Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 1905872, 14 pages
http://dx.doi.org/10.1155/2016/1905872

Research Article

Anticollusion Attack Noninteractive Security Hierarchical Key Agreement Scheme in WHMS

Kefei Mao, Jianwei Liu, and Jie Chen

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Kefei Mao; owen.buaa@gmail.com

Received 16 February 2016; Accepted 4 May 2016

Academic Editor: Jit S. Mandeep

Wireless Health Monitoring Systems (WHMS) have potential to change the way of health care and bring numbers of benefits to patients, physicians, hospitals, and society. However, there are crucial barriers not only to transmit the biometric information but also to protect the privacy and security of the patients' information. The key agreement between two entities is an essential cryptography operation to clear the barriers. In particular, the noninteractive hierarchical key agreement scheme becomes an attractive direction in WHMS because each sensor node or gateway has limited resources and power. Recently, a noninteractive hierarchical key agreement scheme has been proposed by Kim for WHMS. However, we show that Kim's cryptographic scheme is vulnerable to the collusion attack if the physicians can be corrupted. Obviously, it is a more practical security condition. Therefore, we proposed an improved key agreement scheme against the attack. Security proof, security analysis, and experimental results demonstrate that our proposed scheme gains enhanced security and more efficiency than Kim's previous scheme while inheriting its qualities of one-round communication and security properties.

1. Introduction

Wireless Health Monitoring System (WHMS) is a dedicated network environment that supports the biometric information acquisition devices to gather people's health data anytime and anywhere [1]. Moreover, WHMS is a typical example of using wireless technologies to reduce medical expense and improve social benefits, such as detecting the lonely stroke patients timely [2, 3]. Security and privacy are the major concerns in medical activities, and WHMS is not an exception [4–6]. To provide privacy and security assurances in WHMS, it is important to provide security services by using cryptographic algorithms. Thus, obtaining cryptographic keys is an essential operation to achieve the security goals in WHMS. There are several key agreement schemes that have been proposed for WHMS applications [5, 7–10].

The noninteractive scheme is becoming a very active direction in the sensors networks [11–14] because sensor nodes have limited energy and processing and storage abilities. A noninteractive hierarchical key agreement scheme, called the Freshness-Preserving Noninteractive Hierarchical Key Agreement Protocol (FNKAP), was proposed by Kim [8] in 2014. The major advantages of the proposed scheme in Kim [8] go as follows. Firstly, there is only one-round communication to agree on a session key between two entities. Secondly, it is declared that the FNKAP achieves the patient anonymity and the session key confidentiality, and it can resist active and passive security attacks. However, we found that there is a flaw in the FNKAP when the physicians are not to be trusted. The scheme is not strong enough against the collusion attack where there are two adversaries who are a physician and a patient, separately. More precisely, in order to obtain a specific patient's electronic medical data, the adversary can pretend to be sick and become the same physician's patient with the victim in the real world. Then, the adversary bribes any other physician to get the private values of a physician. Finally, the adversary could calculate the session key and decrypts the victim's electronic health data freely. Note that a physician can casually expose the private values because the disclosed values are untraceable in Kim's scheme. As a result, this method of attack is reasonable and straightforward to implement.

The contributions of this paper are twofold. First, we illustrate that there is a weakness in FNKAP and introduce specific attack methods. Second, we propose an enhanced security hierarchical key agreement scheme with noninteracting for WHMS based on pairings. Security proof and analysis illustrate our scheme enhances security strength of FNKAP, and it can resist the collusion attack. Moreover, theoretical analysis results show that our scheme is more efficient than Kim's work.

The rest of this paper is organized as follows. We formalize a basic system structure for WHMS in Section 2, and we also give the security model and define the adversary's ability in the same section. We simply highlight Kim's scheme [8] in Section 3. The weakness of Kim's scheme is discussed in Section 4. We detail our enhanced security hierarchical key agreement scheme against the security attacks in Section 5. We present the analysis of our improvements regarding correctness and security in Section 6. We compare our scheme with Kim's scheme in terms of functionality and performance in Section 7. Finally, this paper is concluded in Section 8.

2. Preliminaries

We first illustrate the basic system structure of WHMS in this section. Moreover, we introduce security threats, security model, bilinear group, and mathematic assumption, separately. Basic notations are provided in Notations section.

2.1. Basic Structure. As depicted in Figure 1, a typical hierarchical key agreement for WHMS involves five types of parties. They are, namely, the u-Health Server (SV), the physicians (PH), the patients (PA), the gateways (GW), and the sensor nodes (SN). There is a hierarchical permission structure from the u-Health Server SV to the physicians PH_i, $i \in (1, 2, \ldots)$, to the gateway $\mathrm{GW}_{i,j}$, $i, j \in (1, 2, \ldots)$ of patients $\mathrm{PA}_{i,j}$ and to the sensor node $\mathrm{SN}_{i,j,d}$, $i, j, d \in (1, 2, \ldots)$ of the gateway $\mathrm{GW}_{i,j}$ [7, 8].

As the root authority, SV is responsible for managing the entities' authorities in WHMS. SV produces the private keys of entities such as GW, SN, and PH. When PA wants to use WHMS, his/her GW and SN should agree on the session keys with the physician (PH), separately. Similarly, when PH wants to send a diagnostic report to PA, he/she should also agree on a session key with PA.

2.2. Security Threats. Kim assumes that the physicians are trusted in paper [8]. However, we point out that the scheme should take the risk of the physician's corruption because it is more practical. In practice, not all physicians are trusted all the time. For example, as reported, the staff of a famous hospital sold the patient's personal medical data in USA [7], and 500 patients' medical information may have been compromised at a medical center in LA because an employee's laptop was stolen [15].

Thus, we assume a security model in which the adversary has the following abilities. First, the adversary can totally control the channel. Therefore, the adversary can eavesdrop, intercept, modify, replay, or inject any data via the channel. Second, the adversary can compromise the secure information from the physicians except for the victim's current physician. Third, the adversary can also compromise several sensor nodes and gateways except for the victim's current GW and SN.

We aim to achieve the following security goals under the above security threats.

Key Agreement. Two entities establish a session key which is only known by specific entities.

Anonymity and Untraceability. The identities of GW and SN should be kept confidential from the adversary and cannot be traced by the adversary.

Resistance Passive and Active Attacks. The scheme is secure against the passive and active attacks.

2.3. Security Model. Inspired by the security model for a noninteractive hierarchical key agreement scheme [11] and the original Bellare-Rogaway key exchange model [16], the security model of our scheme is stated as follows.

Participants. We model the scheme participants as a finite set U of fixed size with each A being a Probabilistic Polynomial Time (PPT) turing machine. Each scheme participant $A \in U$ may execute a polynomial number of protocol instances in parallel. We will refer to sth instance of principal A communicating with peer B as $\prod_{A,B}^{s}$.

Adversary Model. The adversary $\mathscr{A}$ is modeled as a PPT Turing machine and can be given all public parameters of the system, and he/she can access the oracle by issuing some specified queries:

(i) Send($\prod_{A,B}^{s}, D$). The adversary $\mathscr{A}$ sends the message D to the session s executed by A communicating with B. Since our proposal is a noninteractive scheme, the query does not need to be responded to.

(ii) Establish(A). The adversary $\mathscr{A}$ names a node A and obtains all the secret values held by the node. Neither of the patient's gateway and sensor nodes named in the test query or any of their ancestors can be established.

(iii) Reveal($\prod_{A,B}^{s}$). If the query is achieved, the system returns the session key to the adversary $\mathscr{A}$. The session between the target patient's facilities (a gateway and sensor nodes) and the physician cannot be revealed.

(iv) Test($\prod_{A,B}^{s}$). Only one query of this form is allowed for the adversary $\mathscr{A}$. The adversary $\mathscr{A}$ names ID_A and ID_B and executes this query at any time. Then, a number sk is returned as follows. A bit b is chosen at random in $\{0, 1\}$. If $b = 1$ then the adversary gets the secret key shared between the two nodes, and if $b = 0$ it gets

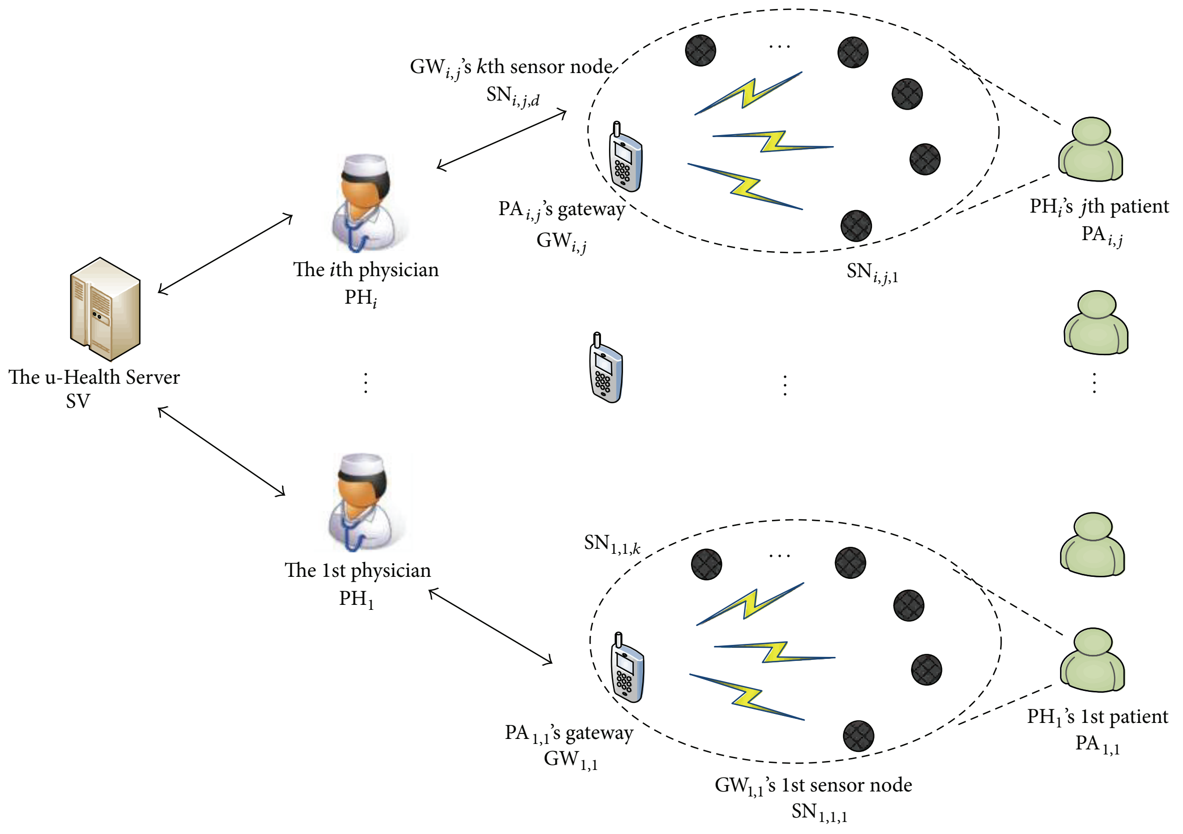


Figure 1: Basic hierarchical key agreement structure in WHMS.

a key chosen at random from the set of all possible shared keys.

