Research Article

An Improved User Authentication Protocol for Healthcare Services via Wireless Medical Sensor Networks

Muhammad Khurram Khan¹ and Saru Kumari²

¹ King Saud University, P.O. Box 92144, Riyadh 11653, Saudi Arabia
² Department of Mathematics, Agra College, Agra, Dr. B. R. A. University, Agra, Uttar Pradesh, India

Correspondence should be addressed to Muhammad Khurram Khan; mkhurram@ksu.edu.sa

Received 15 November 2013; Accepted 23 December 2013; Published 27 April 2014

1. Introduction

Healthcare service sector is one of the major applications of Wireless Sensor Networks (WSNs) acknowledged as Wireless Medical Sensor Network (WMSNs). It deploys tiny medical sensor-nodes (MS-nodes) on the body of the patient to sense crucial physiological signs which can be accessed and analyzed by registered medical professionals. Recently, Khan et al. analyzed Kumar et al.'s scheme proposed for healthcare applications using WMSNs and observed that the scheme is susceptible to many security weaknesses if an adversary extracts the information from the lost smart card of some user. The adversary can access patient’s physiological data without knowing actual password, can deceive medical professionals by sending fake information about patients, can guess the password of a user from the corresponding smart card, and so forth. Besides, the scheme fails to resist insider attack, lacks user anonymity and the session key shared between the user and the MS-node is insecure. To overcome these problems, we propose an improved user authentication scheme for healthcare applications using WMSNs. We show that the scheme is free from the identified weaknesses and excels in performance and efficiency scheme.

Healthcare sector is witnessing a transition from traditional human-labor-dependent services to technology-based smart services. This changeover is the outcome of Wireless Medical Sensor Networks (WMSNs), a transmission technology employed by medical professionals (like nurses, doctors, etc.) to obtain the information like blood pressure, pulse rate, body temperature, ECG of the patients. This is achieved by deploying tiny MS-nodes like blood pressure sensors, pulse oximeter, body temperature sensors, and ECG electrodes on the body of patient. The MS-nodes sense physiological information from patient’s body and then transmit it to the professionals in a wireless manner. Consequently, it cuts the cost of the human labor required for the purpose and facilitates the health professionals to observe and treat the patients as and when required. But patient’s personal medical data may be misused by adversaries like corrupt persons, personal enemies, health insurance professionals, and so forth. Thus, there is need for the security of WMSNs to ensure access to patient’s physiological information only to the authorized health professionals. Employing a user authentication scheme is a suitable method to achieve the desired security and establish a secure, efficient, and reliable healthcare environment via WMSNs.

After the development of simple user authentication schemes like [1–6], schemes for Wireless Sensor Networks (WSNs) [7–13] have also attracted a large community of researchers. Some work has also been proposed for healthcare applications using WSNs [14–17]. In 2012, Kumar et al. [18] observed that most of the schemes proposed for WSNs such as [9, 10, 12, 13] fall short to provide security and also require heavy computational load and high communication cost. They proposed a user authentication scheme using WMSNs for healthcare applications and called it an Efficient-Strong Authentication Protocol (E-SAP) [18]. They claimed that their scheme achieves mutual authentication between the user and the MS-node and also establishes session key between them. They found their scheme finer than other existing protocols concerning cost,
performance, and security. Subsequently, Khan et al. [19] identified that the scheme of Kumar et al. suffers from many security problems if an adversary extracts the information from the stolen smart card of some user. As a consequence, the scheme is exposed to user impersonation attack and insecure session key generation between user and MS-node. They showed that the scheme does not go with the authors’ claim as the mutual authentication between user and MS-node does not imply properly and an adversary can compute the session key to be established between. They also pointed out password guessing attack, insider attack, and MS-node impersonation attack on it. They found that if the identity of any user is revealed, it gives chance to many unauthorized/illegal persons to gain the personal medical data of patients and thereby generates problems for an authorized professional.

We feel that in addition to resist the prevalent threats, a user authentication scheme for WMSN should also provide user anonymity. Therefore, we propose a user anonymous authentication scheme using WMSNs eradicating the identified weaknesses of Kumar et al’s scheme. We aim to provide perfect mutual authentication and secure session key generation between the active participants of the authentication protocol in the scheme. The rest of the paper is arranged as the description follows for the subsequent sections. Section 2 briefly explains the architecture of WMSN and its benefit in healthcare applications. Kumar et al’s scheme is reviewed in Section 3. Section 4 gives review of the analysis of Kumar et al’s scheme by Khan et al. The proposed scheme is illustrated in Section 5 along with its security analysis and performance comparison in presented by Sections 6 and 7, respectively. To end with, Section 8 gives the conclusion of this paper. In this paper, we use professional and user interchangeably.

2. Architecture of WMSN and Its Benefits in Healthcare Services

The architecture of the Wireless Medical Sensor Network is depicted by Figure 1. There are four parties involved in the user authentication protocol employing WMSN as described below:

(i) Users: medical professionals like nurses, doctors, and so forth, looking for physiological data of the patient via WMSN.

(ii) MS-nodes: tiny sensors like temperature sensor, pulse oximeter, and so forth, deployed on the body of the patients.

(iii) GW-node: a powerful master node which plays the role of the registering authority and acts as an interface between the user and the MS-node.

(iv) Patients: they are under vigilance of medical professionals by means of MS-nodes for treatment.

First three participants are the active parties of the user authentication scheme. MS-nodes are tiny sensor having low processing power, limited computational capabilities, and limited energy and storage capacity [20]. GW-node is a powerful node with sufficiently large processing power, computational capabilities, and energy and storage capacity [20]. A user registers itself to the GW-node to become a valid user of the system. Whenever a user (medical professionals) wishes to obtain the physiological data of the patient, he transmits request message to the GW-node. Afterwards, the GW-node verifies the legitimacy of the user, if satisfied then it directs the desired MS-node(s) to answer to the user’s request.

Benefits of WMSN in providing healthcare services are as follows:

(i) Improvement in healthcare services,

(ii) Uninterrupted monitoring of patients,

(iii) Saving human labor, time, and money,

(iv) Protecting sensitive and private medical data of the patient from various adversaries.

3. Review of the Scheme Proposed by Kumar et al.

Initially, the GW-node chooses three secret keys denoted as J, K and Q, each of 256 bits. The GW-node also shares a secret key $K_{gs} = h(Q \parallel ID_g)$ with all deployed MS-nodes by means of some key agreement method [21, 22]. The scheme has five phases each of which are described in succession. But before giving detail of each phase of the scheme, we summarize the notations and description used throughout the paper in Notations.

