication protocols are differentiated by \( S_n \) using messages’ sizes. This means that \( S_n \) would execute the steps of one protocol according to the size of the receiving message. As another option, request messages could include a header to indicate the type of the request the user is asking for. The mentioned ideas are just possible options and we believe that the details required for real implementation should be considered in the implementation based on the requirements of the total solution.

5. Analysis of the Proposed Scheme

5.1. Security Analysis. This part analyzes the security of the proposed scheme in terms of formal verification and analysis of security requirements described in Section 3. The registration and password change protocols of the proposed scheme are excluded from this analysis because they are executed in a secure environment. In the analysis of the user authentication protocol, the threat model discussed in Section 3 is applied.

5.1.1. Formal Proof Based on BAN Logic

(A) Notations and Rules of BAN Logic. This subsection demonstrates the security of the proposed scheme by a well-known formal model called BAN logic [23, 24]. BAN logic has been widely used in different works such as [25–27] to reason about their security validation. The logical notations of BAN logic used in this paper are as described in Table 3. This section also lists 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 or combined (hashed) under \( K \) then
the principal $P$ believes that the principal $Q$ once said the statement $X$:

$$
rac{P :: P \rightarrow K, P \not\in \{X\}_K}{P :: Q \not\in X}, \quad \frac{P :: P \rightarrow K, P \not\in \{X\}_K}{P :: Q \not\in X}.
$$

\[ (1) \]

**Freshness-Conjunctatenation Rule.** Provided that the principal $P$ believes freshness of the statement $X$, the principal $P$ believes freshness of the $(X, Y)$:

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

\[ (2) \]

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

$$
\frac{P \equiv \#(X), P \equiv Q \not\in X}{P \equiv Q \equiv X}.
$$

\[ (3) \]

**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 $X$:

$$
\frac{P \equiv Q \Rightarrow X, P \equiv Q \equiv X}{P \equiv X}.
$$

\[ (4) \]

**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:

$$
\frac{P \equiv X, P \equiv Y}{P \equiv (X, Y)}, \quad \frac{P \equiv (X, Y)}{P \equiv X},
$$

$$
\frac{P \equiv Q \equiv X, P \equiv Q \not\in (X, Y)}{P \equiv Q \equiv X}, \quad \frac{P \equiv Q \not\in (X, Y)}{P \equiv Q \not\in X}.
$$

\[ (5) \]

**(B) Formal Proof.** In the following, it shows the security proof of the authentication protocol using the BAN logic.

**(1) Formal Proof of Online User Authentication.** This part presents the formal proof of the online user authentication protocol, when sensor nodes have active communication link with the gateway node.

In this case, the protocol must satisfy the following goals:

$$
GW \equiv A_1, \quad S_n \equiv B_1, \quad GW \equiv C_1,
$$

\[ (G.1) \quad (G.2) \quad (G.3) \]

$$
U_i \equiv D_i, \quad U_i \equiv K_{U_i, GW}, \quad GW \equiv K_{U_i, GW},
$$

\[ (G.4) \quad (G.5) \quad (G.6) \]

$$
GW \equiv K_{S_n, GW}, \quad S_n \equiv K_{S_n, GW}.
$$

\[ (G.7) \quad (G.8) \]

First, messages of the protocol are transformed to the idealized form as shown below:

$$
ID_j \rightarrow GW:
$$

\[ (M.1) \]

$$
ID_j, (ID_j \parallel h(ID_j \parallel K) \parallel T_1)_{h(ID_j \parallel K)}, T_1,
$$

\[ (A.1) \]

$$
S_n \leftarrow GW:
$$

\[ (M.2) \]

$$
ID_j, (ID_j \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}, T_2,
$$

\[ (A.2) \]

$$
S_n \rightarrow GW: (ID_j \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}, T_3,
$$

\[ (M.3) \]

$$
U_i \leftarrow GW:
$$

\[ (M.4) \]

$$
(ID_j \parallel h(ID_j \parallel K) \parallel T_4)_{h(ID_j \parallel K)}, T_4.
$$

Second, assumptions about the initial state of the protocol are defined. Those assumptions are listed below:

$$
GW \equiv \#(T_1), \quad S_n \equiv \#(T_2), \quad GW \equiv \#(T_3),
$$

\[ (A.3) \quad (A.4) \]

$$
U_i \equiv \#(T_4), \quad U_i \leftarrow U_i \leftarrow GW, \quad GW \equiv U_i \leftarrow GW,
$$

\[ (A.5) \quad (A.6) \]

$$
S_n \leftarrow S_n \leftarrow GW, \quad GW \leftarrow S_n \leftarrow GW,
$$

\[ (A.7) \quad (A.8) \]

$$
GW \equiv S_n \leftarrow GW, \quad GW \equiv U_i \leftarrow A_1, \quad S_n \equiv GW \equiv B_1,
$$

\[ (A.9) \quad (A.10) \]

$$
GW \equiv S_n \equiv C_1, \quad U_i \equiv GW \equiv D_i.
$$

\[ (A.11) \quad (A.12) \]
Finally, the proof steps to the idealized form of the proposed protocol are performed based on BAN logic rules and assumptions. The proof steps are as in Table 21.

The proposed goals were reached by (S.5), (S.10), (S.15), (S.20), (S.21), (S.22), (S.23), and (S.24). In summary, this paper has demonstrated how the proposed online user authentication protocol is secure and provides mutual authentication among $U_i$, $GW$, and $S_n$.