Definition 1 (HKA-security). As a function of the security parameter k, we define the advantage $\mathrm{Adv}^{\mathrm{HKA}}_{\mathscr{A},\sum}$ of the PPT adversary $\mathscr{A}$ in an attacking scheme $\sum$ as

$$\mathrm{Adv}^{\mathrm{HKA}}_{\mathscr{A},\sum} = \left|2\mathrm{Succ}^{\mathrm{HKA}}_{\mathscr{A},\sum} - 1\right|. \quad (1)$$

Here, $\mathrm{Succ}^{\mathrm{HKA}}_{\mathscr{A},\sum}$ is the probability that the adversary queries Test($\prod^{s}_{A,B}$) and outputs a bit b^* such that b is used by the test query. We call a hierarchical key agreement scheme $\sum$ to be HKA secure if for any PPT adversary the $\mathscr{A}$ function is negligible.

2.4. Bilinear Groups

Definition 2 (bilinear map). G_1 is an additive cyclic group of prime order q and G_2 is a multiplicative cyclic group of prime order p. The bilinear pairing is a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ with the following properties [17].

Bilinearity. For all $P, Q \in G_1$ and $a, b \in Z_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

Nondegeneracy. The map does not send all pairs in $G_1 \times G_1$ to the identity in G_2.

Computability. There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$.

2.5. Mathematic Assumption.

The mathematic assumptions used in the paper are listed as follows.

Definition 3 (Elliptic Curve Discrete Logarithm Problem, ECDL problem). Suppose E is an elliptic curve over a finite field F_q. Given $Q, P \in E$ to find the $n \in Z_q^*$, $nP = Q$ is believed to be hard [18].

Definition 4 (Bilinear Diffie-Hellman Problem, BDH problem). BDH problem is defined as follows. There is a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Given $(P, aP, bP, cP \in G_1)$ for $a, b, c \in Z_q^*$, to compute the $\hat{e}(P, P)^{abc} \in G_2$ is believed to be hard [17].

Definition 5 (Decisional Bilinear Diffie-Hellman Problem, DBDH problem). DBDH problem is defined as follows. There is a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Given $(P, aP, bP, cP \in G_1)$ for $a, b, c, r \in Z_q^*$, to differentiate the $\hat{e}(P, P)^{abc} \in G_2$ and $\hat{e}(P, P)^{r} \in G_2$ is believed to be hard [17].

2.6. Notations.

To provide a quick reference, the basic notations used in the paper are listed in Notations section.

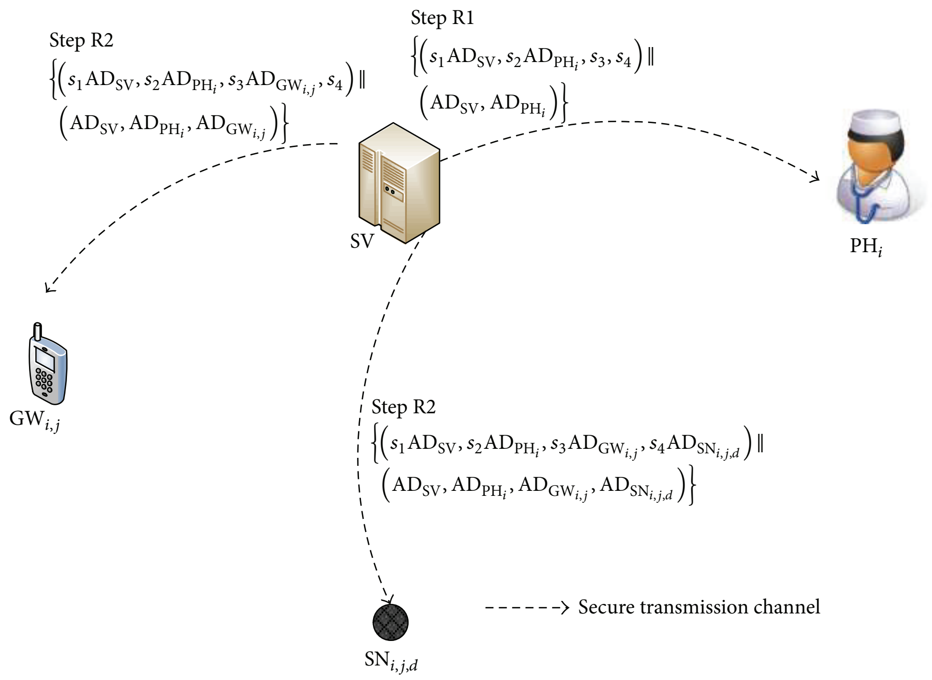

Step R1
$\{(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3, s_4) \parallel (AD_{SV}, AD_{PH_i})\}$
Step R2
$\{(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,j}}, s_4) \parallel (AD_{SV}, AD_{PH_i}, AD_{GW_{i,j}})\}$
SV
PH_i
$GW_{i,j}$
Step R2
$\{(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,j}}, s_4 AD_{SN_{i,j,d}}) \parallel (AD_{SV}, AD_{PH_i}, AD_{GW_{i,j}}, AD_{SN_{i,j,d}})\}$
$SN_{i,j,d}$
Secure transmission channel

FIGURE 2: Physician and patient registration phase in Kim's paper [8].

3. Review of the Kim's Scheme

In this section, we briefly review Kim's key agreement scheme [8], which consists of three phases: System Initialization Phase, Physician and Patient Registration Phase, and Noninteractive Key Agreement and Secure Communication Phase.

3.1. System Initialization Phase. SV generates two groups G_1 and G_2 of prime order p with a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Also, it chooses a cryptographic hash function $H : \{0,1\}^* \rightarrow G_1$. After that, SV picks four random numbers $s_1, s_2, s_3, s_4 \leftarrow Z_q^*$ as the master private keys. Then, SV computes an amplified identity $AD_{SV} = H(ID_{SV})$ and a public key $s_1 AD_{SV}$. Finally, SV keeps the master private keys and the amplified identity, securely.

3.2. Physician and Patient Registration Phase. Before providing service, the patient PA and his/her physician PH must register in SV. Here, the statement $A \Rightarrow B : M$ denotes that B receives a message M from A via a secure channel. The Physician and Patient Registration Phase is basically shown in Figure 2.

Step R1 ($SV \Rightarrow PH_i : Reg_{PH_i}$). When a physician PH_i wants to be a legal e-medical physician, he/she sends his/her identity ID_{PH_i} to SV via a secure channel. Then, SV validates the identity ID_{PH_i}. If the solution is positive, SV sends $Reg_{PH_i} = \{(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3, s_4) \parallel (AD_{SV}, AD_{PH_i})\}$. Here, $AD_{PH_i} = H(ID_{PH_i})$. Finally, PH_i stores the received information, securely.

Step R2 ($SV \Rightarrow GW_{i,j} : Reg_{GW_{i,j}}$ and $SV \Rightarrow SN_{i,j,d} : Reg_{SN_{i,j,d}}$). When a patient $PA_{i,j}$ of PH_i wants to use the service in the WHMS, he/she should register his/her gateway $GW_{i,j}$ and k sensor nodes $SN_{i,j,d}$, $1 \leq d \leq k$ in SV. SV validates the identity ID_{PH_i} submitted by $PA_{i,j}$. If the solution is positive, SV receives the gateway's identity $ID_{GW_{i,j}}$ and the sensor nodes' identity $ID_{SN_{i,j,d}}$, $1 \leq d \leq k$. Then, SV sends $Reg_{GW_{i,j}}$ and $Reg_{SN_{i,j,d}}$ to them via a secure channel. Here, $Reg_{GW_{i,j}}$ and $Reg_{SN_{i,j,d}}$ as follows:

$$\begin{aligned}
Reg_{GW_{i,j}} &= \left\{\left(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,j}}, s_4\right) \parallel \right.\\
&\left.\left(AD_{SV}, AD_{PH_i}, AD_{GW_{i,j}}\right)\right\},\\
Reg_{SN_{i,j,d}} & \\
&= \left\{\left(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,j}}, s_4 AD_{SN_{i,j,d}}\right) \parallel \right.\\
&\left.\left(AD_{SV}, AD_{PH_i}, AD_{GW_{i,j}}, AD_{SN_{i,j,d}}\right)\right\}.
\end{aligned} \tag{2}$$

Finally, $GW_{i,j}$ and $SN_{i,j,d}$ store their received information, securely.

3.3. Noninteractive Key Agreement and Secure Communication. In this phase, the sensor node $SN_{i,j,d}$ and the gateway $GW_{i,j}$ of the patient PA_i and the physician PH_i agree on a fresh session key for establishing a secure communication channel. Here, the statement $A \rightarrow B : M$ denotes that B receives a message M from A via a unsecure channel. The Noninteractive Key Agreement and Secure Communication is basically shown in Figure 3.

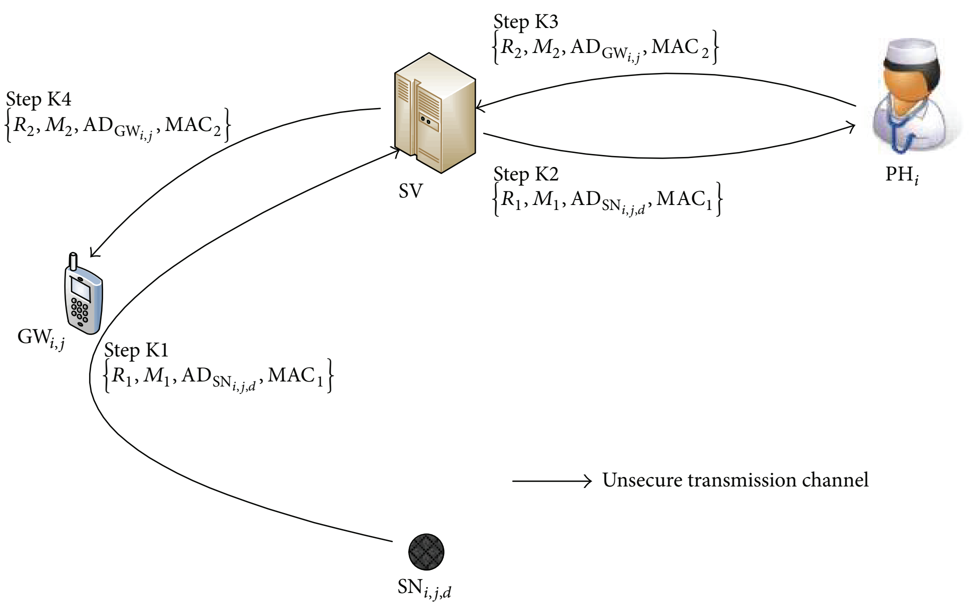

Step K3
$\{R_2, M_2, \mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{MAC}_2\}$
Step K4
$\{R_2, M_2, \mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{MAC}_2\}$
SV
PH$_i$
Step K2
$\{R_1, M_1, \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{MAC}_1\}$
GW$_{i,j}$
Step K1
$\{R_1, M_1, \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{MAC}_1\}$
Unsecure transmission channel
SN$_{i,j,d}$

Figure 3: Noninteractive key agreement and secure communication in Kim's paper [8].