3.1. User Registration Phase. The user (professional) $U$ registers itself to the GW-node in registration center of the hospital, in the following manner:

1. user submits his chosen identity $ID_u$ and password $PW_u$ to GW-node using a secure channel;

2. on receiving $(ID_u, PW_u)$, the GW-node computes $C_{ug} = E_I(ID_u \parallel ID_g)$ and $N_u = h(ID_u \oplus PW_u \oplus K)$;

3. GW-node stores $(h(\cdot), C_{ug}, N_u, K)$ into SC and issues $SC = (h(\cdot), C_{ug}, N_u, K)$ to $U$, where $K$ is a long-term secretkey of the GW-node.
3.2. Patient Registration Phase. A patient has to register itself in the registration center of the hospital [23]. Patient submits her/his name to the registration center. On receiving patient’s name, the registration center chooses a suitable medical sensor kit (i.e., MS-nodes and GW-node) according to the disease of the patient and assigns medical professionals (users). Then the registration center transmits the identity ID_u of the patient along with medical sensors kit information to the assigned professionals/users. Finally, a technician deploys MS-node on the body of the patient.

3.3. Login Phase. A professional logs in to the GW-node in order to gain patients’ medical data via WMSN. The user inserts her/his SC into the smart card reader and inputs ID_u and PW_u. Then the SC performs the following:

1. It calculates \( N_u^* = h(ID_u \oplus PW_u \oplus K) \) and compares \( N_u^* \) with the stored \( N_u \). If \( N_u^* = N_u \), the smart card continues further and else ends this session.

2. It generates a random nonce \( M \) to compute \( CID_u = E_K(h(ID_u) \| M) \| S_n \| C_{ug} \| T_u) \).

3. It sends login request \( \{CID_u, T_u\} \) to the GW-node with \( T_u \) as the current timestamp.

3.4. Authentication Phase. On receiving the login request \( \{CID_u, T_u\} \) from the professional, the GW-node verifies the authenticity of \( U \) and computes a message to transmit to the desired MS-node in the following manner:

1. It acquires current timestamp \( T_g \) and, for \( (T_g - T_u) > \Delta T \), discards the login request; otherwise it proceeds further.

2. It decrypts \( CID_u \) as \( D_K(CID_u) \) to obtain \( \{h(ID_u)^\delta, M, S_n, C_{ug} \| T_u^\delta\} \). Also, it decrypts \( C_{ug} \) as \( D_J(C_{ug}) \) to obtain \( \{ID_u^*, ID_g^*\} \).

3. It then computes \( h(ID_u^*) \) and verifies the equivalences \( h(ID_u^*) = h(ID_u)^\delta \), \( ID_g^* = ID_g \), and \( T_u = T_u^\delta \), if all the three equivalences hold, then it believes the login request to come from \( U \); otherwise it terminates the login session.

4. It acquires \( T_g \), another current timestamp and computes \( A_u = E_{SK_g}(ID_u \| S_u \| C_{ug} \| T_g \| T_u) \). Then, the GW-node sends \( \{A_u, T_g\} \) to the MS-node.

When the MS-node receives \( \{A_u, T_g\} \) from the GW-node, it performs the following steps:

5. It acquires current timestamp \( T_g^\delta \) and, for \( (T_g^\delta - T_g) > \Delta T \), discards the received message; otherwise it continues further.

6. The MS-node \( S_n \) performs the decryption \( D_{SK_g}(A_u) \) and obtains \( \{ID_u^*, S_u^*, M^*, T_g^*, T_u\} \) to make sure that the request has come from the legal GW-node.

7. It compares \( S_u^* \) with \( S_u \) and \( T_g^* \) with \( T_g \), and if any of these fail to match, then it discards the message; otherwise it continues further.

8. It computes session key \( K_{sess_u:S_n} = h(ID_u \| S_n \| M^* \| T_u) \). Then it acquires \( T_s \), another current timestamp, and computes \( L = E_{K_{sess_u:S_n}}(S_n \| M^* \| T_s) \). The MS-node sends \{\( L, T_s\)\} to the user.

When \( U \) receives \{\( L, T_s\)\} from the MS-node, its SC performs the following steps:

9. It acquires current timestamp \( T_u' \) and, for \( (T_u' - T_s) > \Delta T \), discards the received message. Or else, it proceeds further.

10. It computes session key \( K_{sess_U:S_n} = h(ID_u \| S_n \| M \| T_u) \). Then it performs the decryption \( D_{K_{sess_u:S_n}}(L) \) and obtains \( S_n \) and \( M^* \). It compares \( S_n \) with \( S_u \) and \( M^* \) with \( M \); if both the equivalences hold only, then the session key is assumed to be established securely.

3.5. Password Change Phase. \( U \) can change her/his password through the following stepwise procedure:

1. \( U \) inserts her/his SC into the terminal and inputs ID_u and PW_u.

2. Then SC computes \( N_u^* = h(ID_u \oplus PW_u \oplus K) \) and compares \( N_u^* \) with \( N_u \). If both the values match allows the user to enter a new password, otherwise discards the process.

3. \( U \) enters new password \( PW_{\text{new}} \).

4. SC computes \( (N_u)_{\text{new}} = h(ID_u \| PW_{\text{new}}) \) and then replaces \( N_u \) with \( (N_u)_{\text{new}} \).

4. Review of the Analysis of Kumar et al.’s Scheme

This section presents a review of the security problems of Kumar et al.’s scheme identified by Khan et al. [19]. This analysis is based on the assumption that an adversary \( U_a \) can recover [24, 25] the information stored in smart card.

If \( U_a \) extracts values \( h(\cdot), C_{ug}, N_u, K \) from the lost SC of a user, then he holds the master key \( K \) which is stored in the SC of each user (professional). Consequently, the scheme becomes vulnerable to different attacks described as follows.

4.1. User Impersonation Attack. Having master key \( K \) in hand, \( U_a \) can impersonate any user of the system to obtain patient’s physiological information. To impersonate \( U \), the attacker \( U_a \) intercepts the login request \( \{CID_u, T_u\} \) of \( U \) and decrypts \( CID_u = D_K(CID_u) = h(ID_u) \| M \| S_n \| C_{ug} \| T_u \) to obtain \( h(ID_u), M, S_n, C_{ug} \). Now \( U_a \) acquires a current timestamp \( T_u \) and a random nonce \( M_a \). Then computes \( CID_a = E_K(h(ID_u) \| S_n \| C_{ug} \| T_u) \) and sends \( \{CID_a, T_u\} \) to GW-node. Clearly this message will successfully go through GW-node authentication test as it contains valid values \( h(ID_u), S_n, C_{ug} \) and fresh values \( \{M_a, T_u\} \).