(2) Formal Proof of Offline or Gateway-Less User Authentication

(a) Permit Issue Subprotocol. In the following, there is the security proof of the proposed permit issue sub-protocol. In this sub-protocol, goals to be reached are as follows:

$$GW \equiv A_i,$$  \hspace{1cm} (G.1)

$$U_i \equiv \text{Permit}_{in}.$$  \hspace{1cm} (G.2)

First, messages of the protocol are transformed to the idealized form as shown below:

$$U_i \rightarrow GW : \quad ID_i, (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}' T_1, \quad (M.1)$$

$$S_n \leftarrow GW : \{\text{Permit}_{in}, T_{permit}\}_{K_{U_i, GW}}, T_{permit} \quad (M.2)$$

Second, the following assumptions are made about the initial state of the sub-protocol to analyze the proposed scheme:

$$GW \equiv \#(T_1), \quad (A.1)$$

$$U_i \equiv T_{\text{permit}}, \quad (A.2)$$

$$U_i \equiv h(ID_i \parallel K), \quad (A.3)$$

$$GW \equiv U_i \rightarrow GW, \quad (A.4)$$

$$GW \equiv U_i \rightarrow A_i, \quad (A.5)$$

$$U_i \equiv GW \rightarrow \text{Permit}_{in}, \quad (A.6)$$

Finally, proof steps to the idealized form of the proposed sub-protocol are performed based on the BAN logic rules and the assumptions as in Table 22.

The proposed goals were reached by (S.5) and (S.10). In summary, this paper has demonstrated how the proposed permit issue sub-protocol is secure and provides mutual authentication between $U_i$ and $GW$.

(b) User Authentication Subprotocol. In the following, there is the security proof of the proposed user authentication sub-protocol using the BAN logic. The sub-protocol will satisfy the following goals:

$$S_n \equiv P_i, \quad (G.1)$$

$$U_i \equiv Q_i, \quad (G.2)$$

$$S_n \equiv K_{U_i, S_n}, \quad (G.3)$$

$$U_i \equiv K_{U_i, S_n}, \quad (G.4)$$

First, here is the transformation of protocol’s messages to the idealized form:

$$U_i \rightarrow S_n : \quad ID_i, (ID_i \parallel h(ID_i \parallel K) \parallel T_{\text{permit}}) \parallel T_1)_{h(ID_i \parallel K, \text{Permit})} \quad (M.1)$$

$$S_n \leftarrow S_n : \quad ID_i \parallel h(ID_i \parallel K) \parallel T_{\text{permit}}) \parallel T_2)_{h(ID_i \parallel K, \text{Permit})} \quad (M.2)$$

Second, the following assumptions are made about the initial state of the scheme to analyze the proposed scheme:

$$S_n \equiv \#(T_1), \quad (A.1)$$

$$U_i \equiv \#(T_2), \quad (A.2)$$

$$U_i \equiv h(ID_i \parallel K, \text{Permit}) \quad (A.3)$$

$$S_n \equiv U_i \rightarrow P_i, \quad (A.4)$$

$$U_i \equiv S_n \rightarrow Q_i, \quad (A.5)$$

Finally, the proof steps to the idealized form of the proposed sub-protocol are performed based on the BAN logic rules and the assumptions as in Table 23.

The proposed goals were reached by (S.5), (S.10), (S.11), and (S.12). In summary, this paper has demonstrated how the proposed user authentication sub-protocol is secure and provides mutual authentication between $U_i$ and $S_n$.

5.1.2. Security Verification. This subsection analyzes the security of the proposed solution from the point of view of basic security requirements mentioned in Section 3. This section also analyzes how the proposed solution is secure against possible attacks. As mentioned previously, the paper assumes that the communication channels used in user authentication are insecure and that there exists an attacker who can intercept all messages exchanged among $U_i$, $GW$, and $S_n$.

In addition, the paper assumes 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.

(A) Security Analysis of Basic Requirements

Confidentiality. Confidentiality of messages is guaranteed by usage of secret values. Communication between $U_i$ and $GW$ is encrypted using $K_i = h(ID_i \parallel K)$ while communication between $GW$ and $S_n$ is encrypted using $K_n = h(S_n \parallel K)$. Additionally, the communication messages in offline user authentication are encrypted by using a value generated from $\text{Permit}_{in}$. Furthermore, session keys ($K_{U_i, GW}$, $K_{S_n, GW}$, and $K_{U_i, S_n}$) are generated to guarantee the transmission of confidential values after authentication.

Integrity. Integrity of messages transmitted during the authentication process is guaranteed by verification in each step of protocols. In each verification step, entities, that is, $U_i$, $GW$, and $S_n$, compare the received data with values autonomously calculated by them to confirm the authenticity of messages.

Freshness. In network security protocols, the freshness and uniqueness of messages allow the provision of a strong defense against replay attacks. These two properties are reachable by using a time variant parameter inside messages, such
as random numbers, sequence numbers, and timestamps, being most common the usage of random numbers (nones) and timestamps. This work has used the timestamp method because it is always more efficient in terms of number of communication rounds compared to the nonce-based counterpart [28].

(B) Security Verification from Possible Attacks

Privileged-Insider Attack. In the proposed solution, \(U_i\) transmits his/her pseudo-password \(PPW_i = h(PW_i) \oplus r_i\) 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, 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.

Stolen-Verifier Attack. One of the features of the proposed protocol is the absence of a password/verifier table in \(GW\) and \(S_n\). This feature prevents our solution from stolen-verifier attacks.

Replay Attack. Timestamps are used to avoid replay attacks. In each message, a different timestamp is used to guarantee its freshness.