Step K1 ($\mathrm{SN}_{i,j,d} \rightarrow \mathrm{SV} : \{R_1, M_1, \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{MAC}_1\}$). $\mathrm{SN}_{i,j,d}$ chooses a random number r_1 and computes $R_1 = r_1\mathrm{AD}_{\mathrm{SN}_{i,j,d}}$. The fresh session key sk_1 is computed as follows:

$$\begin{aligned}\mathrm{sk}_1 &= \widehat{e}\left(s_1\mathrm{AD}_{\mathrm{SV}}, \mathrm{AD}_{\mathrm{SV}}\right) \cdot \widehat{e}\left(s_2\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \widehat{e}\left(s_3\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \widehat{e}\left(s_4\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{r_1}.\end{aligned} \tag{3}$$

Then, $\mathrm{SN}_{i,j,d}$ computes $M_1 = E_{\mathrm{sk}_1}(\mathrm{Data}_i)$ and $\mathrm{MAC}_1 = H(\mathrm{sk}_1 \parallel R_1 \parallel M_1)$, where Data_i is the data collected by $\mathrm{SN}_{i,j,d}$.

Step K2. When PH_i is authenticated by SV, he/she can check the data of the patient PA_i. PH_i computes the fresh session key sk_1' as follows:

$$\begin{aligned}\mathrm{sk}_1' &= \widehat{e}\left(s_1\mathrm{AD}_{\mathrm{SV}}, \mathrm{AD}_{\mathrm{SV}}\right) \cdot \widehat{e}\left(s_2\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \widehat{e}\left(\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{GW}_{i,j}}\right)^{s_3} \cdot \widehat{e}\left(\mathrm{AD}_{\mathrm{PH}_i}, R_1\right)^{s_4}.\end{aligned} \tag{4}$$

Then, PH_i computes $\mathrm{MAC}_1' = H(\mathrm{sk}_1' \parallel R_1 \parallel M_1)$. Only if MAC_1' is equal to MAC_1 does PH_i assure the correctness of sk_1'. Then, PH_i decrypts M_1 to get Data_i by using the key sk_1'.

Step K3 ($\mathrm{PH}_i \rightarrow \mathrm{GW}_{i,j} : \{R_2, M_2, \mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{MAC}_2\}$). When PH_i wants to send the electronic health report to PA_i, he/she chooses a random number r_2 and computes $R_2 = r_2\mathrm{AD}_{\mathrm{PH}_i}$. PH_i computes the fresh session key sk_2 as follows:

$$\begin{aligned}\mathrm{sk}_2 &= \widehat{e}\left(s_1\mathrm{AD}_{\mathrm{SV}}, \mathrm{AD}_{\mathrm{SV}}\right) \cdot \widehat{e}\left(s_2\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \widehat{e}\left(\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{GW}_{i,j}}\right)^{s_3} \\ &\quad \cdot \widehat{e}\left(\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{GW}_{i,j}}\right)^{s_4 r_2}.\end{aligned} \tag{5}$$

In addition, PH_i computes $M_2 = E_{\mathrm{sk}_2}(\mathrm{Data}_i)$ and $\mathrm{MAC}_2 = H(\mathrm{sk}_2 \parallel R_2 \parallel M_2)$. Here, Data_i is the electronic health report composed by PH_i.

Step K4. When $\mathrm{GW}_{i,j}$ is authenticated by SV, he/she can receive the report of the patient PA_i from SV. $\mathrm{GW}_{i,j}$ computes the fresh session key sk_2' as follows:

$$\begin{aligned}\mathrm{sk}_2' &= \widehat{e}\left(s_1\mathrm{AD}_{\mathrm{SV}}, \mathrm{AD}_{\mathrm{SV}}\right) \cdot \widehat{e}\left(s_2\mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \widehat{e}\left(s_3\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right) \cdot \widehat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_4}.\end{aligned} \tag{6}$$

Then, $\mathrm{GW}_{i,j}$ computes $\mathrm{MAC}_2' = H(\mathrm{sk}_2' \parallel R_2 \parallel M_2)$. Only if MAC_2' is equal to MAC_2 does $\mathrm{GW}_{i,j}$ assure the correctness of sk_2'. Then, $\mathrm{GW}_{i,j}$ decrypts M_2 to get Data_i by using the key sk_2'.

4. Security Analysis of Kim's Scheme

The author of [8] proposed a noninteractive key agreement scheme for freshness-preserving in WHMS. Under our security model, there is a weakness in the scheme as explained in the following section.

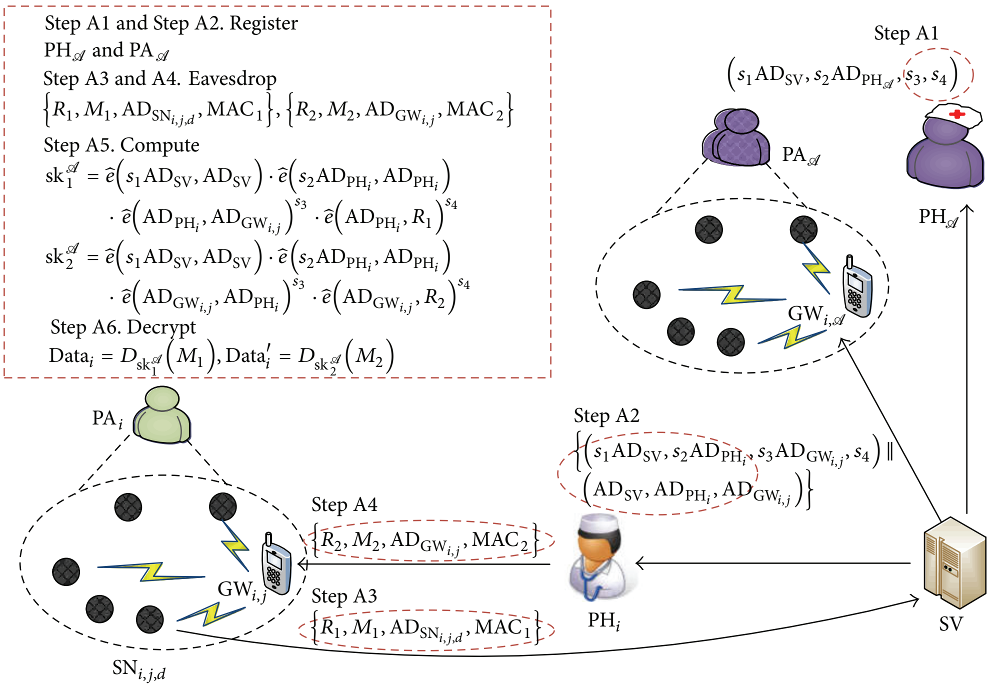

Step A1 and Step A2. Register
$PH_{\mathscr{A}}$ and $PA_{\mathscr{A}}$
Step A3 and A4. Eavesdrop
$\{R_1, M_1, AD_{SN_{i,j,d}}, MAC_1\}, \{R_2, M_2, AD_{GW_{i,j}}, MAC_2\}$
Step A5. Compute
$sk_1^{\mathscr{A}} = \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \cdot \hat{e}(AD_{PH_i}, AD_{GW_{i,j}})^{s_3} \cdot \hat{e}(AD_{PH_i}, R_1)^{s_4}$
$sk_2^{\mathscr{A}} = \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \cdot \hat{e}(AD_{GW_{i,j}}, AD_{PH_i})^{s_3} \cdot \hat{e}(AD_{GW_{i,j}}, R_2)^{s_4}$
Step A6. Decrypt
$Data_i = D_{sk_1^{\mathscr{A}}}(M_1), Data_i' = D_{sk_2^{\mathscr{A}}}(M_2)$
Step A1
$(s_1 AD_{SV}, s_2 AD_{PH_{\mathscr{A}}}, s_3, s_4)$
$PA_{\mathscr{A}}$
$PH_{\mathscr{A}}$
$GW_{i,\mathscr{A}}$
PA_i
Step A2
$\{(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,j}}, s_4) \parallel (AD_{SV}, AD_{PH_i}, AD_{GW_{i,j}})\}$
Step A4
$\{R_2, M_2, AD_{GW_{i,j}}, MAC_2\}$
$GW_{i,j}$
Step A3
$\{R_1, M_1, AD_{SN_{i,j,d}}, MAC_1\}$
PH_i
SV
$SN_{i,j,d}$

FIGURE 4: Security against collusion attack.

4.1. Security against Collusion Attack. We now demonstrate that Kim's scheme is vulnerable to the collusion attack as claimed. One adversary $\mathscr{A}_1$ has registered as a legal physician $PH_{\mathscr{A}}$, and the other adversary $\mathscr{A}_2$ has registered as a normal patient $PA_{\mathscr{A}}$, as shown in Figure 4. The adversaries can obtain the electronic health data of any patient who is diagnosed by the same physician PH_i with the adversary $\mathscr{A}_2$. The adversaries attack a patient $PA_{i,j}$ as follows.

Step A1. Assume that $\mathscr{A}_1$ is an attacker who has registered as a physician $PH_{\mathscr{A}}$ in SV, and then he/she can legally receive a private key set $(s_1 AD_{SV}, s_2 AD_{PH_{\mathscr{A}}}, s_3, s_4)$ from SV (Step R1). Then, $\mathscr{A}_1$ sends a part of private key set $(s_1 AD_{SV}, s_2 AD_{PH_{\mathscr{A}}}, s_3, s_4)$ to $\mathscr{A}_2$.

Step A2. $\mathscr{A}_2$ is an adversary who has registered as a patient of the physician PH_i. He/she can legally receive a secure data set $(AD_{SV}, AD_{PH_i}, AD_{GW_{i,\mathscr{A}}})$ and a private key set $(s_1 AD_{SV}, s_2 AD_{PH_i}, s_3 AD_{GW_{i,\mathscr{A}}}, s_4)$ of his/her gateway $GW_{i,\mathscr{A}}$ from SV (Step R2).

Step A3. Suppose $SN_{i,j,d}$ is a victim PA's smart node that sends information through the gateway $GW_{i,j}$. PA is diagnosed by the same physician PH_i with $\mathscr{A}_2$. When $SN_{i,j,d}$ runs the Step K1, an adversary can intercept the data $\{R_1, M_1, AD_{SN_{i,j,d}}, MAC_1\}$ because the communications are unsecure between $SN_{i,j,d}$ and SV.

Step A4. When PH_i sends the electronic health report to PA_i at the Step K3, an adversary can intercept data $\{R_2, M_2, AD_{GW_{i,j}}, MAC_2\}$ because the communications are also unsecure between PH_i and SV.