4.2. Lacks User Anonymity. \( U_a \) can obtain the hashed value of the identity of any user by decrypting the first component
of the login request. For instance, if \( U_a \) intercepts the login request \( \{C \bar{I}D_u, T_u\} \) of \( U \), then he can obtain \( h(ID_u) \) by decrypting \( C \bar{I}D_u \) using \( K \). Having hashed value of user’s identity \( h(ID_u) \) in hand, \( U_a \) can guess the corresponding identity \( ID_u \) of \( U \). Thus, the scheme fails to provide user anonymity.

4.3. Password Guessing Attack. We further extend the above two threats to a more harmful vulnerability. If \( U_a \) successfully guesses the identity \( ID_u \) of the user from whose smart card he extracts the secret key \( K \), then he can guess the password \( PW_u \) of \( U \). For this, \( U_a \) guesses \( PW_u \) as the probable password, computes \( N_u = h(ID_u \oplus PW_u \oplus K) \), and verifies if \( N_u = N_u \). If so, it implies success of \( U_a \) in guessing the \( PW_u \) of \( U \). In fact, it is complete violation of security since \( U_a \) holds user’s SC along with user’s identity \( ID_u \) and password \( PW_u \) so he can behave as the legal user \( U \).

4.4. Illegal Logged-In Users Using Legal Identity. \( U_a \) can guess the identity \( ID_u \) of any user as described in Section 4.2; he can misuse \( ID_u \) for crafting other damage to the security of the scheme as described below.

1. \( U_a \) applies for her/his registration by submitting \( ID_u \) and \( PW_u \), where \( PW_u \) is a random password chosen by \( U_a \).

2. In return, the GW-node provides \( U_a \) a \( SC_a = \{h(\cdot), C_{ust}, N_u, K\} \) with \( N_u = h(ID_u \oplus PW_u \oplus K) \) and \( C_{ust} = E_f(ID_u \| ID_g) \).

The role of password in the login-authentication procedure of the scheme is up to confirming the legitimacy of the user by her/his smart card. From then on, only user’s identity \( ID_u \) is used to authenticate \( U \) at the GW-node. As a result, there are two pictures.

(i) \( U_a \) can successfully log in as the legal user \( U \) with the received \( SC_a = \{h(\cdot), C_{ust}, N_u, K\} \). \( U_a \) inserts her/his SC into the terminal and inputs \( ID_u \) and \( PW_u \). Once \( ID_u \) and \( PW_u \) are verified, \( SC_a \) computes \( C \bar{I}D_u = E_K(h(ID_u) \| M \| S_n \| C_{ust} \| T_u) \) and sends the login request \( \{C \bar{I}D_u, T_u\} \) to the GW-node. Clearly, the GW-node considers it as a valid login request from the legitimate user \( U \) since it is computed using valid \( ID_u \) in \( C_{ust} \).

(ii) \( U_a \) has open option to distribute the user’s identity \( ID_u \) among malicious persons interested to obtain patient’s private health data in an illicit way. These persons can register themselves in similar manner as just explained in the previous scenario and can access data through MS-node. \( U_a \) can also distribute the values \( \{h(ID_u), S_n, C_{ust}\} \) in place of \( ID_u \) among these persons. Then it is possible to impersonate \( U \) as described in Section 4.1. In case such an illegal access is detected by the system, it will raise a question on the credibility of the valid user (medical professional) whose identity \( ID_u \) is misused by \( U_a \).

4.5. Insecure Session-Key. \( U_a \) can compute the session key to be used between a user and a MS-node during a particular session. Suppose \( U_a \) recovers the values \( \{h(ID_u), S_n, M\} \) out of \( C \bar{I}D_u \) of the intercepted login request of \( U \). Then he attempts to guess the identity \( ID_u \) as described in Section 4.2 and uses timestamp \( T_u \) from the corresponding intercepted login request \( \{C \bar{I}D_u, T_u\} \). Then \( U_a \) can easily compute the session key \( K_{sess} \) as described in Section 4.4.1. Then he can compute the session key \( K_{sess} = h(ID_u \| S_n \| M \| T_u) \) to be used by \( U \) and the MS-node with identity \( S_n \). Hence, the shared session key \( K_{sess} \) is insecure and \( U_a \) can decrypt the confidential messages communicated between MS-node and \( U \).

4.6. MS-Node Impersonation Attack. An active attacker \( U_a \) having secret key \( K \) obtained from a lost or stolen SC can perform decryption of \( C \bar{I}D_u \)’s for as many users as he wants. As a result, he can obtain the hashed value like \( h(ID_u) \) of all the target users. Next, \( U_a \) can guess the identity \( ID_u \) for each \( h(ID_u) \) and tabulates the values \( \{h(ID_u), ID_u\} \). After that, \( U_a \) can impersonate the MS-node to deceit legitimate users as explained below.

1. As \( U_a \) finds a login request \( \{C \bar{I}D_u, T_u\} \) on the network, he intercepts and blocks it and quickly decrypts \( C \bar{I}D_u \) to see if \( h(ID_u) \) included in it is present in the table maintained or not. If not then it relays the login request to GW-node.

2. If \( h(ID_u) \) exists in the tabular record, then \( U_a \) keeps the login request blocked and uses \( ID_u \) from the record, values \( \{M, S_n\} \) from current decryption, and \( T_u \) from login request; \( U_a \) quickly computes \( K_{sess} = h(ID_u \| S_n \| M \| T_u) \).

3. It computes \( L = E_{K_{sess}}(S_n \| M \| T_u) \) and sends \( \{L, T_u\} \) to \( U \), where \( T_u \) is the current timestamp chosen by \( U_a \).

4. Obviously \( L \) will qualify the verification test at the user side as it consists of valid \( \{S_n, M\} \) and fresh timestamp \( T_u \).

It is noticeable that \( K_{sess} \), the common session key is computed by \( U \) and \( U_a \) but \( U \) believes it to be confidential between him and the MS-node. Moreover, \( U_a \) can misguide the user doctor by sending fake data about the patient. Consequently, the patient may receive false treatment, thus denying the goal of healthcare through WMSN.