Parallel Session Attack. In previous works, the attacker can obtain a valid authentication request message for the next timestamp because they use the XOR operation in a vulnerable way (see reference [8] for more details). In the proposed protocol, we have eliminated the possibility of parallel session attack by sending hash values instead of values resulting from XOR operations. Therefore, even though another legal user of the system eavesdrops on \(U_i\)’s message \(\{ID_i, A_i, T_i\}\) or \(\{ID_j, P_j, T_j, T_{perm}i\}\), he/she cannot obtain the secret values \(A_i\) or \(P_j\) for the next timestamp \(T_2\) because they are irreversible one-way hash values.

Guessing Attack. In the proposed scheme, secret values are never sent in plaintext but encrypted by a one-way hash function or symmetric cryptographic algorithm. Therefore, even when the adversary got \(A_j, B_j, C_j, D_j, EP_j, P_j\), or \(Q_j\), he or she could not guess any secret value (i.e., \(PW_i, K_i, K_n, or K\)) because of the secure property of the hash and symmetric cryptographic algorithms.

Gateway-Node Bypassing Attack. The reason for the possibility of a \(GW\) bypassing attack in [3, 6] is due to the sharing of secret parameter \(x_n\) with \(S_n\) and \(U_i\). If the value of \(x_n\) 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 [7] is due to the secret value \(x\) stored in sensor nodes which can be extracted using similar method of extracting \(x\) from a smart card [11-13]; if \(x\) is extracted, the adversary can execute the \(GW\) bypassing attack using \(DID_j = h(ID_j \parallel PW_j) \oplus h(x_n \parallel T_j)\) and \(A_f = h(DID_j \parallel S_n \parallel x_n \parallel T_j)\).

In the proposed protocol, \(U_i\)’s smart card and \(S_n\) do not store either \(x_n\) or \(x\) but instead store other individual secret values \(N_i = h(ID_j \parallel PPW_i) \oplus (ID_j \parallel K)\) and \(K_n = h(S_n \parallel K)\) which are unique per smart card and sensor node. Therefore, even if the \(K_n\) value was extracted from a sensor node, the rest of nodes will still maintain their security. On the other hand, even if the \(N_i\) were extracted from a smart card, the \(N_i\) would be unusable without the correct \(PPW_i\) value because the \(A_i\) used for authentication is generated by using the \(h(ID_j \parallel K)\) which can only be obtained with the correct \(h(ID_j \parallel PPW_i)\) value.

User Impersonation. An adversary who wants to impersonate a valid user \(U_i\) to log into the network must calculate a valid \(A_i\) (for the online and offline user authentication) or \(P_i\) (for the permit issue). Since \(A_i = h(ID_j \parallel K_i \parallel T_1)\) and \(P_i = h(Permit_{ij} \oplus K_i \parallel T_1)\) are calculated by a one-way hash function, the adversary cannot decipher such values. Additionally, \(A_i\) and \(P_i\) cannot be created arbitrarily because they are based on the secret value \(K_i\) obtainable only with the correct \(N_i\) and \(PW_i\).

Gateway-Node Impersonation. An adversary who wants to impersonate a valid \(GW\) must calculate a valid \(B_i\) (for the online user authentication) or \(EP_i\) (for the permit issue). Since \(B_i = h(ID_j \parallel K_n \parallel T_2)\) and \(EP_i = EP_{GW}(Permit_{ij} \parallel T_{perm})\) are calculated using one-way hash function or a secure symmetric cryptographic algorithm, the adversary cannot decipher such values. Additionally, \(B_i\) and \(EP_i\) cannot be created arbitrarily because they are based on the secret value \(K_n\) which is only known by the authentic \(GW\).

Sensor Node Impersonation. An adversary who wants to impersonate a valid \(S_n\) must calculate a valid \(C_{ij}\) (for the online user authentication) or \(Q_j\) (for the offline user authentication). Since \(C_{ij} = h(ID_j \parallel K_n \parallel T_2)\) and \(Q_j = h(ID_j \parallel K_n \parallel T_{perm}) \parallel T_2\) are calculated using one-way hash function, the adversary cannot decipher such values. Additionally, \(C_{ij}\) and \(Q_j\) cannot be created arbitrarily because they are based on the secret value \(K_n\) which is only known by the authentic \(GW\).

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. Assuming that the \(U_i\)’s smart card is not cloned, the proposed protocol successfully prevents this threat 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 \((A_i, B_i, or P_i)\) 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 timestamp. (2) On the other hand, an insider with a valid smart card can try to discover the secret values by performing brute-force attacks. However, the determination of those values is infeasible because they are stored using secure one-way hash functions. If higher level of protection for \(K\) was required, additional random numbers \(R_i\) and \(R_n\) could be added for the generation of \(K_i = h(ID_j \parallel K \parallel R_i)\) and \(K_n = h(S_n \parallel K \parallel R_n)\).
respectively, which would be stored secretly in the GW. By using those 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 in bits of $R_i$ and $R_n$.

(C) Security Verification of Other Security Requirements

Mutual Authentication. The proposed protocol provides both mutual authentication between $U_i$ and GW as well as between GW and $S_n$.

(i) Online User Authentication. (1) The mutual authentication between $U_i$ and GW is verified as follows. GW verifies the authenticity of $U_i$ by comparing $A_i$ sent by $U_i$ with the $A_i^*$ value calculated by itself. $A_i$ can only be computed by the authentic $U_i$ because it is based on the secret value $h(ID_i \parallel K)$ only calculable with the valid $N_i$ and $PW_i$ which are personal to each $U_i$. In the same way, $U_i$ verifies the authenticity of GW by comparing $D_i$ sent by GW with the $D_i^*$ value calculated by GW. $D_i$ can only be computed by the authentic GW because it is based on the secret value $K$ only known by the authentic GW. (2) On the other hand, the mutual authentication between GW and $S_n$ is verified as follows. $S_n$ verifies the authenticity of GW by comparing $B_i$ sent by GW with the $B_i^*$ value calculated by itself. $B_i$ can only be computed by the authentic GW because it is based on the secret value $K$. In the same way, GW verifies the authenticity of $S_n$ by comparing $C_i$ sent by $S_n$ with the $C_i^*$ value calculated 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 $S_n$.