Step A5. $\mathscr{A}_2$ can compute the session key after the above steps. $\mathscr{A}_2$ receives (s_3, s_4) from $\mathscr{A}_1$ at Step A1. Then, he/she gets $(s_1 AD_{SV}, s_2 AD_{PH_i})$ and (AD_{SV}, AD_{PH_i}) at Step A2. Moreover, the information $\{R_1, M_1, AD_{SN_{i,j,d}}, MAC_1\}$ and $\{R_2, M_2, AD_{GW_{i,j}}, MAC_2\}$ is intercepted at Steps A3 and A4, separately. Therefore, $\mathscr{A}_2$ can compute the same session keys sk_1 and sk_2 as follows:

$$\begin{aligned}
sk_1^{\mathscr{A}} &= \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \\
&\quad \cdot \hat{e}(AD_{GW_{i,j}}, AD_{PH_i})^{s_3} \cdot \hat{e}(AD_{PH_i}, R_1)^{s_4} \\
&= \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \\
&\quad \cdot \hat{e}(s_3 AD_{GW_{i,j}}, AD_{PH_i}) \cdot \hat{e}(s_4 AD_{SN_{i,j,d}}, AD_{PH_i})^{r_1} \\
&= sk_1, \\
sk_2^{\mathscr{A}} &= \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \\
&\quad \cdot \hat{e}(AD_{PH_i}, AD_{GW_{i,j}})^{s_3} \cdot \hat{e}(AD_{GW_{i,j}}, R_2)^{s_4} \\
&= \hat{e}(s_1 AD_{SV}, AD_{SV}) \cdot \hat{e}(s_2 AD_{PH_i}, AD_{PH_i}) \\
&\quad \cdot \hat{e}(AD_{PH_i}, AD_{GW_{i,j}})^{s_3} \cdot \hat{e}(AD_{GW_{i,j}}, AD_{PH_i})^{s_4 r_2} \\
&= sk_2.
\end{aligned} \tag{7}$$

Step A6. $\mathscr{A}_2$ decrypts M_1 and M_2 to obtain the victim's medical information using the session keys $\text{sk}_1^{\mathscr{A}}$ and $\text{sk}_2^{\mathscr{A}}$, respectively.

5. Our Proposed Scheme

In this section, we propose an improved scheme that can overcome the flaw of Kim's scheme in Section 4. Our scheme construction is inspired by the practical noninteractive key distribution scheme in [12] and Kim's paper [8]. Our scheme consists of four operational phases: Setup Phase, Key Generation Phase, Key Agreement from SN to PH Phase, and Key Agreement from PH to GW Phase. The details of our scheme are described as follows.

5.1. Setup Phase. In this phase, the u-Health Server SV, as the Private Key Generator (PKG), takes as inputs a security parameter k and the maximal number of the physicians N. Then, SV outputs the system public parameters params and the master private key sets sk. SV publishes params and keeps sk private.

Similar to the identity-based cryptography scheme, SV generates two groups G_1 and G_2 of prime order p with a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$. However, it chooses three cryptographic hash functions $H_1 : \{0,1\}^* \to G_1$, $H_2 : G_2 \times G_1^3 \to \{0,1\}^k$, and $H_3 : \{0,1\}^* \times G_1^3 \to \{0,1\}^k$. After that, SV generates $3+N$ random numbers $\{s_1, s_2, s_3, sr_i \leftarrow Z_q^* \mid i \in (1,2,\ldots,N)\}$ and selects a random generator $P_0 \in G_1$. Finally, SV keeps the master key $\text{sk} = \{s_1, s_2, s_3, sr_i \mid i \in (1,2,\ldots,N)\}$ secret and publishes params $= (q, G_1, G_2, P_0, \hat{e}, H_1, H_2, H_3, s_1P_0, s_2P_0, s_3P_0, s_1sr_iP_0, s_2sr_iP_0, s_3sr_iP_0 \mid i \in (1,2,\ldots,N))$. Here, P_0 is used to verify the correctness of the secret key sets.

It is important to note that although our proposal increase the storage space because of the values sr_i, $i \in [1,N]$, there is a one-to-one mapping between a physician PH_i and a value sr_i. In addition, the list of physicians must be stored in SV. Thus, we can use the mapping to reduce the storage space. For instance, SV gets the list of the registration physicians. Then, SV chooses a secret hash function $H^* : \{0,1\}^q \to Z_q^*$ and a random value sr_0. Finally, SV can compute the i times hash function $H^*(\cdot)$ to get the secret value $sr_i = H^*(\cdots(H^*(sr_0)))$. In this way, SV only needs to store the selected hash function and initial value sr_0, secretly. On one hand, the proposal can save the storage resources by using the hash function. On the other hand, it increases the consumption of the computing resources. In order to balance the computing cost and the storage space, SV can store not only the initial value sr_0, but also some intermediate random values sr_j. We introduce the scheme by using the secret values sr_i, $i \in [1,N]$ to help the analysis.

5.2. Key Generation Phase. In this phase, SV takes the identity ID_{PH_i} as an input and outputs a secret key set sk_{PH_i}. Moreover, PH_i takes his/her secret key set sk_{PH_i} and the identities $\text{ID}_{\text{GW}_{i,j}}$ and $\text{ID}_{\text{SN}_{i,j,d}}$ as inputs and outputs two secret key sets $\text{sk}_{\text{GW}_{i,j}}$ and $\text{sk}_{\text{SN}_{i,j,d}}$, separately.

Step 1. A physician PH_i, $i \in (1,2,\ldots,N)$ submits his/her identity ID_{PH_i} to SV for registration. If the identity of PH_i is validated, Then, SV computes $\text{AD}_{\text{SV}_i} = H_1(\text{ID}_{\text{SV}} \parallel sr_i)$, $\text{AD}_{\text{PH}_i} = H_1(\text{ID}_{\text{PH}_i} \parallel sr_i)$, and a private key set of PH_i, $i \in (1,2,\ldots,N)$ as follows:

$$\text{sk}_{\text{PH}_i} = \left(s_1 sr_i \text{AD}_{\text{PH}_i}, s_2 sr_i, s_3 sr_i\right). \tag{8}$$

In addition, SV packs a data package containing a private key set and two amplified identities $\{\text{sk}_{\text{PH}_i} \parallel (\text{AD}_{\text{SV}_i}, \text{AD}_{\text{PH}_i})\}$ and delivers the data package to PH_i via a secure channel. Here, the secure channel could be a smart card passed by a trusted person. Finally, PH_i keeps the received information, securely.

Step 2. When a patient $\text{PA}_{i,j}$ goes to see a doctor in a real clinic, they decide to use the WHMS to monitor his/her health directed by a physician PH_i. The patient submits his/her identity $\text{ID}_{\text{PA}_{i,j}}$ and the identity of gateway $\text{ID}_{\text{GW}_{i,j}}$ and sensor nodes $\text{ID}_{\text{SN}_{i,j,d}}$ to PH_i for registration. If the patients' identity is validated, PH_i generates a random value $a_{i,j} \in Z_q^*$ and computes the amplified identities $\text{AD}_{\text{GW}_{i,j}} = H_1(\text{ID}_{\text{GW}_{i,j}} \parallel a_{i,j})$ and $\text{AD}_{\text{SN}_{i,j,d}} = H_1(\text{ID}_{\text{SN}_{i,j,d}} \parallel a_{i,j})$. PH_i computes the private key sets of $\text{GW}_{i,j}$ and $\text{SN}_{i,j,d}$ as follows:

$$\begin{aligned} \text{sk}_{\text{GW}_{i,j}} &= \left(s_1 sr_i \text{AD}_{\text{PH}_i}, s_2 sr_i \text{AD}_{\text{GW}_{i,j}}\right), \\ \text{sk}_{\text{SN}_{i,j,d}} &= \left(s_1 sr_i \text{AD}_{\text{PH}_i}, s_3 sr_i \text{AD}_{\text{SN}_{i,j,d}}\right). \end{aligned} \tag{9}$$

Next, PH_i packs a data package containing a private key set and three amplified identities $\{\text{sk}_{\text{GW}_{i,j}} \parallel (\text{AD}_{\text{SV}_i}, \text{AD}_{\text{PH}_i}, \text{AD}_{\text{GW}_{i,j}})\}$ and delivers the data package to $\text{GW}_{i,j}$ via a secure channel. Furthermore, SV packs a data package containing a private key set $\{\text{sk}_{\text{SN}_{i,j,d}} \parallel (\text{AD}_{\text{SV}_i}, \text{AD}_{\text{PH}_i}, \text{AD}_{\text{GW}_{i,j}}, \text{AD}_{\text{SN}_{i,j,d}})\}$ and delivers it to $\text{SN}_{i,j,d}$ via a secure channel. Finally, $\text{GW}_{i,j}$ and $\text{SN}_{i,j,d}$ store their received information in a secure area, respectively.

5.3. Key Agreement from SN to PH Phase. In this phase, a sensor node $\text{SN}_{i,j,d}$ of the patient $\text{PA}_{i,j}$ makes a connection with the physician PH_i. The sensor node $\text{SN}_{i,j,d}$ and the physician PH_i achieve a key agreement.

Step 1. When a sensor node $\text{SN}_{i,j,d}$ wants to upload the patient's medical data, $\text{SN}_{i,j,d}$ chooses a random number r_1 and computes $R_1 = r_1 \text{AD}_{\text{SN}_{i,j,d}}$ using its amplified identity $\text{AD}_{\text{SN}_{i,j,d}}$. The session key sk_1 is calculated as follows:

$$\begin{aligned} K_1 &= \hat{e}\left(\text{AD}_{\text{SN}_{i,j,d}}, s_1 sr_i \text{AD}_{\text{PH}_i}\right) \\ &\quad \cdot \hat{e}\left(s_3 sr_i \text{AD}_{\text{SN}_{i,j,d}}, \text{AD}_{\text{PH}_i}\right)^{r_1}, \\ \text{sk}_1 &= H_2\left(K_1 \parallel R_1 \parallel \text{AD}_{\text{SN}_{i,j,d}} \parallel \text{AD}_{\text{PH}_i}\right). \end{aligned} \tag{10}$$

Then, $\text{SN}_{i,j,d}$ computes $M_1 = E_{\text{sk}_1}(\text{Data}_{i,j,d})$ and $V_1 = H_3(\text{sk}_1 \parallel M_1 \parallel T_1 \parallel R_1 \parallel \text{AD}_{\text{SN}_{i,j,d}} \parallel \text{AD}_{\text{PH}_i})$. Here, $\text{Data}_{i,j,d}$ is the data collected by $\text{SN}_{i,j,d}$ and T_1 is a current timestamp. Finally, $\mathrm{SN}_{i,j,d}$ sends a message package $D_1 = \{R_1, \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}, T_1, M_1, V_1\}$ to SV.