4.7. Lacking of Mutual Authentication between (i) MS-Node and GW-Node, (ii) \( U \) and MS-Node. In Kumar et al’s scheme, after verifying the login request of \( U \), GW-node computes and sends an ensuring message \( \{A_u, T_u\} \) to the required MS-node. Undoubtedly, the equivalence \( S_n^* = S_n \) confirms the legality of GW-node to MS-node but reverse is not achieved. Thus, GW-node has no way to ensure itself of connecting with real MS-node. Hence, mutual authentication between MS-node and GW-node is not achieved in the scheme.

Besides, the authors claim that their scheme achieves mutual authentication between MS-node and user \( U \). Mutual authentication between \( U \) and MS-node is established using the session key \( K_{sess} = h(ID \| S_n \| M \| T_u) \).
But as shown in Sections 4.5 and 4.6, $U_u$ can compute $K_{sess_u}$ and impersonate $S_n$, respectively. Therefore, mutual authentication between $U$ and MS-node is not achieved in the scheme.

4.8. Insider Attack. For convenience people use the same password for more than one application. During registration phase of the scheme, user submits her/his password plaintext $PW_u$ to GW-node. So, the system administrator at the GW-node easily comes to know the password of each user and he can use it to impersonate $U$ at servers, where $U$ is registered with the same password. Although authors assume the hospital registration center as a trusted authority, we think that often the trustworthy breaches the trust. Therefore, plaintext password $PW_u$ should not be submitted to any second party.

5. The Proposed Scheme

The proposed scheme has the same number of phases as in Kumar et al’s scheme. Each of the phases is detailed below along with Tables 1, 2, and 3. The GW-node keeps only one master secret key $K$ (length 256 bits). Besides, the GW-node shares a secret key $K_{gw}$ with MS-nodes using some key agreement method [21, 22].

5.1. User Registration Phase. The user (professional) $U$ registers itself to the GW-node in registration center of the hospital, in the following manner.

1. User chooses her/his identity $ID_u$ and submits it to the GW-node using a secure channel.
2. On receiving $ID_u$ the GW-node computes $C_{ug} = E_K(ID_u \parallel ID_g)$, $K_u = h(K \parallel ID_u \parallel ID_g)$, and $K_g = h(ID_g \parallel K)$. (3) GW-node stores $\{h(\cdot), C_{ug}\}$ into a SC and provides $SC = \{h(\cdot), C_{ug}\}$ along with values $\{K_u, K_g\}$ to $U$ through the secure channel.
4. On obtaining $SC = \{h(\cdot), C_{ug}\}$ and $\{K_u, K_g\}$, the user $U$ chooses his password $PW_u$ and computes $N_u = h(ID_u \parallel PW_u \parallel K_u)$, $PK_u = K_u \oplus (ID_u \parallel PW_u)$, and $PK_g = K_g \oplus (PW_u \parallel ID_u)$. Finally, $U$ inserts $N_u, PK_u$, and $PK_g$ in SC, so that $SC = \{h(\cdot), C_{ug}, N_u, PK_u, PK_g\}$. (2) It generates a random nonce $M$ and computes $C_{u1} = C_{ug} \oplus h(K_g)$ and $CID_{u1} = E_{K_u}(h(ID_u) \parallel M \parallel S_n \parallel C_{ug} \parallel T_u)$. (3) SC sends $\{CID_{u1}, C_{u1}, T_u\}$ as login request to GW-node, where $T_u$ is the current timestamp.

5.2. Patient Registration Phase. This phase is identical to that in Kumar et al’s scheme so we avoid its explanation here.

5.3. Login Phase. A professional logs in the GW-node in order to gain patients’ medical data via WMSN. The user inserts her/his SC into the smart card reader and inputs $ID_u$ and $PW_u$. Then the SC performs the following.

1. It retrieves $K_{ug} = P_{K_u} \oplus (ID_u \parallel PW_u)$, $K_g = PK_u \oplus (PW_u \parallel ID_u)$ and computes $N_u^* = h(ID_u \parallel PW_u \parallel K_u)$. For $N_u^* = N_u$ it continues further; otherwise it stops the session.
2. It acquires current timestamp $T_g$ and, for $(T_g - T_u) > \Delta T$, discards the login request; otherwise it proceeds further.
3. It retrieves $C_{ug} = C_{u1} \oplus h(K_g)$ and decrypts $CID_{u1}$ as $D_K(CID_{u1})$ to obtain $\{h(ID_u), M, S_n, C_{ug} \parallel T_u\}$. It verifies the equivalence $C_{ug} = C_{ug}$, and if correct, then it decrypts $C_{ug}$ as $D_K(C_{ug})$ to obtain $ID_u^*$ and $ID_g^*$. (3) It then computes $h(ID_u^*)$ and verifies the equivalences $h(ID_u^*) = h(ID_g^*)$ and $ID_u^* = ID_g$ and $T_u = T_u^*$, if all the three equivalences hold then believes the login request to come from $U$; otherwise terminates the login session.
4. It acquires $T_g$ as current timestamp, computes $C_{g1} = K_g \oplus (M \parallel T_g \parallel T_u)$, and sends $\{C_{g1}, T_g\}$ to $U$. It acquires $T_g$ as another current timestamp and computes $C_{g2} = h(K_g) \oplus (CID_{g1} \parallel T_g \parallel M \parallel T_u)$ and $A_u = h(CID_u \parallel K_{gw} \parallel T_g \parallel S_n \parallel T_g)$. Then, the GW-node sends $\{C_{g2}, A_u, T_g\}$ to the MS-node.

On receiving $\{C_{g1}, T_g\}$ from the GW-node, $U$ verifies the legitimacy of GW-node as follows.

1. It checks if $(T_u - T_g) > \Delta T$; if so, it dumps the session; otherwise it continues further.
2. It obtains $(M^* \parallel T_g^* \parallel T_u^*) = C_{g1} \oplus K_g$ and verifies the equivalence $M^* = M, T_g^* = T_g$, and $T_u^* = T_u$; if each holds, then GW-node is authenticated; otherwise it terminates the login session. After this mutual authentication, $U$ and GW-node compute $K_{sess_u-GW} = h(M \parallel ID_u \parallel T_g)$, as the session key. On receiving $\{C_{g2}, A_u, T_g\}$ from the GW-node, the MS-node performs the following operations.