(ii) Offline or Gateway-Less User Authentication/Permit Issue Subprotocol. The mutual authentication between $U_i$ and GW is verified as follows. GW verifies the authenticity of $U_i$ by comparing $A_i$ sent by $U_i$ with the $A_i^*$ value calculated by itself. $A_i$ can only be computed by the authentic $U_i$ because it is based on the secret value $h(ID_i \parallel K)$ only calculable with the valid $N_i$ and $PW_i$ which are personal to each $U_i$. On the other hand, $U_i$ verifies the authenticity of GW by comparing $T_{perm}$ sent in plaintext by GW with the $T_{perm}$ value contained inside $EP_j$. $EP_j$ can only be computed by the authentic GW because it is based on the secret value $K$ only known by GW.

(iii) Offline or Gateway-Less User Authentication/User Authentication Subprotocol. The mutual Authentication between $U_i$ and $S_n$ is verified as follows. $S_n$ verifies the authenticity of $U_i$ by comparing $P_i$ sent by $U_i$ with the $P_i^*$ value calculated by itself. $P_i$ can only be computed by the authentic $U_i$ because it is based on the secret value $Perm_{init} @ K_i$ only calculable by the authentic $U_i$. On the other hand, $U_i$ verifies the authenticity of $S_n$ by comparing $Q_i$ sent by $S_n$ with the $Q_i^*$ value calculated by $U_i$. $Q_i$ can only be computed by the authentic $S_n$ because it is based on the secret value $K_n$ only known by $S_n$.

Password Change Phase. Our proposal offers a lightweight 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 $S_n$.

(D) Summary of Security Features of the Proposed Solution

Table 4 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 attack, gateway-node bypassing attack), also provides a secure password change phase and 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.

5.2 Performance and Cost Analysis

5.2.1 Cryptographic Operations. Table 5 indicates the number of cryptographic operations required in each protocol per entity for online user authentication. 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, the additional operations are not an impediment for real implementation. Additionally, it is possible to say that the additional operations are justifiable considering that our scheme includes security features that previous works do not offer, which is indispensable for implementing a reliable

| Security feature                                      | Proposed | Das’s [3] | Nyang-Lee’s [4] | Huang et al.’s [5] | Chen-Shih’s [6] | Khan-Alghathbar’s [7] |
|------------------------------------------------------|----------|-----------|-----------------|-------------------|-----------------|----------------------|
| Secure password change/update                        | Yes      | No        | No              | Yes               | No              | Yes                  |
| Protection against insider’s attack                  | Yes      | No        | No              | Yes               | No              |                         |
| Protection against GW bypassing attack               | Yes      | No        | No              | No                | No              | Yes                  |
| Protection against parallel session attack           | Yes      | No        | Yes             | No                | No              | No                   |
| Mutual authentication between $GW$ and $S_n$         | Yes      | No        | Yes             | No                | No              | Yes                  |
| Mutual authentication between $U_i$ and GW           | Yes      | No        | Yes             | No                | No              | No                   |
| Session key establishment                            | Yes      | No        | Yes             | No                | No              | No                   |
| Offline user authentication                          | Yes      | No        | No              | No                | No              | No                   |

Table 4: List of enhanced security features of the proposed scheme.
5.2.2. Number of Messages. Analyzing the number of messages transmitted and received by sensor nodes is considered important because it affects the energy consumption of those devices. This paper only focuses on the protocols where the resource limited sensor nodes participate in (i.e., online and offline user authentication protocols). Table 7 shows the number of transmissions and receptions executed by different entities in online user authentication protocols in different proposals. It illustrates how the proposed protocol maintains equal or less number of messages than previous works to show its competitiveness.

Going into details of Table 7, the present work also presents the content of messages transmitted and received by different entities and their sizes in each solution (see Tables 8, 9, 10, 11, 12, and 13). This work has considered that the size of the different data inside of messages, such as user identification, timestamps, hash values, and login confirmation messages, is equal because the intention of this part is to analyze the number of data units in each protocol (analysis using more realistic data sizes is executed in Section 5.2.3). As you can see in Tables 8–13, the most lightweight solutions in terms of messages sizes in sensor nodes are those proposed by Das, Huang et al., and Chen-Shih. However, messages sizes in those protocols are reduced because they omit steps required to provide mutual authentication between the gateway-node and sensor nodes opening serious vulnerabilities. Therefore, even though they offer less communication, they cannot be considered as optimal solutions. On the other hand, Table 9 shows how Nyang-Lee's proposal has the highest communication overhead requiring 4 data units for transmission and 4 data units for reception in sensor nodes. Finally, Alghathbar's proposal and the proposed solution have the same communication overhead in sensor nodes with 2 data units in transmission and 3 data units in reception.

On the other hand, Table 14 shows the number of messages in offline user authentication. In this protocol, only one message exchange between $U_i$ and $S_n$ is executed, showing how it is efficient in this aspect. In case of $S_n$, it only requires the transmission and reception of 2 and 4 data units, respectively.