Step 2. After receiving the data package D_1, SV verifies the timestamp T_1 whether it is within the valid time for communication. If it is invalid, the key agreement terminates. Otherwise, it can assure the package by judging $V_1^* = V_1$ as follows:

$$\begin{aligned} K_1^* &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i}, \\ \mathrm{sk}_1^* &= H_2\left(K_1^* \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i}\right), \\ V_1^* &= H_3\left(\mathrm{sk}_1^* \parallel M_1 \parallel T_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i}\right). \end{aligned} \tag{11}$$

Only if V_1^* is equal to V_1 included in D_1 does SV assure the source of package from a sensor node $\mathrm{AD}_{\mathrm{SN}_{i,j,d}}$ and send a notice to PH_i. Finally, SV store the package D_1 in its database.

Step 3. When PH_i is authenticated by SV, he/she can check the data of a sensor node $\mathrm{SN}_{i,j,d}$. PH_i computes the fresh session key sk_1' as follows:

$$\begin{aligned} K_1' &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i}, \\ \mathrm{sk}_1' &= H_2\left(K_1' \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i}\right). \end{aligned} \tag{12}$$

In addition, PH_i computes $V_1' = H_3(\mathrm{sk}_1' \parallel M_1 \parallel T_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ by using the information of D_1. Only if V_1' is equal to V_1 does PH_i assure the correctness of sk_1' and decrypt M_1 to get $\mathrm{Data}_{i,j,d}$ by using the key sk_1'.

5.4. Key Agreement from PH to GW Phase. In this phase, the physician PH_i makes a connection with a patient's gateway $\mathrm{GW}_{i,j}$, and they agree on a fresh session key for communication.

Step 1. When PH_i wants to communicate with $\mathrm{PA}_{i,j}$ such as sending the electronic health report, he/she chooses a random number r_2 and computes $R_2 = r_2 \mathrm{AD}_{\mathrm{PH}_i}$. PH_i computes the fresh session key sk_2 as follows:

$$\begin{aligned} K_2 &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_2 sr_i r_2}, \\ \mathrm{sk}_2 &= H_2\left(K_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}}\right). \end{aligned} \tag{13}$$

In addition, PH_i computes $M_2 = E_{\mathrm{sk}_2}(\mathrm{Data}_{i,j})$ and $V_2 = H_3(\mathrm{sk}_2 \parallel M_2 \parallel T_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}})$. Here, $\mathrm{Data}_{i,j}$ is the electronic health report composed by PH_i, and T_2 is a current timestamp. Finally, PH_i sends a message package $D_2 = \{R_2, \mathrm{AD}_{\mathrm{PH}_i}, \mathrm{AD}_{\mathrm{GW}_{i,j}}, T_2, M_2, V_2\}$ to SV.

Step 2. After receiving the data package D_2, SV checks the validity of the timestamp T_2. If it has grown stale, SV quits the session. Otherwise, SV can assure the package by judging $V_2^* = V_2$ as follows:

$$\begin{aligned} K_2^* &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_2 sr_i}, \\ \mathrm{sk}_2^* &= H_2\left(K_2^* \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}}\right), \\ V_2^* &= H_3\left(\mathrm{sk}_2^* \parallel M_2 \parallel T_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}}\right). \end{aligned} \tag{14}$$

Only if V_2^* is equal to V_2 included in D_2 does SV assure the source of package from a physician $\mathrm{AD}_{\mathrm{PH}_i}$ and send a notice to $\mathrm{PA}_{i,j}$. Finally, SV stores the package D_2 in its database.

Step 3. When $\mathrm{GW}_{i,j}$ is authenticated by SV, he/she can get the report of the patient $\mathrm{PA}_{i,j}$ from SV. $\mathrm{GW}_{i,j}$ computes the fresh session key sk_2' as follows:

$$\begin{aligned} K_2' &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \cdot \hat{e}\left(s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right), \\ \mathrm{sk}_2' &= H_2\left(K_2' \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}}\right). \end{aligned} \tag{15}$$

Then, $\mathrm{GW}_{i,j}$ computes $V_2' = H_3(\mathrm{sk}_2' \parallel M_2 \parallel T_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}})$. Only if V_2' is equal to V_2 does $\mathrm{GW}_{i,j}$ assure the correctness of sk_2' and decrypt M_2 to get $\mathrm{Data}_{i,j}$ by using the key sk_2'.

6. Correctness and Security

In this section, we present the correctness of our improved scheme. Then, we illustrate that our enhanced key agreement scheme can overcome the two security weaknesses of security analysis of FNKAP by security analysis.

6.1. Correctness. We verify the correctness of key agreement in our scheme as follows:

$$\begin{aligned} K_1 &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \hat{e}\left(s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{r_1} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \\ &\quad \cdot \hat{e}\left(r_1 \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i} \\ &= K_1^* \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i} \\ &= K_1'. \end{aligned} \tag{16}$$

Thus, the agreed session keys sk_1, sk_1^*, and sk_1' computed by PH_i, SV, and $\mathrm{SN}_{i,j,d}$ are equal. The same as above, we prove that sk_2 is equal to sk_2' because K_2 is equal to K_2^* and K_2':

$$\begin{aligned} K_2 &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \\ &\quad \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_2 sr_i r_2} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \\ &\quad \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, r_2 \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_2 sr_i} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_2 sr_i} \\ &= K_2^* \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}\right) \cdot \hat{e}\left(s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right) \\ &= K_2'. \end{aligned} \tag{17}$$

6.2. Security Proof. In the following, we will show that our scheme is provably secure under DBDH assumption in the random oracle model. We treat H_1, H_2, and H_3 as three random oracles.

Theorem 6. *Let G_1 and G_2 be two groups of order q and $\hat{e}$ be a bilinear mapping that together satisfy the DBDH assumption. Let the hash functions H_1, H_2, and H_3 used in the scheme be modeled as the random oracles. Suppose that the DBDH assumption holds; the proposed scheme is a secure key agreement in our security model.*

Proof. Suppose an adversary $\mathscr{A}$ is an attack algorithm that breaks our scheme in the probability ϵ; we will show how to use the ability of $\mathscr{A}$ to build an algorithm $\mathscr{B}$ that solves the DBDH assumption with probability of at least ϵ'. Thus, $\mathscr{A}$'s advantage must be negligible because the DBDH assumption holds. □

We refer to $\mathscr{B}$ as "the simulator" because it simulates a real attacking environment for $\mathscr{A}$. $\mathscr{B}$ is initialized with the DBDH parameters $\{G_1, G_2, \hat{e}, q\}$ and the points $\{P, aP, bP, cP \in G_1, D, R \in G_2\}$, $D = \hat{e}(P,P)^{abc}$, and $R = \hat{e}(P,P)^r$. The idea of the proof is that $\mathscr{B}$ will embed the DBDH problem into the queries issued by $\mathscr{A}$. Since the hash function H_2 is modeled as random oracle, after the adversary issues the test query, it has only two unneglected cases to distinguish the tested session key sk_1 or sk_2 from a random string.

Case 1 (key-replication attack). The adversary $\mathscr{A}$ forces a nonmatching session to have the same session key with the Test($\prod_{A,B}^s$). In this case, the adversary $\mathscr{A}$ can get the session key by querying the nonmatching session. However, the input of hash function H_2 includes the entities' identities and the random nonce. Furthermore, they and a timestamp are integrally protected by H_3. For example, in Step 1, the session key $\mathrm{sk}_1 = H_2(K_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ includes the identities $\mathrm{AD}_{\mathrm{PH}_i}$ and $\mathrm{AD}_{\mathrm{SN}_{i,j,d}}$ and the random nonce R_1. The certification value $V_1 = H_3(\mathrm{sk}_1 \parallel M_1 \parallel T_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ includes them and the timestamp T_1. Therefore, two nonmatching sessions cannot have the same values and when H_2 and H_3 are modeled as a random oracle, the success probability of key-replication attack is negligible.

Case 2 (forging attack). The adversary $\mathscr{A}$ queries H_2 on the value $H_2(K_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ or $H_2(K_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}})$ in the test query. Obviously, in this case the adversary $\mathscr{A}$ can compute the value K_1 or K_2 by itself.

In the following, we mainly analyze the Case 2 forging attack. A simulator $\mathscr{B}$ is interested in using the $\mathscr{A}$ to turn $\mathscr{A}$'s advantage in distinguishing the tested session key from a random string into an advantage in solving the DBDH problem. During the game, $\mathscr{B}$ has to answer all queries of the $\mathscr{A}$.

Setup. $\mathscr{B}$ simulates the *Setup* algorithm as follows. $\mathscr{B}$ starts by choosing security and public parameters for our scheme using its input DBDH parameters $\{G_1, G_2, \hat{e}, q\}$ and $\{P, aP, bP, cP \in G_1,\ Q = D \text{ or } R,\ D = \hat{e}(P,P)^{abc},\ R = \hat{e}(P,P)^r\}$. $\mathscr{B}$ also chooses a random master key set $\{s_1, s_2, s_3, sr_i \mid i \in (1, \ldots, N)\}$ from Z_q^*, as the PKG would do. Using these keys, $\mathscr{B}$ sets the random generator $P_0 = P$, and then SV's public parameters are params $= \{q, G_1, G_2, P, \hat{e}, H_1, H_2, H_3, cP, s_2P, s_3P, sr_i cP, s_2 sr_i P, s_3 sr_i P \mid i \in (1, \ldots, N)\}$. $\mathscr{B}$ invokes the adversary $\mathscr{A}$, providing it with the public parameters params. Note that the DBDH parameters cP have been embedded in the game and the simulator $\mathscr{B}$ has no idea about the value c. With the probability at least $1/n(k)^2$, $\mathscr{B}$ guesses the adversary $\mathscr{A}$ will select one patient $\mathrm{ID}_{\mathrm{PA}_{i,j}}$ and his/her physician $\mathrm{ID}_{\mathrm{PH}_i}$. With the probability at least $1/s(k)$, $\mathscr{B}$ guesses the adversary $\mathscr{A}$ will select the session s as test session.

Queries. When the adversary $\mathscr{A}$ makes his/her queries, the simulator $\mathscr{B}$ answers the queries in arbitrary order as follows. Note that $\mathrm{ID}_{\mathrm{PH}_i}$, $\mathrm{ID}_{\mathrm{GW}_{i,j}}$, and $\mathrm{ID}_{\mathrm{SN}_{i,j,d}}$ are the guessed victims physician and devices.

$H_1(\cdot)$. In order to enhance simulation's fidelity, $\mathscr{B}$ maintains an initially empty list H_1^{list} of tuples $(\mathrm{ID}_A, R_A, u_A, h_{A,R_A}) \in \{0,1\}^* \times G_1 \times Z_q^* \times \{0,1\}^k$. When $\mathscr{A}$ queries the oracle H_1 as an input $(\mathrm{ID}_A \parallel R_A)$, $\mathscr{B}$ responds to the query in the following way.

(i) $\mathscr{B}$ checks the list H_1^{list}; if (ID_A and R_A) are already there, then $\mathscr{B}$ responds with stored value h_{A,R_A}.