1. It checks if $(T_u - T_g) > \Delta T$; if so, terminates the session; otherwise it proceeds further.
2. It obtains $(CID_{u}^* \parallel T_g^* \parallel M^* \parallel T_u^*) = C_{g2} \oplus h(K_{gw})$, computes $A_u^* = h(CID_{u}^* \parallel K_{gw} \parallel T_g^* \parallel S_n \parallel T_g)$, and compares $A_u^*$ with $A_u$. The equivalence $A_u^* = A_u$ verifies the legitimacy of the GW-node and hence of $U$.
3. It acquires $T_{sg}$ as current timestamp, computes $C_{u2} = h(T_g \parallel K_{gw} \parallel T_{sg}) \oplus h(CID_u \parallel S_n \parallel T_g)$, and sends $\{C_{u2}, T_{sg}\}$ to the GW-node. Also it computes $C_{g2}^* = h(S_n \parallel M^* \parallel T_u^* \parallel T_s)$, where $T_s$ is another
Table 1: User registration phase of the Proposed scheme.

| User (U)                  | GW-node                                      |
|---------------------------|----------------------------------------------|
| User registration phase:  |                                              |
| Chooses $ID_u$            | $C_{ag} = E_k(ID_u \| ID_g)$                 |
|                           | $K_u = h(K \| ID_u \| ID_g)$                 |
|                           | $K_g = h(ID_u \| K)$                        |
| Chooses $PW_u$            |                                              |
| $N_u = h(ID_u \| PW_u \| K_u)$ |                                              |
| $PK_u = K_u \oplus (ID_u \| PW_u)$ |                                              |
| $PK_g = K_g \oplus (PW_u \| ID_u)$ |                                              |
| Inserts $N_u$, $PK_u$, and $PK_g$ in SC so that SC = $\{h(\cdot), C_{ag}, N_u^*, PK_u, PK_g\}$ | |

Table 2: Login and authentication phase of the Proposed scheme.

| User (U)                  | GW-node                                      | MS-node                                      |
|---------------------------|----------------------------------------------|----------------------------------------------|
| Login and authentication phase: |                                              |                                              |
| $U$: inserts $ID_u$ and $PW_u$ |                                              |                                              |
| SC: $K_u \leftarrow PK_u \oplus (ID_u \| PW_u)$, | $K_{g1} = K_g \oplus (M \| T_g \| T_u)$, | $A_u = h(CID_u \| K_{g1} \| T_g \| S_u \| T_g)$, |
| $K_g \leftarrow PK_g \oplus (PW_u \| ID_u)$, | $N_u^* = h(ID_u \| PW_u \| K_u)$,           |                                              | $C_{g2} = h(K_{g1} \| S_{\theta} \| T_g \| M \| T_u)$, |
| $N_u^* = N_u$             |                                              |                                              | $K_{g2} = h(ID_u \| ID_g \| T_u^*) = ID_g \& T_u^*$, |
| $C_{a1} = C_{ag} \oplus h(K_g)$, |                                              |                                              | For $(T_g^* - T_u^*) \leq \Delta T$ |
| $CID_u = E_k(h(ID_u) \| M \| S_u \| C_{ag} \| T_u)$, |                                              |                                              | $C_{ag} \leftarrow C_{a1} \oplus h(K_g)$, |
| $\{CID_u,GW,T_u\}$       |                                              |                                              | $\{h(ID_u)^3, M, S_u, C_{ag,3}, T_u^3\} \leftarrow D_k(CID_u)$, |
|                           | For $C_{ag,3} = C_{ag}$                     |                                              | For $C_{g1} = K_g \oplus (M \| T_g \| T_u)$, |
|                           | $\{ID_u^*, ID_g^*\} \leftarrow D_k(C_{ag})$ |                                              | $A_u\leftarrow A_u \oplus h(CID_u \| K_{g1} \| T_g \| S_u \| T_g)$, |
|                           | For $h(ID_u^*) = h(ID_u)^3, ID_g^* = ID_g \& T_u^*$, |                                              | For $(T_s^* - T_{g2}^*) \leq \Delta T$ |
|                           |                                              |                                              | $K_{sess_{U,GW}} = h(M \| ID_u \| T_g)$, |
|                           |                                              |                                              | $C_{g1} = h(T_g \| K_{g1} \| T_{g2} \| S_u \| T_{g2}) \oplus h(CID_u \| S_u)$, |
|                           |                                              |                                              | $C_{g1} = h(S_u \| M \| T_u^* \| T_g)$, |
|                           |                                              |                                              | $K_{sess_{GW,SM}} = h(K_{g1} \| T_{g2} \| M)$, |
|                           |                                              |                                              | $C_{g2} = C_{g2} \leftarrow C_{g2} \oplus h(T_g \| K_{g1} \| T_{g2} \| S_u \| T_{g2})$, |
|                           |                                              |                                              | $K_{sess_{SM,GW}} = h(M \| T_s \| S_u)$, |
current timestamp of MS-node. Then, the MS-node sends \( \{C_s^*, T_s\} \) to \( U \).

On receiving \( \{C_s^*, T_s\} \) from the MS-node, the GW-node performs the following operations.

1. It checks if \( (T'_u - T_s) > \Delta T \); if so, terminates the session; otherwise it proceeds further.
2. It obtains \( (h(CID_u \| S_u))^* = C_{s2} \oplus h(T'_u \| K_{g2} \| T_s) \) computes \( h(CID_u \| S_u) \), and compares it with \( (h(CID_u \| S_u))^* \). The equivalence \( h(CID_u \| S_u) = h(CID_u \| S_u) \) verifies the legitimacy of MS-node.

After this mutual authentication, GW-node and MS-node compute \( K_{ sessGW-SN} = h(K_{g2} \| T_s \| M) \), as the session key.

On receiving \( \{C_{s2}, T_s\} \) from the MS-node, \( U \) performs the following.

1. It checks if \( (T'_u - T_s) > \Delta T \); if so, it dumps the session, otherwise it proceeds further.
2. It computes \( C_{s2} = h(S_u \oplus M \| T_s \| T_u) \) and compares it with \( C_{s2} \). Then, for \( C_{s2} = C_{s2}^* \) the authenticity of MS-node is verified.

After this mutual authentication, \( U \) and MS-node compute \( K_{ sessU-SN} = h(M \| T_s \| S_u) \), as the session key.

5.5. Password Change Phase. \( U \) can change her/his password in the following manner. For this, \( U \) inserts her/his \( SC \) into the terminal, inputs her \( ID_u \) and \( PW_u \), and opts to change his password. Then the following steps are performed to update a new password.