5.2.3. Energy Consumption. One of the most sensible limitations in sensor nodes is their energy capacity. Therefore, it is important to analyze the energy consumption in those devices. The energy consumptions of user station and gateway node have not been analyzed because they do not suffer from this aspect. Following the fact that the battery power of a sensor node is depleted by computational processing and radio consumption [30], this paper has calculated the energy consumption overhead caused by (1) cryptographic

| Subprotocol                | Entity | Number of operations |
|----------------------------|--------|----------------------|
| Permit issue subprotocol   | $U_i$  | $5h + 1se$           |
|                            | $GW$   | $5h + 1se$           |
| Authentication subprotocol | $U_i$  | $5h$                 |
|                            | $S_n$  | $3h$                 |

h: hash, se: symmetric encryption, and sd: symmetric decryption.
Table 7: Number of transmissions and receptions in online user authentication.

| Entity | Proposed | Das’ [3] | Nyang-Lee’s [4] | Huang et al’s [5] | Chen-Shih’s [6] | Khan-Alghathbar’s [7] |
|--------|----------|----------|-----------------|------------------|----------------|----------------------|
|        | Tx       | Rx       | Tx              | Rx               | Tx             | Rx                   |
| U_i    | 1        | 1        | 1               | 1                | 1              | 1                    |
| GW     | 2        | 2        | 2               | 2                | 2              | 2                    |
| S_n    | 1        | 1        | 1               | 1                | 1              | 1                    |

Tx: transmission; Rx: reception.

Table 8: Details of communication messages of Das’ proposal [3].

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {DID, C_i, T}     | {Login OK}     | 3          |
| GW     | {DID, A_i, T}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}        |                | 1          |

Table 9: Details of communication messages of Nyang-Lee’s proposal [4].

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {DID, C_i, T}     |                | 3          |
| GW     | {DID, C_i, T}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}        |                | 1          |

Table 10: Details of communication messages of Huang et al’s proposal [5].

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {DID, C_i, T}     |                | 3          |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}        |                | 1          |

Table 11: Details of communication messages of Chen-Shih’s proposal [6].

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {DID, C_i, T}     |                | 3          |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}        |                | 1          |

Table 12: Details of communication messages of Khan-Alghathbar’s proposal [7].

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {DID, C_i, T}     |                | 3          |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}        |                | 1          |

Table 13: Details of communication messages of the proposed scheme (online user authentication).

| Entity | Transmission (Tx) | Reception (Rx) | Total size |
|--------|-------------------|----------------|------------|
| U_i    | {ID_i, A_i, T_i}  |                | 3          |
| GW     | {ID_i, B_i, T_i}, {D_i, T_i} | {ID_i, A_i, T_i}, {C_i, T_i} | 5 |
| S_n    | {C_i, T_i}        |                | 2          |

Tаблица 7: Число передач и приема в онлайн- процедурах аутентификации пользователя.

| Entity | Предложенное | Дас [3] | Нянг-Лэй [4] | Хуанг и др. [5] | Чен-Ши [6] | Кхан-Альгатбар [7] |
|--------|--------------|---------|--------------|-----------------|------------|------------------|
|        | Tx | Rx | Tx | Rx | Tx | Rx | Tx | Rx | Tx | Rx |
| U_i    | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| GW     | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  |
| S_n    | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |

Tx: передача; Rx: прием.

Таблица 8: Подробности сообщений передачи и приема в предложении Дас [3].

| Entity | Передача (Tx) | Прием (Rx) | Общее размер |
|--------|--------------|------------|-------------|
| U_i    | {DID, C_i, T} | {Login OK} | 3           |
| GW     | {DID, A_i, T}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}   |            | 1           |

Таблица 9: Подробности сообщений передачи и приема в предложении Нянг-Лэй [4].

| Entity | Передача (Tx) | Прием (Rx) | Общее размер |
|--------|--------------|------------|-------------|
| U_i    | {DID, C_i, T} |            | 3           |
| GW     | {DID, C_i, T}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}   |            | 1           |

Таблица 10: Подробности сообщений передачи и приема в предложении Хуанг и др. [5].

| Entity | Передача (Tx) | Прием (Rx) | Общее размер |
|--------|--------------|------------|-------------|
| U_i    | {DID, C_i, T} |            | 3           |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}   |            | 1           |

Таблица 11: Подробности сообщений передачи и приема в предложении Чен-Ши [6].

| Entity | Передача (Tx) | Прием (Rx) | Общее размер |
|--------|--------------|------------|-------------|
| U_i    | {DID, C_i, T} |            | 3           |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}   |            | 1           |

Таблица 12: Подробности сообщений передачи и приема в предложении Кхан-Альгатбар [7].

| Entity | Передача (Tx) | Прием (Rx) | Общее размер |
|--------|--------------|------------|-------------|
| U_i    | {DID, C_i, T} |            | 3           |
| GW     | {DID, A_i, T’}, {Login OK} | {DID, C_i, T}, {Login OK} | 4 |
| S_n    | {Login OK}   |            | 1           |

Таблица 13: Подробности сообщений передачи и приема в предложенной схеме (в онлайн- процедурах аутентификации пользователя).
Table 14: Details of communication messages of the proposed scheme (offline user authentication).

| Entity | Number of messages | Transmission (Tx) | Reception (Rx) |
|--------|-------------------|------------------|---------------|
|        |                   | Content of messages | Total size | Content of messages | Total size |
| $U_i$  | 1                 | $\{ID_i, P_i, T_i, T_{perm} \}$ | 4 | 1 | $\{Q_i, T_i \}$ | 2 |
| $S_n$  | 1                 | $\{Q_i, T_j \}$ | 2 | 1 | $\{ID_i, P_i, T_i, T_{perm} \}$ | 4 |

Table 15: Energy consumption of cryptographic operations executing different online user authentication protocols.