(ii) Otherwise, if $\mathrm{ID}_A = \mathrm{ID}_{\mathrm{PH}_i}$ and $R_A = sr_i$, $\mathscr{B}$ randomly chooses $u_{\mathrm{PH}_i} \in Z_q^*$, and it computes $h_{\mathrm{PH}_i, sr_i} = u_{\mathrm{PH}_i} aP$. Then, it inserts $(\mathrm{ID}_{\mathrm{PH}_i}, sr_i, u_{\mathrm{PH}_i}, h_{\mathrm{PH}_i, sr_i})$ into the H_1^{list}. Finally, it responds with $H_1(\mathrm{ID}_{\mathrm{PH}_i} \parallel sr_i) = h_{\mathrm{PH}_i, sr_i}$.

(iii) Otherwise, if $\mathrm{ID}_A = \mathrm{ID}_{\mathrm{GW}_{i,j}}$, $\mathscr{B}$ randomly chooses $u_{\mathrm{GW}_{i,j}} \in Z_q^*$ and computes the value $h_{\mathrm{GW}_{i,j}, a_{i,j}} = u_{\mathrm{GW}_{i,j}} bP$. Then, it inserts $(\mathrm{ID}_{\mathrm{GW}_{i,j}}, a_{i,j}, u_{\mathrm{GW}_{i,j}}, h_{\mathrm{GW}_{i,j}, a_{i,j}})$

into the H_1^{list}. Here, $a_{i,j} = R_A$. Finally, it responds with $H_1(\text{ID}_{\text{GW}_{i,j}} \parallel a_{i,j}) = h_{\text{GW}_{i,j},a_{i,j}} = u_{\text{GW}_{i,j}} bP$.

(iv) Otherwise, if $\text{ID}_A = \text{ID}_{\text{SN}_{i,j,d}}$, $\mathscr{B}$ randomly chooses $u_{\text{SN}_{i,j,d}} \in Z_q^*$ and computes the value $h_{\text{SN}_{i,j,d},a_{i,j}} = u_{\text{SN}_{i,j,d}} bP$. Then, it inserts $(\text{ID}_{\text{SN}_{i,j,d}}, a_{i,j}, u_{\text{SN}_{i,j,d}}, h_{\text{SN}_{i,j,d},a_{i,j}})$ into the H_1^{list}. Here, $a_{i,j} = R_A$. Finally, it responds with $H_1(\text{ID}_{\text{GW}_{i,j}} \parallel a_{i,j}) = h_{\text{GW}_{i,j},a_{i,j}} = u_{\text{SN}_{i,j,d}} bP$.

(v) Otherwise, $\mathscr{B}$ randomly chooses $u_A \in Z_q^*$, computes $h_{A,R_A} = u_A P$, and inserts $(\text{ID}_A, R_A, u_A, h_{A,R_A})$ in the list. Finally, it responds with $H_1(\text{ID}_A \parallel R_A) = h_{A,R_A} = u_A P$.

$H_2(\cdot)$. The simulator $\mathscr{B}$ maintains an initially empty list H_2^{list} with entries of the form $(K_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B, h_{AB,R_{AB}}) \in G_2 \times G_1^3 \times \{0,1\}^k$. When $\mathscr{A}$ queries the oracle H_2 as a input $(K_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B)$, the simulator $\mathscr{B}$ responds to the query in the following way.

(i) $\mathscr{B}$ checks the list H_2^{list}; if $(K_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B)$ is already there, $\mathscr{B}$ responds with the value $h_{AB,R_{AB}}$.

(ii) Otherwise, $\mathscr{B}$ randomly chooses $h_{AB,R_{AB}} \in \{0,1\}^k$ and sends back the value to $\mathscr{A}$. Finally, $\mathscr{B}$ stores the new tuple $(K_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B, h_{AB,R_{AB}})$ in the list H_2^{list}.

$H_3(\cdot)$. The simulator $\mathscr{B}$ maintains an initially empty list H_3^{list} with entries of the form $(K_{AB}, M_{AB}, T_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B, h_{AB,T_{AB}}) \in \{0,1\}^k \times \{0,1\}^* \times G_1^3 \times \{0,1\}^k$. The simulator $\mathscr{B}$ responds to these queries in the following ways.

(i) $\mathscr{B}$ checks the list H_3^{list}; if $(K_{AB}, M_{AB}, T_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B)$ is already there, $\mathscr{B}$ responds with the value $h_{AB,R_{AB}}$.

(ii) Otherwise, $\mathscr{B}$ randomly chooses $h_{AB,R_{AB}} \in \{0,1\}^k$ and sends back $h_{AB,R_{AB}}$ to $\mathscr{A}$. Finally, $\mathscr{B}$ stores the new tuple $(K_{AB}, M_{AB}, T_{AB}, R_{AB}, \text{AD}_A, \text{AD}_B, h_{AB,T_{AB}})$ in the list H_3^{list}.

Establish(ID_A). When receiving this query, $\mathscr{B}$ responds to the query in the following way.

(i) If ID_A is the target physician or the target patient's gateway or sensor nodes, $\mathscr{B}$ aborts the game.

(ii) Otherwise, if ID_A is a physician PH_j, $\mathscr{B}$ looks in H_1^{list} for the entries $(\text{ID}_{\text{SV}_j}, sr_j, u_{\text{SV}_j}, h_{\text{SV}_j,sr_j})$ and $(\text{ID}_{\text{PH}_j}, sr_j, u_{\text{PH}_j}, h_{\text{PH}_j,sr_j})$. Then, $\mathscr{B}$ returns $\{(sr_j u_{\text{PH}_j} cP, s_2 sr_j, s_3 sr_j) \parallel u_{\text{SV}_j} P, u_{\text{PH}_j} P\}$.

(iii) Otherwise, if ID_A is a patient's $\text{GW}_{j,k}$, $\mathscr{B}$ looks in H_1^{list} for the entries $(\text{ID}_{\text{SV}_j}, sr_j, u_{\text{SV}_j}, h_{\text{SV}_j,sr_j})$, $(\text{ID}_{\text{PH}_j}, sr_j, u_{\text{PH}_j}, h_{\text{PH}_j,sr_j})$, and $(\text{ID}_{\text{GW}_{j,k}}, sr_j, u_{\text{GW}_{j,k}}, h_{\text{GW}_{j,k},sr_j})$. Then, $\mathscr{B}$ returns $\{(sr_j u_{\text{PH}_j} cP, s_2 sr_j u_{\text{GW}_{j,k}} P) \parallel u_{\text{SV}_j} P, u_{\text{PH}_j} P, u_{\text{GW}_{j,k}} P\}$.

(iv) Otherwise, if ID_A is a physician $\text{SN}_{j,k,d}$, the simulator $\mathscr{B}$ looks in H_1^{list} for the entries $(\text{ID}_{\text{SV}_j}, sr_j, u_{\text{SV}_j}, h_{\text{SV}_j,sr_j})$, $(\text{ID}_{\text{PH}_j}, sr_j, u_{\text{PH}_j}, h_{\text{PH}_j,sr_j})$, $(\text{ID}_{\text{GW}_{j,k}}, sr_j, u_{\text{GW}_{j,k}}, h_{\text{GW}_{j,k},sr_j})$, and $(\text{ID}_{\text{SN}_{j,k,d}}, sr_j, u_{\text{SN}_{j,k,d}}, h_{\text{SN}_{j,k,d},sr_j})$. Then, $\mathscr{B}$ returns $\{(sr_j u_{\text{PH}_j} cP, s_3 sr_j u_{\text{SN}_{j,k,d}} P) \parallel u_{\text{SV}_j} P, u_{\text{PH}_j} P, u_{\text{GW}_{j,k}} P, u_{\text{SN}_{j,k,d}} P\}$.

Send($\prod_{A,B}^s, D$). Since the H_2 and H_3 are the random oracles, the adversary cannot change the communication message. the simulator $\mathscr{B}$ needs only to store the values according to the scheme. Moreover, the parameters are included in the data D, which can be found in the lists H_3^{list} and H_2^{list}.

Reveal($\prod_{A,B}^s$). $\mathscr{B}$ maintains a list sk^{list} with tuples of the form $(\text{ID}_A, \text{ID}_B, R_{AB}, \prod_{A,B}^s)$. The simulator $\mathscr{B}$ responds to the query in the following way.

(i) If ID_A and ID_B are the target physician and the target patient's gateway or sensor nodes, $\mathscr{B}$ aborts the game.

(ii) Otherwise, if ID_A is a target physician PH_i and ID_B is not target patients' facilities, $\mathscr{B}$ proceeds in the following way to respond:

(a) If ID_B is an identity of gateway, $\mathscr{B}$ computes $K_{AB} = \hat{e}(u_{\text{GW}_{i,j^*}} cP, sr_i u_{\text{PH}_i} aP) \cdot \hat{e}(s_2 sr_i u_{\text{GW}_{i,j^*}} P, R_2)$. Then, $\mathscr{B}$ finds the value $h_{AB,R_{AB}}$ from H_2^{list} and returns $h_{AB,R_{AB}}$ as the response.

(b) Otherwise ID_B should be an identity of sensor node; the simulator $\mathscr{B}$ computes $K_{AB} = \hat{e}(u_{\text{SN}_{i,j^*,d^*}} cP, sr_i u_{\text{PH}_i} aP) \cdot \hat{e}(R_1, s_3 sr_i u_{\text{PH}_i} aP)$. Then, $\mathscr{B}$ finds the value $h_{AB,R_{AB}}$ from H_2^{list} and returns $h_{AB,R_{AB}}$ as the response.

(iii) Otherwise, if ID_A is another physician PH_j and ID_B is his/her patients facilities, $\mathscr{B}$ proceeds in the following way to respond:

(a) If ID_B is an identity of gateway, $\mathscr{B}$ computes $K_{AB} = \hat{e}(u_{\text{GW}_{j,j^*}} cP, sr_j u_{\text{PH}_j} P) \cdot \hat{e}(s_2 sr_j u_{\text{GW}_{j,j^*}} P, R_2)$. Then, $\mathscr{B}$ finds the value $h_{AB,R_{AB}}$ from H_2^{list} and returns $h_{AB,R_{AB}}$ as the response.

(b) Otherwise ID_B should be an identity of sensor node; $\mathscr{B}$ computes $K_{AB} = \hat{e}(u_{\text{SN}_{j,j^*,d^*}} cP, sr_j u_{\text{PH}_j} P) \cdot \hat{e}(R_1, s_3 sr_j u_{\text{PH}_j} P)$. Then, $\mathscr{B}$ finds the value $h_{AB,R_{AB}}$ from H_2^{list} and returns $h_{AB,R_{AB}}$ as the response.

Test($\prod_{A,B}^s$). $\mathscr{A}$ issues a test query. Suppose the identity tuple of the first node A is ID_{PH_i} and the second target node B is $\text{ID}_{\text{GW}_{i,j}}$ or $\text{ID}_{\text{SN}_{i,j,d}}$.

(i) If A and B do not belong to our guessed victims PH_i and $\text{PA}_{i,j}$, $\mathscr{B}$ aborts the game.