1. SC retrieves \( K_u = PK_u \oplus (ID_u \| PW_u), K_g = PK_g \oplus (PW_u \| ID_u) \) and computes \( N_u^* = h(ID_u \| PW_u \| K_u) \). If \( N_u^* \neq N_u \), then it proceeds further after asking for a new password; otherwise it discards the password change request.
2. \( U \) enters new password \( (PW_u)_{new} \).
3. SC computes \( (N_u)_{new} = h(ID_u \| (PW_u)_{new} \| K_u), (PK_u)_{new} = K_u \oplus (ID_u \| (PK_u)_{new}), \) and \( (PK_g)_{new} = K_g \oplus ((PW_u)_{new} \| ID_u) \).
4. SC replaces \( N_u, PK_u, \) and \( PK_g \) with \( (N_u)_{new}, (PK_u)_{new}, \) and \( (PK_g)_{new} \), respectively.

6. Analysis of the Security of the Proposed Scheme

This section, examines the security of the proposed scheme. We will display that the proposed scheme is secure under the same assumption subject to which Kumar et al’s scheme is attackable. The assumption is that an attacker \( U_a \) can extract \([24, 25]\) the information stored inside smart card.

6.1. Resisting User Impersonation Attack. To impersonate as the user, \( U_a \) has to compute a valid login request. Suppose \( U_a \) obtains the lost smart card of \( U \) and extracts the values \( \{C_{sag}, N_u, PK_u, PK_g\} \) stored in it. Though \( C_{sag} \) is involved in both the components \( \{CID_u, C_s\} \) of the login request, without \( C_{sag} \), \( h(ID_u) \), and \( S_u \) computation of these components is incomplete. To recover \( K_u \) from \( PK_u \), the attacker \( U_a \) needs to know of user’s identity and password. On the contrary, to obtain \( ID_u \) from \( PK_u \) or \( PK_g \), the attacker \( U_a \) should hold \( K_u \) or \( K_g \), respectively. Further, it is not feasible to obtain \( ID_u \) or \( K_u \) from \( N_u \) due to noninvertible nature of hash function. Thus, the scheme resists user impersonation attack.

6.2. Providing User Anonymity. If \( U_a \) intercepts the login request \( \{CID_u, C_{sag}, T_u\} \) of \( U \), then he needs \( K_u \) to obtain \( h(ID_u) \) by decrypting \( CID_u \). But \( U_a \) neither knows \( K_u \) nor can recover it by extracting information \( \{h(\cdot), C_{sag}, N_u, PK_u, PK_g\} \) from the lost smart card of some user; say \( U \). To take out \( K_u \) from \( PK_u \), the attacker \( U_a \) should know user’s identity and password. In fact, key \( K_u \) required to encrypt/decrypt \( CID_u \) is not stored directly in user’s smart card and is different for each user. Therefore, \( U_a \) cannot obtain \( h(ID_u) \) and guess the identity as in Kumar et al’s scheme. On the other hand, to procure identity \( ID_u \) from \( N_u, PK_u, \) or \( PK_g \) is infeasible. It requires knowledge of keys \( K_u \) and \( K_g \) to gain \( ID_u \) out of \( PK_u \) or \( PK_g \), respectively. Moreover, one-way property of hash function does not allow extraction of \( ID_u \) out of \( N_u \). Therefore, \( U_a \) cannot gain the identity of a user and hence the scheme provides user anonymity.

6.3. Resisting Password Guessing Attack. In order to guess \( U \)’s password \( PW_u \), from \( N_u = h(ID_u \| PW_u \| K_u) \), obtained from the lost SC of \( U \), the attacker \( U_a \) requires knowledge of
ID\textsubscript{u} and K\textsubscript{u}. As described in Section 6.2, U\textsubscript{a} cannot gain the identity of a user either from the lost smart card of a user or from an intercepted login request. Besides, K\textsubscript{u} is not available as plaintext in U\textsubscript{a}'s SC and is not obtainable from PK\textsubscript{u} without having exact values of ID\textsubscript{u} and PW\textsubscript{u}. Thus, the scheme resists password guessing attack.

6.4. Resisting Illegal Logged-In Users Using Legal Identity. Since it is not possible to guess or know the identity ID\textsubscript{u} of a logging user, U\textsubscript{a} cannot register itself to the GW-node with legal identity ID\textsubscript{u} and fake password PW\textsubscript{u}. Thus, U\textsubscript{a} cannot harm the security of the scheme by misusing the identity. As a result, the scenario of many illegal users logged in with legal identity ID\textsubscript{u} of a registered user is not possible in the proposed scheme.

6.5. Providing Secure Session Key between Every Pair of the Participating Entities. The proposed scheme establishes session key between every pair of participating entities. Session key between U and GW-node is K\textsubscript{sess\_U\_GW} = h(M \parallel ID\textsubscript{u} \parallel T\textsubscript{g}) which depends on three values M, ID\textsubscript{u}, and T\textsubscript{g}. Although user's identity ID\textsubscript{u} is fixed, M and T\textsubscript{g} are different for each session imparting dynamic nature to K\textsubscript{sess\_U\_GW}. However T\textsubscript{g} is available in [C\textsubscript{g1}, T\textsubscript{g}] from the open network but U\textsubscript{a} cannot compute K\textsubscript{sess\_U\_GW} without having M and ID\textsubscript{u}. Session key between GW-node and MS-node is K\textsubscript{sess\_GW\_MS} = h(K\textsubscript{g2} \parallel T\textsubscript{g} \parallel M) which is dynamic because of fresh timestamp T\textsubscript{g} and one time usable random number M. Although K\textsubscript{g2} is fixed but is known only to the GW-node and the MS-node so no one except these two entities can compute the valid K\textsubscript{sess\_GW\_MS}. Moreover, U\textsubscript{a} cannot access M from CID\textsubscript{u} without knowing K\textsubscript{u}, from C\textsubscript{g1} without knowing K\textsubscript{g2}; and from C\textsubscript{g2} due to noninvertible nature of hash function. Session key between U and MS-node is K\textsubscript{sess\_U\_MS} = h(M \parallel T\textsubscript{g} \parallel S\textsubscript{u}) which an attacker cannot compute without knowing M. Thus, the scheme establishes independent and secure session keys between every pair of the participating entities.