| Scheme                          | Energy (mJ) | MIPS R4000 | MC68328 |
|---------------------------------|-------------|------------|---------|
| Proposed                        | 0.0041472   | 0.023616   |         |
| Das’ [3]                        | 0.0032256   | 0.018368   |         |
| Nyang-Lee’s [4]                 | 0.0067040   | 0.0011040  |         |
| Huang et al.’s [5]              | 0.0027648   | 0.0015744  |         |
| Chen-Shih’s [6]                 | 0.0032256   | 0.0018368  |         |
| Khan-Alghathbar’s [7]           | 0.0052992   | 0.0030176  |         |
| Shih’s schemes, respectively, and Login OK messages are 64 bits each, and the sizes of secret values such as ID of users, gateway node, and sensor nodes, and the sizes of random numbers are 160 bits, the sizes of timestamps, identifications (ID of users, gateway node, and sensor nodes), and Login OK messages are 64 bits each, and the sizes of secret values such as ID of users, gateway node, and sensor nodes, respectively. However, we believe that it is not an impediment for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

(A) Energy Consumption Overhead of Cryptographic Operations. First, for the calculation of energy used by cryptographic operations, this paper has used the energy consumption estimates indicated in [16]. In [16], the authors estimated that the energy per bit consumed by MIPS R4400 and MC68328 “DragonBall” processors for performing AES encryption/decryption operations is 9 nJ/bit and 41 nJ/bit, respectively, while for the SHA-1 hashing function the same processors consume 7.2 nJ/bit and 41 nJ/bit, respectively. Additionally, this work has assumed that the size of random numbers is 160 bits, the sizes of timestamps, identifications (ID of users, gateway node, and sensor nodes), and the sizes of secret values such as $x_a$, $x_i$, $x_n$, $K_n$ are 160 bits each. Using those values, the energy consumption of cryptographic operation in each scheme was calculated (see Table 15). Here is an example of how the energy consumption of cryptographic operation were calculated: in the proposed online authentication protocol, two hash values are calculated by $S_n$, that is, $B^*_i$ and $C_j$. As the input values for the generation of $B^*_i$ and $C_j$ are 288 bits each, the energy used in generating $B^*_i$ and $C_j$ will be 7.2 nJ/bit $\times$ 288 bits $= 0.002074$ mJ for MIPS R4400 processor and 41 nJ/bit $\times$ 288 bits $= 0.011808$ mJ for MC68328 processor. Therefore, the total energy used by the cryptographic operations in the proposed online user authentication protocol will be the summation of the energy consumptions in generating $B^*_i$ and $C_j$, that is, 0.0001472 mJ and 0.023616 mJ for MIPS R4400 and MC68328 processors, respectively. Table 15 shows how the proposed solution consumes 0.000092, 0.00138, and 0.000092 mJ more than M. Das, Huang et al., and Chen-Shih’s schemes, respectively, and consumes 0.012557 and 0.0001152 mJ less than Nyang-Lee and Khan-Alghathbar’s schemes, respectively, when using the MIPS R4400 processor. Using the data of the table, it is also possible to deduce that the proposed solution consumes 0.002097 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions. Similar conclusion can be reached with the energy consumption values when using the MC68328 microprocessor.

On the other hand, this work also has calculated the energy consumption of sensor nodes in executing cryptographic operations during the offline user authentication sub-protocol (see Table 16). It shows how this approach consumes 0.001152 and 0.000656 mJ more than the online user authentication using MIPS R4400 and MC68328 processors, respectively. However, we believe that it is not an impediment for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

(B) Energy Consumption Overhead of Radio Communication. For the calculation of energy consumption of radio communication produced by the proposed protocols, the present work has assumed a simple model where the radio dissipates $E_{\text{elec}} = 50$ nJ/bit to run the transmitter or receiver circuitry and $E_{\text{amp}} = 10$ pJ/bit/m$^2$ to run the transmitter amplifier (see Table 17), similar to the model used in [31–33]. Additionally, for the calculation of the energy consumption, this work has used the parameters indicated in Table 17 and has assumed that the implemented hash function is SHA1 (160 bits hash value), the sizes of random numbers are 160 bits, the sizes of timestamps, identifications (ID of users, gateway node, and sensor nodes), and Login OK messages are 64 bits each, and the sizes of secret values such as $x_a$, $x_i$, $x_n$, $K_n$ are 160 bits each.

Table 18 indicates the energy used by a sensor node in executing the online user authentication in different schemes.
Here we show an example of how such values were calculated. In the proposed online authentication protocol, the message \( \{ID_i, B_i, T_s\} \) is received and the message \( \{C_i, T_s\} \) is sent by \( S_n \). In this example, the total radio communication energy consumption is calculated by adding the energy used in reception and transmission of messages. As the receiving message's size is 288 bits, the radio communication energy used for reception will be \( 50 \text{nJ}/\text{bit} \times 288 \text{ bits} = 0.0144 \text{ mJ} \). On the other hand, as the transmitting message's size is 224 bits, the radio communication energy used for transmission will be \((50 \text{nJ} + 10 \text{pJ}/\text{bit/m}^2) \times 224 \text{ bits} = 0.0168 \text{ mJ}\). Then, the total radio communication energy used in the proposed online authentication protocol will be \(0.0144 \text{ mJ} + 0.0168 \text{ mJ} = 0.0312 \text{ mJ}\). Table 18 shows how the proposed solution consumes 0.0072 mJ more than M. Das, Huang et al.'s and Chen-Shih's schemes and less than Nyang-Lee's and Khan-Alghathbar's schemes, respectively. It also illustrates how the proposed solution consumes 0.000176 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions.

On the other hand, the offline user authentication sub-protocol only consumes 0.0032 mJ more than the proposed online user authentication protocol. We believe that this difference is acceptable considering the extraordinary situation when the offline user authentication is used.