(ii) Otherwise, $\mathscr{B}$ queries AD_{SV_i}, AD_{PH_i}, $\text{AD}_{\text{GW}_{i,j}}$, and $\text{AD}_{\text{SN}_{i,j,d}}$.

(a) If $B = \mathrm{GW}_{i,j}$, $\mathscr{B}$ computes $K_2 = Q^{u_{\mathrm{GW}_{i,j}} u_{\mathrm{PH}_i} sr_i} \cdot \hat{e}(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2)^{s_2 sr_i}$.

(b) Otherwise, $\mathscr{B}$ computes $K_1 = Q^{u_{\mathrm{SN}_{i,j,d}} u_{\mathrm{PH}_i} sr_i} \cdot \hat{e}(R_1, \mathrm{AD}_{\mathrm{PH}_i})^{s_3 sr_i}$.

$\mathscr{B}$ looks in the list H_2^{list} and returns the value sk_1 or sk_2 to the adversary $\mathscr{A}$.

The test query is answered by $\mathscr{B}$ with its DBDH input D or R. Consider the following two cases:

(i) If $Q = D$, since $D = \hat{e}(P,P)^{abc}$ in the DBDH instance, then

$$\begin{aligned} K_1 &= D^{u_{\mathrm{SN}_{i,j,d}} u_{\mathrm{PH}_i} sr_i} \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i} \\ &= \hat{e}\left(u_{\mathrm{SN}_{i,j,d}} bP, u_{\mathrm{PH}_i} aP\right)^{sr_i c} \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(R_1, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_3 sr_i}, \\ K_2 &= D^{u_{\mathrm{GW}_{i,j}} u_{\mathrm{PH}_i} sr_i} \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_2 sr_i} \\ &= \hat{e}\left(u_{\mathrm{GW}_{i,j}} bP, u_{\mathrm{PH}_i} aP\right)^{sr_i c} \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_2 sr_i} \\ &= \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, \mathrm{AD}_{\mathrm{PH}_i}\right)^{s_1 sr_i} \cdot \hat{e}\left(\mathrm{AD}_{\mathrm{GW}_{i,j}}, R_2\right)^{s_2 sr_i}. \end{aligned} \tag{18}$$

Thus, the response by $\mathscr{B}$ corresponds to the real values sk_1 and sk_2.

(ii) If $Q = R$, since $R = \hat{e}(P,P)^r$ is random, then the response by $\mathscr{B}$ to the test query of $\mathscr{A}$ is a random element in G_2.

If the adversary $\mathscr{A}$ succeeds in getting the session key sk_1 or sk_2, it shall distinguish between the value sk_1 or sk_2 and a random value; then, it outputs the correct bit $b = 1$ or $b = 0$. $\mathscr{B}$ can give the correct answer to the DBDH problem by using $\mathscr{A}$'s output.

The success probability of $\mathscr{B}$ is

$$\epsilon' \geq \frac{1}{s(k)\, n(k)^2\, t(k)} \epsilon. \tag{19}$$

Here, ϵ is the probability that the adversary $\mathscr{A}$ succeeds in launching the attack. $t(k)$ is the polynomial bound on the number of the adversary $\mathscr{A}$'s queries.

If the adversary $\mathscr{A}$ succeeds with nonnegligible probability to attack our scheme, we can also solve the DBDH problem with a nonnegligible probability. Thus, our scheme is based on the DBDH problem.

6.3. Security Analysis. In the following, we will directly analyze how our proposed scheme achieves entity anonymity and untraceability and resists collusion attack and whether the security requirements have been satisfied.

Proposition 7. *The proposed scheme can resist the replay attack.*

Proof. It should be noted that our proposed scheme inherits the structure of FNKAP. We also use the random numbers r_1 and r_2 to achieve the freshness key agreement. The adversary cannot compute the r_i, $i \in \{1,2\}$ from $R_1 = r_1 \mathrm{AD}_{\mathrm{SN}_{i,j,d}}$ and $R_2 = r_2 \mathrm{AD}_{\mathrm{PH}_i}$ because of the difficulty of the ECDL problem. Moreover, the proposed scheme can efficiently resist the replay attack by considering the following scenarios. (1) An adversary cannot replay the data package D_1 to cheat SV and PH_i. During the Key Agreement from SN to PH Phase, when SV receives a data package D_1, it verifies the timestamp T_1 with the current time. If the data package is a replay attack, SV will detect it. Moreover, if the adversary changes the timestamp T_1 in D_1, SV will find the behavior by checking the equation $V_1^* = H_3(\mathrm{sk}_1^* \parallel M_1 \parallel T_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ because it cannot obtain the session key sk_1^*. (2) An adversary cannot replay the data package D_2 to cheat SV and PH_i. Similar to the above, an adversary cannot replay the data package D_2 to cheat SV and $\mathrm{PA}_{i,j}$. During the key agreement from PH to GW Phase, when SV receives a data package D_2, it verifies the timestamp T_2 with the current time. If the data package is a replay attack, then SV will detect it. Moreover, if the adversary changes the timestamp T_2 in D_2, SV will find the behavior by checking the equation $V_2^* = H_3(\mathrm{sk}_2^* \parallel M_2 \parallel T_2 \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}})$ because it cannot know the session key sk_2^*. □

Proposition 8. *The proposed scheme can provide basic forward secrecy.*

Proof. To establish session key between SN and PH, $\mathrm{SN}_{i,j,d}$ and PH_i use various $r_1 \mathrm{AD}_{\mathrm{SN}_{i,j,d}}$ for each session. Thus, the current session key $\mathrm{sk}_1 = H_2(K_1 \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ is disclosed, and an adversary cannot obtain the information about $K_1 = \hat{e}(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}) \cdot \hat{e}(s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i})^{r_1}$. In other words, the adversary cannot get more opportunities to guess previous key $\mathrm{sk}_1^* = H_2(K_1^* \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ than before, even if he/she knows the current key sk_1. Similarly, because R_2 is equal to $r_2 \mathrm{AD}_{\mathrm{PH}_i}$, the adversary cannot gain any benefits to guess previous key $\mathrm{sk}_2^* = H_2(K_2^* \parallel R_2 \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel \mathrm{AD}_{\mathrm{GW}_{i,j}})$ between PH and GW compared to before, even if he/she knows the current key sk_2. Thus, our proposal can provide basic forward secrecy. □

Proposition 9. *The proposed scheme can prevent fraud attack.*

Proof. Our proposal provides mutual authentication between PH_i and $\mathrm{GW}_{i,j}$ or PH_i and $\mathrm{SN}_{i,j,d}$. The proposed scheme can prevent fraud attack by considering the following scenarios. (1) An adversary cannot impersonate $\mathrm{SN}_{i,j,d}$ to cheat PH_i. PH_i can authenticate $\mathrm{SN}_{i,j,d}$ by verifying V_1' in Step 3. Since the adversary cannot obtain $s_3 sr_i$ or $s_3 sr_i r_1 \mathrm{AD}_{\mathrm{SN}_{i,j,d}}$, he/she cannot compute $K_1' = \hat{e}(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}) \cdot \hat{e}(R_1, \mathrm{AD}_{\mathrm{PH}_i})^{s_3 sr_i}$, $K_1^* = \hat{e}(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i})^{s_1 sr_i} \cdot \hat{e}(R_1, \mathrm{AD}_{\mathrm{PH}_i})^{s_3 sr_i}$, or $K_1 = \hat{e}(\mathrm{AD}_{\mathrm{SN}_{i,j,d}}, s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}) \cdot \hat{e}(s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d}}, \mathrm{AD}_{\mathrm{PH}_i})^{r_1}$. Thus, the adversary cannot get $\mathrm{sk}_1' = H_2(K_1' \parallel R_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i})$ and $V_1' = H_3(\mathrm{sk}_1' \parallel R_1 \parallel M_1 \parallel \mathrm{AD}_{\mathrm{SN}_{i,j,d}} \parallel \mathrm{AD}_{\mathrm{PH}_i} \parallel T_1)$, sequentially. Thus, the adversary cannot generate the valid verifier to PH_i. (2) An adversary cannot impersonate PH_i to cheat $\mathrm{GW}_{i,j}$. Similar to the above, $\mathrm{GW}_{i,j}$ can authenticate PH_i by verifying V_2' in Step 3. Since the adversary cannot obtain $s_2 sr_i$ or $s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}}$, he/she cannot compute K_2', K_2^*, or K_2. Thus, the adversary cannot get sk_2' and V_2', sequentially. Thus, the adversary cannot generate the valid verifier to $\mathrm{GW}_{i,j}$. □

Proposition 10. *The proposed scheme can provide entity anonymity and untraceability.*

Proof. In the proposed scheme, the adversary can obtain the amplified identities $H_1(\mathrm{ID}_{\mathrm{SV}} \parallel sr_i)$, $H_1(\mathrm{ID}_{\mathrm{PH}_i} \parallel sr_i)$, $H_1(\mathrm{ID}_{\mathrm{GW}_{i,j}} \parallel a_{i,j})$, and $H_1(\mathrm{ID}_{\mathrm{SN}_{i,j,d}} \parallel a_{i,j})$ instead of $H(\mathrm{ID}_{\mathrm{SV}})$, $H(\mathrm{ID}_{\mathrm{PH}_i})$, $H(\mathrm{ID}_{\mathrm{GW}_{i,j}})$, and $H(\mathrm{ID}_{\mathrm{SN}_{i,j,d}})$ in Steps K1 and K3. Here, sr_i and $a_{i,j}$ are big random numbers in Z_q^*. Therefore, the adversary cannot verify whether the guessed identity is correct or incorrect by testing all possible identities without the secret sr_i and $a_{i,j}$. For example, to guess $H_1(\mathrm{ID}_{\mathrm{PH}_i} \parallel sr_i)$, the adversary should input the guess values of $\mathrm{ID}_{\mathrm{PH}_i}$ and sr_i at the same time. Suppose the identity $\mathrm{ID}_{\mathrm{PH}_i}$ is composed of m bits; it is infeasible for adversary to launch an exhausted search for 2^{m+q} possible solutions. Here, q is the group order of Z_q^*, and it is a big random number. In particular, if the physicians reregister on a period, sr_i would be fresh regularly. Thus, this risk of corruption will be lower to $\mathrm{ID}_{\mathrm{PH}_i}$. Moreover, $a_{i,j}$ is also a big random number in Z_q^*, and each patient has a different value. Even if it is the same patient, there are different values on the various diagnoses. Based on the similar reason, the adversary cannot know the identities of $\mathrm{ID}_{\mathrm{GW}_{i,j}}$ and $\mathrm{ID}_{\mathrm{SN}_{i,j,d}}$ or trace them. Furthermore, it is also intractable to derive the identity from $H_1(\mathrm{ID}_{\mathrm{SV}} \parallel sr_i)$, $H_1(\mathrm{ID}_{\mathrm{PH}_i} \parallel sr_i)$, $H_1(\mathrm{ID}_{\mathrm{GW}_{i,j}} \parallel a_{i,j})$, and $H_1(\mathrm{ID}_{\mathrm{SN}_{i,j,d}} \parallel a_{i,j})$ because H_1 is a secure one-way cryptography hash function. Thus, our proposal can achieve anonymity and untraceability. □