6.6. Resisting Sensor-Node Impersonation Attack. In order to impersonate the MS-node, U\textsubscript{a} should be able to compute the response messages sent by it to the GW-node and U. To compute C\textsubscript{u} = h(T\textsubscript{g} \parallel K\textsubscript{g2} \parallel T\textsubscript{sg}) \oplus h(CID\textsubscript{u} \parallel S\textsubscript{u}) and C\textsubscript{g2} = h(S\textsubscript{u} \parallel M^* \parallel T\textsubscript{g} \parallel T\textsubscript{u}* \parallel T\textsubscript{g}) the knowledge of K\textsubscript{g2} and M* is required, respectively. Since K\textsubscript{g2} = h(K \parallel ID\textsubscript{g2}) is shared secretly by GW-node with MS-node using some key agreement method [21, 22] and its computation involves master secret key K and identity ID\textsubscript{g2} of the GW-node, U\textsubscript{a} cannot access or compute K\textsubscript{g2}. Further M/ M* is not retrievable from CID\textsubscript{u}, C\textsubscript{g1}, and C\textsubscript{g2} without knowing K\textsubscript{u} and K\textsubscript{g2}, respectively. Moreover, one-way property of hash function prohibits extraction of M/ M* from C\textsubscript{g2}. Hence U\textsubscript{a} cannot impersonate the MS-node to make fool of the user and GW-node.

6.7. Providing Mutual Authentication between Every Pair of the Participating Entities. At each of the three ends, any received message undergoes at least two-step verification test to verify the authenticity of the sender. For every message, firstly timestamp is checked for freshness followed by one or more equivalences holding tests. The proposed scheme achieves mutual authentication between U and GW-node by exchange of messages [CID\textsubscript{u}, C\textsubscript{g1}, T\textsubscript{g}] and [C\textsubscript{g1}, T\textsubscript{g}]. When GW-node receives [CID\textsubscript{u}, C\textsubscript{g1}, T\textsubscript{g}] from U, in addition to timestamp freshness test, the equivalences h(ID\textsubscript{u}*) = h(ID\textsubscript{u})\textsuperscript{5}, ID\textsubscript{g} = ID\textsubscript{g}, and T\textsubscript{u} = T\textsubscript{g} are required to guarantee the legitimacy of U. Similarly, for [C\textsubscript{g1}, T\textsubscript{g}] received by U from GW-node, the equivalence M* = M, T\textsubscript{u}* = T\textsubscript{g}, and T\textsubscript{u} should hold to prove the validity of the GW-node.

Mutual authentication between the GW-node and the MS-node is achieved through the messages [C\textsubscript{g2}, A\textsubscript{u}, T\textsubscript{g}]{,} and [C\textsubscript{g1}, T\textsubscript{g}]. Corresponding to the message [C\textsubscript{g2}, A\textsubscript{u}, T\textsubscript{g}]{,} the equivalence A\textsubscript{u} = A\textsubscript{u} is imperative to confirm the legitimacy of GW-node and hence of U to MS-node. On the other hand, only the designated MS-node can compute C\textsubscript{u} = h(T\textsubscript{g} \parallel K\textsubscript{g2} \parallel T\textsubscript{sg}) \oplus h(CID\textsubscript{u} \parallel S\textsubscript{u}) and the authorized GW-node can retrieve correct h(CID\textsubscript{u} \parallel S\textsubscript{u}) from C\textsubscript{u} as the computation and retrieval involves use of K\textsubscript{g2} hence mutually authenticate the entities to each other.

As just discussed, U is authenticated to the MS-node via message [C\textsubscript{g2}, A\textsubscript{u}, T\textsubscript{g}]{,} with which GW-node is verified. Finally, the legitimacy of MS-node is ensured to U by means of the equivalence C\textsubscript{g2} = C\textsubscript{g2}*. In this way, our scheme provides perfect mutual authentication.

6.8. Resisting Insider Attack. During registration phase, U submits only his identity ID\textsubscript{u} to the GW-node at the hospital registration center. The GW-node provides secret keys K\textsubscript{u} and K\textsubscript{g} to the user. Then using his chosen password PW\textsubscript{u} and identity ID\textsubscript{u}, the user U itself computes N\textsubscript{u} = h(ID\textsubscript{u} \parallel PW\textsubscript{u} \parallel K\textsubscript{g}) and embeds K\textsubscript{u} and K\textsubscript{g} as PK\textsubscript{u} = K\textsubscript{u} \oplus (ID\textsubscript{u} \parallel PW\textsubscript{u}) and PK\textsubscript{g} = K\textsubscript{g} \oplus (PW\textsubscript{u} \parallel ID\textsubscript{g}). Finally, U inserts N\textsubscript{u}, PK\textsubscript{u}, and PK\textsubscript{g} in SC. Since the insider of the system never receives user’s password, privileged insider attack is not applicable on the scheme.

7. Performance Analysis of the Proposed Scheme via Comparison

Now, we compare our scheme with Kumar et al.’s scheme [18] to present a comparative analysis of its performance and efficiency. Table 4 is about memory space required by smart card and computational complexity/cost in both the schemes. Table 5 exhibits the performance of both the schemes. For convenience, we assume that the identity ID\textsubscript{u}, password PW\textsubscript{u}, random numbers \{M\}, timestamps \{T\textsubscript{u}, T\textsubscript{g}\}, etc., and outputs of one-way hash function \{h(ID\textsubscript{u} \parallel PW\textsubscript{u} \parallel K\textsubscript{g})\}, etc. are 128-bit long.

Table 4 shows that the memory space required by the smart card in Kumar et al.’s scheme and the proposed scheme is 512 bits and 640 bits, respectively. Further, it is noticeable that our scheme adds some hash functions (h(·)) but remarkably cuts the number of time consuming symmetric cryptography operation (S\textsubscript{g}) at each of the three ends. The most important aspect is that there is no symmetric
Table 4: Comparison of efficiency: memory space and computational cost/complexity.

| Schemes                      | Kumar et al.’s [18] | Ours  |
|------------------------------|----------------------|-------|
| Memory space needed by SC    | 512 bits             | 640 bits |
| Computational complexity     |                      |       |
| Registration phase (U)       | Nil                  | $1h()$ |
| Registration phase (GW-node) | $1h() + 1S_y$        | $2h() + 1S_y$ |
| Login and authentication phase (SC) | $4h() + 2S_y$ | $6h() + 1S_y$ |
| Login and authentication phase (GW-node) | $1h() + 3S_y$ | $7h() + 1S_y$ |
| Login-authentication phase (MS-node) | $1h() + 2S_y$ | $7h()$ |