(C) Total Energy Consumption Overhead. The total energy consumption overhead in sensor nodes can be calculated adding the energy consumption of cryptographic operations and radio communication. Table 19 shows the total energy consumption overhead of different schemes. It shows that the proposed solution consumes 0.008122, 0.008582, and 0.008122 mJ more than M. Das, Huang et al.'s, and Chen-Shih's schemes, respectively, and consumes 0.03816 and 0.00595 mJ less than Nyang-Lee's and Khan-Alghathbar's schemes, respectively, when using the MIPS R4000 processor. It also illustrates how the proposed solution consumes 0.00386 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions. Similar conclusion can be reached with the energy consumption of schemes when using the MC68328 microprocessor. It is important to say that the additional energy overhead from other works is justifiable considering that our scheme includes security features that previous works do not offer, which is indispensable for implementing a reliable and trustworthy network.

On the other hand, Table 20 shows the total energy consumption of authenticating sensor nodes executing the proposed offline user authentication sub-protocol. It shows that this approach consumes 0.004352 and 0.00976 mJ more than the online user authentication using MIPS R4000 and MC68328 processors, respectively. However, we believe that it is not an impediment for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

(D) Effect of Energy Consumption Overhead in the Wireless Sensor Network. Although there are differences among the energy consumption in different schemes, as they are small, it could not be neglected. Therefore, this paper has analyzed how much the energy consumption overhead of the proposed solution affects the lifetime of the sensor network. According to [16], one of the most common typical batteries in sensor nodes is the MN1500 Duracell AA with energy potential of 15.39 kJ. Based on this data, this work calculates how long a sensor node can survive executing the proposed user authentication scheme. In this simulation, we assume that the user authentication can use from 1% to 5% of the total energy while the rest of energy is used by other functionalities of the sensor node, such as path maintenance, data gathering, and data transmission.

Knowing that common sensor network applications are not dedicated for a massive user access yet, this work assumes that it is acceptable to use the average of one user authentication per minute as parameter. However, to understand how the proposed user authentication can act from higher demand, the present paper also considers the case when the average number of user authentication per minute is five. Figure 12 shows the number of months a sensor node with MIPS R4000 microprocessor can survive when executing 1 and 5 online user authentications per minute assuming that

| Scheme | Total energy (mJ) |
|--------|------------------|
| Proposed | 0.0396992 |
| MIPS R4000 | 0.0353472 |
| MC68328 | 0.0354816 |
| Das'[3] | 0.0272256 |
| Nyang-Lee’s [4] | 0.0735040 |
| Huang et al.’s [5] | 0.0267648 |
| Chen-Shih’s [6] | 0.0272256 |
| Khan-Alghathbar’s [7] | 0.0412992 |

Table 18: Energy consumption of radio communication in different protocols.

| Scheme | Energy (mJ) |
|--------|--------------|
| Proposed online/offline user authentication | 0.0312/0.0344 |
| Das’[3] | 0.0240 |
| Nyang-Lee’s [4] | 0.0568 |
| Huang et al.’s [5] | 0.0240 |
| Chen-Shih’s [6] | 0.0240 |
| Khan-Alghathbar’s [7] | 0.0360 |

Table 19: Total energy consumption overhead of authenticating sensor nodes executing different online user authentication protocols.

Table 20: Total energy consumption overhead of authenticating sensor nodes executing the proposed offline user authentication sub-protocol.
1% to 5% of the total energy of the battery is dedicated for user authentication. It shows that a sensor node can survive from 20.2 to 100.8 months depending on the number of authentications (1 and 5 per minute) using only 1% of the total energy of the sensor node. The same figure also indicates that the sensor node can survive up to 325 months using 5% of the total energy.

Figure 13 also shows the number of months a sensor node can survive executing 1 and 5 online user authentications per minute using 1% to 5% of the total energy and using the MC68328 microprocessor. It shows that a sensor node can survive from 13.0 to 65 months depending on the number of authentications per minute using only 1% of the total energy of the sensor node. The same figure also indicates that the sensor node can survive up to 325 months using 5% of the total energy.

Note. If high demand of user authentication (number of authentication request per minute) in online user authentication protocol was required, the following solution could be provided. It is possible to establish a predefined session period to maintain the authentication session between the gateway node and sensor node. So, if another user requests for authentication for such node, the authentication can be

---

**Table 21**

| Expression | Description |
|------------|-------------|
| $GW \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S1) by (M.1) |
| $GW \implies U_i \triangleright (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S2) by (S1), (A.6), and message-meaning rule |
| $GW \implies U_i \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S3) by (S2), (A.1), freshness-conjunction rule, and nonce-verification rule |
| $GW \triangleleft A_i$ | (S4) by (S3), (A.9), and jurisdiction rule. |

---

**Table 22**

| Expression | Description |
|------------|-------------|
| $GW \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S1) by (M.1) |
| $GW \implies U_i \triangleright (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S2) by (S1), (A.4), and message-meaning rule |
| $GW \implies U_i \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ | (S3) by (S2), (A.1), freshness-conjunction rule, and nonce-verification rule |
| $GW \triangleleft A_i$ | (S4) by (S3), (A.5), and jurisdiction rule. |
| $U_i \triangleleft \{Permit_m,T_{Remain}\}_{K_{U_{GW}}} \triangleleft IDs_T_{Remain}$ | (S6) by (M.2) |
| $U_i \triangleleft \{U_{GW} \rightarrow GW\}_{K_{U_{GW}}} \triangleleft U_i \triangleleft GW$ | (S7) by (M.2) and as $A_i = (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i K)}$ (G.1) |
| $U_i \triangleleft GW \rightarrow Permit_m$ | (S8) by (S7), (A.2), freshness-conjunction rule, and nonce-verification rule |
| $U_i \triangleleft Permit_m$ | (S10) by (S9), (A.6), and jurisdiction rule (G.2) |
executed only between the $U_i$ and $GW$ and reuse the connection between $GW$ and $S_n$ created before in the previous user authentication.

Figures 14 and 15 show the number of months for which a sensor node with MIPS R4000 and MC68328 microprocessor can survive when executing 1 and 5 offline user authentications per minute assuming that 1% to 5% of the total energy of the battery is dedicated for user authentication. It shows that the survival period of a sensor node (from 11 to 448.7 months) depends on the microprocessor, number of authentication per minute, and percentage of total energy dedicated for user authentication. Once again, we believe this durability is consistent for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.
6. Conclusion

With the increase of different types of sensor network implementations such as medical, ecology, and military operation applications, there have been many proposals which tried to give secure user authentication schemes for them. However, even though they deliver important advances in this area, they still incorporate serious vulnerabilities and limitations. In those circumstances, this paper proposes a user authentication mechanism which considers the security, performance, and usability factors. The security is guaranteed by an intensive analysis in terms of formal verification and analysis of possible attacks. The optimization of performance is achieved by using lightweight cryptography and in most of cases, only hash functions and XOR operations; additionally, the number of messages is reduced by using timestamps instead of challenge response of random nonces. Finally, the usability requirements are satisfied by considering both the online and offline user authentication use cases, the feature which was not considered in previous works.

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

Acknowledgment

This work was supported by the Sogang University Research Grant of 2011 (no. 2011I0026).

References

[1] M. H. Eldefrawy, M. K. Khan, K. Alghathbar, and E. S. Cho, “Broadcast authentication for wireless sensor networks using nested hashing and the chinese remainder theorem,” Sensors, vol. 10, no. 9, pp. 8683–8695, 2010.

[2] A. Hagedorn, D. Starobinski, and A. Trachtenberg, “Rateless Deluge: over-the-air programming of wireless sensor networks using random linear codes,” in Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN ’08), pp. 457–466, April 2008.

[3] M. L. Das, “Two-factor user authentication in wireless sensor networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086–1090, 2009.

[4] D. Nyang and M. Lee, “Improvement of Da’s two-factor authentication protocol in wireless sensor networks,” Cryptology ePrint archive 2009/631, 2012, http://eprint.iacr.org/2009/631.pdf.

[5] 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.

[6] 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.

[7] M. K. Khan and K. Alghathbar, “Cryptography and security improvements of two-factor user authentication in wireless sensor networks,” Sensors, vol. 10, no. 3, pp. 2450–2459, 2010.

[8] S. 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, 11 pages, 2012.

[9] G. Werner-Allen, K. Lorincz, M. Welsh et al., “Deploying a wireless sensor network on an active volcano,” IEEE Internet Computing, vol. 10, no. 2, pp. 18–25, 2006.

[10] 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.

[11] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Proceedings of the 19th International Advances in Cryptography 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, 2006, http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Jack.pdf.

[14] M. E. Raluca, M. E. 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, 2012, http://sine.ni.com/nips/ncds/view/p/lang/en/ nid/206919.

[16] D. W. Carman, P. S. Krus, and B. J. Matt, “Constraints and approaches for distributed sensor network security,” Tech. Rep. 00-010, NAI Labs, Network Associates, Glenwood, Md, USA, 2000.

[17] G. Bertoni, L. Breveglieri, and M. Venturi, “Power aware design of an elliptic curve coprocessor for 8 bit platforms,” in Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops ’06), pp. 337–341, March 2006.

[18] D. J. Malan, M. Welsh, and M. D. Smith, “A public-key infrastructure for key distribution in TinyOS based on elliptic
curve cryptography,” in Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON ’04), pp. 71–80, October 2004.

[19] K. Piotrowski, P. Langendoerfer, and S. Peter, “How public key cryptography influences wireless sensor node lifetime,” in Proceedings of the 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN ’06), pp. 169–176, October 2006.

[20] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in Proceedings of the Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops ’05), pp. 324–328, 2005.

[21] C. C. Chang, D. J. Nagel, and S. Muftic, “Balancing security and energy consumption in wireless sensor networks,” Lecture Notes in Computer Science, vol. 4864, pp. 469–480, 2007.

[22] J. P. Kaps and B. Sunar, “Energy comparison of AES and SHA-1 for ubiquitous computing,” Lecture Notes in Computer Science, vol. 4097, pp. 372–381, 2006.

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

[24] Y. G. Kim, C. J. Moon, D. W. Jeong, and D. K. Baik, “Formal verification of bundle authentication mechanism in OSGi service platform: BAN Logic,” International Journal of Software Engineering and Knowledge Engineering, vol. 16, no. 2, pp. 153–173, 2006.

[25] 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.

[26] 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.

[27] J. L. Tsai, T. C. Wu, and K. Y. Tsai, “New dynamic ID authentication scheme using smart cards,” International Journal of Communication Systems, vol. 23, no. 12, pp. 1449–1462, 2010.

[28] C. Gamage, J. Leiwo, and Y. Zheng, Timestamps for Network Authentication Protocols Revisited, 1999.

[29] R. Ying, “Building systems using software components,” Journal of Software Technology, vol. 9, no. 1, 2006.

[30] J. Park and S. Sahni, “An online heuristic for maximum lifetime routing in wireless sensor networks,” IEEE Transactions on Computers, vol. 55, no. 8, pp. 1048–1056, 2006.

[31] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, 2002.

[32] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, 2004.

[33] S. G. Yoo, S. H. Kang, and J. Kim, “SERA: a secure energy reliability aware data gathering for sensor networks,” Multimedia Tools and Applications, pp. 1–30, 2011.