Table 5: Comparison of performance.

| Security characteristics                              | Kumar et al.’s [18] | Ours  |
|-------------------------------------------------------|----------------------|-------|
| Resists user impersonation attack                      | No                   | Yes   |
| Resists password guessing attack                       | No                   | Yes   |
| Resists multi-logged-in users attack                   | No                   | Yes   |
| Resists sensor-node impersonation attack               | No                   | Yes   |
| Resists insider attack                                 | No                   | Yes   |
| Provides user anonymity                               | No                   | Yes   |
| Provides verification mechanism in SC                  | Yes                  | Yes   |
| Provides freely password change facility               | Yes                  | Yes   |
| Provides U and GW-node mutual authentication           | No                   | Yes   |
| Provides GW-node and MS-node mutual authentication     | No                   | Yes   |
| Provides U and MS-node mutual authentication           | No                   | Yes   |
| Establishes secure session key between U and GW-node   | No                   | Yes   |
| Establishes secure session key between GW-node and MS-node | No   | Yes |
| Establishes secure session key between U and MS-node   | No                   | Yes   |

8. Conclusion

A secure and efficient user authentication scheme is essential to offer reliable and proficient healthcare services via WMSNs. This work is motivated by the security problems of Kumar et al.’s scheme for healthcare services using WMSNs. In this paper, we have designed a user authentication scheme to eradicate the security problems of Kumar et al.’s scheme. Our scheme is user anonymous and is free from risks occurring due to loss of smart card of a user. It defies insider attack and password guessing attack. The most important feature of the scheme is that it establishes mutual authentication and provides session key between every pair of the participating entities, that is, user, GW-node, and MS-node.

Notations

$U$: User (professional)

...
Conflict of Interests

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

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-288.

References

[1] Y.-Y. Wang, J.-Y. Liu, F.-X. Xiao, and J. Dan, “A more efficient and secure dynamic ID-based remote user authentication scheme,” Computer Communications, vol. 32, no. 4, pp. 583–585, 2009.

[2] M. K. Khan, S.-K. Kim, and K. Alghathbar, “Cryptanalysis and security enhancement of a ‘more efficient & secure dynamic ID-based remote user authentication scheme,” Computer Communications, vol. 34, no. 3, pp. 305–309, 2011.

[3] M. Kumar, M. K. Gupta, and S. Kumari, “An improved efficient remote password authentication scheme with smart card over insecure network,” International Journal of Network Security, vol. 13, no. 3, pp. 167–177, 2011.

[4] S. Kumari, M. K. Gupta, and M. Kumar, “Cryptanalysis and security enhancement of Chen et al’s remote user authentication scheme using smart card,” Central European Journal of Computer Science, vol. 2, no. 1, pp. 60–75, 2012.

[5] M. K. Khan, S. Kumari, and M. K. Gupta, “More efficient key-hash based fingerprint remote authentication scheme using mobile device,” Computing, 2013.

[6] S. Kumari and M. K. Khan, “Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme,” International Journal of Communication Systems, 2013.

[7] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Shengwei, “A dynamic user authentication scheme for wireless sensor networks,” in Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, vol. 1, pp. 318–327, Taichung, Taiwan, June 2006.

[8] H.-R. Tseng, R.-H. Jan, and W. Yang, “An improved dynamic user authentication scheme for wireless sensor networks,” in Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM ’07), pp. 986–990, Washington, DC, USA, November 2007.

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

[10] B. Vaidya, J. J. P. C. Rodrigues, and J. H. Park, “User authentication schemes with pseudonymity for ubiquitous sensor network in NGN,” International Journal of Communication Systems, vol. 23, no. 9-10, pp. 1201–1222, 2010.

[11] M. K. Khan and K. Alghathbar, “Security analysis of ‘two-factor user authentication in wireless sensor networks,” in Proceedings of the 4th International Conference on Information Security and Assurance (ISA ’10), vol. 6059 of Lecture Notes in Computer Science, pp. 55–60, Miyazaki, Japan, June 2010.

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

[13] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, “An enhanced two-factor user authentication scheme in wireless sensor networks,” Ad-Hoc & Sensor Wireless Networks, vol. 10, no. 4, pp. 1–11, 2010.

[14] S. Saleem, S. Ullah, and H. S. Yoo, “On the security issues in wireless body area networks,” Journal of Digital Content Technology and Its Applications, vol. 3, no. 3, pp. 178–184, 2009.

[15] S. Ullah and K. S. Kwak, “Body area network for ubiquitous healthcare applications: theory and implementation,” Journal of Medical Systems, vol. 35, no. 5, pp. 1243–1244, 2011.

[16] S. Ullah, H. Higgins, B. Braem et al., “A comprehensive survey of wireless body area networks—on PHY, MAC, and network layers solutions,” Journal of Medical Systems, vol. 36, no. 3, pp. 1065–1094, 2012.

[17] W.-Y. Chung, “Multi-modal sensing M2M healthcare service in WSN,” KSII Transactions on Internet and Information Systems, vol. 6, no. 4, pp. 1090–1105, 2012.

[18] P. Kumar, S.-G. Lee, and H.-J. Lee, “E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks,” Sensors, vol. 12, no. 2, pp. 1625–1647, 2012.

[19] M. K. Khan, S. Kumari, and P. Singh, “Cryptanalysis of an efficient-strong authentication protocol (e-sap) for healthcare applications using wireless medical sensor networks,” KSII Transactions on Internet and Information Systems, vol. 7, no. 5, pp. 967–979, 2013.

[20] E. H. Callaway, Wireless Sensor Networks, Architectures and Protocols, Taylor & Francis, Boca Raton, Fla, USA, 2003.

[21] C. Chen, D. He, S. Chan, J. Bu, Y. Gao, and R. Fan, “Lightweight and provably secure user authentication with anonymity for the global mobility network,” International Journal of Communication Systems, vol. 24, no. 3, pp. 347–362, 2011.

[22] Z. L. Ping and W. Yi, “An ID-based key agreement protocol for wireless sensor networks,” in Proceedings of the 1st International Conference on Information Science and Engineering (ICISE ’09), pp. 2542–2545, Nanjing, China, December 2009.

[23] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, “SAGE: a strong privacy-preserving scheme against global eavesdropping for ehealth systems,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, pp. 365–378, 2009.

[24] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO ’99), pp. 388–397, Santa Barbara, Calif, USA, 1999.

[25] 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.