Proposition 11. *The proposed scheme can withstand the collusion attack.*

Proof. In our proposal, SV distributes different secret values $(s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}, s_2 sr_i, s_3 sr_i)$ for various physicians PH_i. Thus, the adversary physician $\mathrm{PH}_{\mathscr{A}}$ and his/her patients $\mathrm{PA}_{\mathscr{A},j}$ cannot get the victim's information $s_2 sr_{\mathrm{i}}$, $s_3 sr_i$, directly. Furthermore, the adversary $\mathscr{A}$ who has registered as a normal patient of the physician PH_i can legally obtain $(s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}, s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}})$ from SV. However, he/she cannot obtain $s_1 sr_i$ or $s_2 sr_i$ from $(s_1 sr_i \mathrm{AD}_{\mathrm{PH}_i}, s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}})$ except when he/she can solve the ECDL problem. Similarly, the adversary $\mathscr{A}$ cannot obtain $s_3 sr_i$ from $s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d}}$ because of the difficulty of the ECDL problem. Obviously, our scheme destructs the attack conditions at Steps A1 and A2 in Section 4. As a result, the scheme can resist the collusion attack and prevent the adversary from generating the session keys sk_1 and sk_2. Furthermore, if an insider adversary wants to attack the key agreement from SN to PH, he/she should get the secure information about $s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d}}$. The adversary receives up to $s_3 sr_i \mathrm{AD}_{\mathrm{SN}_{i,j,d^*}}$; he/she cannot get the information except whens he/she can solve the ECDL problem. Similarly, an insider adversary cannot launch an attack to the key agreement from PH to GW phase, because he/she cannot get the value $s_2 sr_i \mathrm{AD}_{\mathrm{GW}_{i,j}}$. Thus, our proposal resists the collusion attack, effectively. □

7. Functionality and Performance Comparison

In this section, security and functionality are compared between our scheme and FNKAP. Then, we illustrate a comparison of the communication and computation costing performances.

7.1. Functionality Comparison. As shown in Table 1, our scheme not only provides the functionality in [8] but also resists the collusion attack. Therefore, we can conclude that the proposed scheme achieves a higher security level than FNKAP.

7.2. Performance Comparison. To compare the actual computational costs, we have implemented our scheme and Kim's scheme with JPBC Library (Java Pairing-Based Cryptography Library [19]) in an ARM platform and a desktop platform. The detailed parameters of the platform are listed in Table 2. To provide a similar environment in WHMS, the weak processing ability is simulated on an android smartphone (HTC M7) running Android 4.1 with Snapdragon APQ8064 1.7 GHz, and the powerful processing ability is simulated on a desktop computer running Windows 7 with Intel Core i5-3470.

Table 3 summarizes the detailed parameters about the elliptic curve and pairing parameters for JPBC. We use a 512 bits elliptic curve $y^3 = x^3 + x$ to evaluate our scheme in the platforms. In Table 4, top row is the results in the ARM platform, and the second row is the result in the desktop platform. Here, all the experiment results are averaged over 10 independent runs.

In order to provide detailed comparison, we test the basic operation in Z_q^*, G_1, and G_2, separately. The time of a pairing computation is indicated by T_p. The time of a hash operation is indicated by T_h. The time complexity of computing multiplication in Z_q^*, G_1, and G_2 is indicated by T_{m_q}, T_{m_1}, and T_{m_2}, respectively. The time of the addition in Z_q^* and G_1 is indicated by T_{a_q} and T_{a_1}, independently. The time of the exponentiation in G_2 is indicated by T_e. Note that the time of hash operation T_h is the smallest because it needs very limited computation. On the contrary, the time of pairing operation T_p is the highest consumption.

Tables 5 and 6 illustrate the performance comparison with Kim's scheme. In Tables 5 and 6, the notation id is a unit length of identity; the notation pr is a unit length of private key. First, in order to achieve the session key freshness, we maintain one-round communication to exchange a random value R_1 or R_2 in FNKAP. Second, our scheme increases the amplified identity randomness against the passive offline attack. However, the amplified identity space is equal to that of FNKAP because the amplified identity is still a hash value. Third, the private key space of GW and SN decreases because

Table 1: Security and functionality comparison with Kim's scheme.

	Replay and fraud attack	Basic forward secrecy	Anonymity and untraceability	Collusion attack
Kim's	Resistance	Yes	Yes	No
Ours	Resistance	Yes	Yes	Resistance

Table 2: Detailed platform parameters.

Testbed	CPU	RAM	Operating system
ARM	Snapdragon APQ8064 1.7 GHz	2 GB	Android 4.1
Desktop	Intel Core i5 3.2 × 4 GHz	4 GB	Windows 7–32 bits and Open JDK 1.8.0

Table 3: Detailed elliptic curve and pairing parameters.

Elliptic curve	Group order of Z_q^*	Element size in G_1
Type A: $y^3 = x^3 + x$	Around 160 bits	Around 512 bits

Table 4: Detailed performance parameters.

	T_p	T_h	T_{m_q}	T_{a_q}
ARM	442 ms	≪1 ms	<1 ms	<1 ms
Desktop	25 ms	≪1 ms	≪1 ms	=1 ms
	T_e	T_{m_2}	T_{m_1}	T_{a_1}
ARM	450 ms	1 ms	124 ms	<1 ms
Desktop	4 ms	≪1 ms	22 ms	<1 ms

Table 5: Basic performance comparison with Kim's scheme [8].

	Round	Private key space			
		SV	PH	GW	SN
Kim's	One	4pr	4pr	4pr	4pr
Ours	One	$(3 + N)$pr	3pr	2pr	2pr

we reduce the redundancy of private key information to GW and SN. Moreover, it shrinks the risk of insider attack because only SV knows total secure information. Fourth, the computation time of our scheme is near half of FNKAP because we decrease half of the pairing operations. Finally, we should point out that our scheme computation and store cost for the SV are higher than those of Kim's work. More precisely, we should choose and store N random numbers more than FNKAP, and $4N$ multiplications in Z_p^* should be added in Initial Section. Commonly, the above propositions only increase the computation cost and the storage requirement in SV. SV has enough computing and storing power to hold the operations because the u-Health Server is usually a server cluster. Furthermore, the computing operations are only increased in Initial Phase. For the resources limited entities GW and SN, the computation and storage requirements do not increase instead of decreasing. Thus, the scheme is feasible to key agreement in WHMS.

Our proposed scheme inherits the advantage of Kim's hierarchical scheme in WHMS. At the same time, our scheme provides security enhancement against collusion attack in our security model. Furthermore, it preserves the low computation and private key space in SN and GW compared to FNKAP. Therefore, it is an enhanced security hierarchical key agreement scheme with the noninteractive property that is suitable for the application in WHMS.

8. Conclusions

In this paper, we have illustrated that there is a security weakness in Kim's work [8] under a practical security model with the physicians corruption. The security flaw is due to the fact that the physicians' parts of the private key are the same. Therefore, the adversary, as a legal physician, can acquire the entire patient's private information. To enhance the scheme, we proposed an authenticated key agreement scheme which randomizes each physician's private key. Moreover, we have reduced the numbers of the private keys and the operations of the bilinear pairing. Thus, the performance of our scheme is more suitable for the WHMS environment than Kim's work. We also prove the security of our scheme. The proof shows that the proposed scheme is secure under the DBDH assumption in the random oracle model.

Notations

PH_i:	The ith physician
$\mathrm{PA}_{i,j}$:	The ith physician's jth patient
SV:	The u-Health Server
$\mathrm{GW}_{i,j}$:	The $\mathrm{PA}_{i,j}$'s gateway
$\mathrm{SN}_{i,j,d}$:	The $\mathrm{PA}_{i,j}$'s dth sensor node
ID_A:	The identity of an entity A
AD_A:	The amplified identity of ID_A
sk_1 and sk_2:	The session key established between two entities
$H(\cdot)$, $H_1(\cdot)$, $H_2(\cdot)$, and $H_3(\cdot)$:	The cryptographic hash functions
$E_K(M)$:	Encryption of a message M using an symmetric key K
$\cdot$:	Multiplication operator
‖:	Concatenation operator.

Table 6: Computation performance comparison with Kim's scheme [8].

	ID space	Computation		
		PH	SN/GW	Sum
Kim's SN to PH	4id	$T_h + 2T_{a_1} + 4T_p + 3T_{m_2} \approx 102$ ms	$T_h + 2T_{a_1} + 4T_p + 3T_{m_2} \approx 1773$ ms	1875 ms
Our SN to PH	4id	$2T_h + T_{a_1} + 2T_p + T_{m_2} \approx 51$ ms	$2T_h + 2T_{a_1} + 2T_p + T_{m_2} \approx 887$ ms	938 ms
Kim's PH to GW	4id	$T_h + 3T_{a_1} + 4T_p + 3T_{m_2} \approx 103$ ms	$T_h + T_{a_1} + 4T_p + 3T_{m_2} \approx 1772$ ms	1875 ms
Our PH to GW	4id	$2T_h + 2T_{a_1} + 2T_p + T_{m_2} \approx 52$ ms	$2T_h + 2T_p + T_{m_2} \approx 885$ ms	937 ms

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors thank Dr. Jianghong Wei, Dr. Haosu Cheng, Dr. Hui Lu, and Dr. Honghao Zhao for their support. This paper is supported by the National Key Basic Research Program (973 Program) through Project 2012CB315905, by the Natural Science Foundation through Projects 61370190, 61272501, 61173154, 61402029, and 61003214, and by the Beijing Natural Science Foundation through Projects 4132056 and 4122041.

References

[1] V. Custodio, F. J. Herrera, G. López, and J. I. Moreno, "A review on architectures and communications technologies for wearable health-monitoring systems," *Sensors*, vol. 12, no. 10, pp. 13907–13946, 2012.

[2] F. Touati and R. Tabish, "U-healthcare system: state-of-the-art review and challenges," *Journal of Medical Systems*, vol. 37, no. 3, pp. 9949–9969, 2013.

[3] H. F. Rashvand, V. Traver Salcedo, E. Montón Sánchez, and D. Iliescu, "Ubiquitous wireless telemedicine," *IET Communications*, vol. 2, no. 2, pp. 237–254, 2008.

[4] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mHealthcare social network," *Mobile Networks and Applications*, vol. 16, no. 6, pp. 683–694, 2011.

[5] W. Liu, J. Liu, Q. Wu, W. Susilo, H. Deng, and B. Qin, "SAKE: scalable authenticated key exchange for mobile e-health networks," *Security and Communication Networks*, 2015.

[6] X. Lin, R. Lu, X. S. 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.

[7] Q. Huang, X. Yang, and S. Li, "Identity authentication and context privacy preservation in wireless health monitoring," *International Journal of Computer Network and Information Security*, vol. 4, pp. 53–60, 2011.

[8] H. Kim, "Freshness-preserving non-interactive hierarchical key agreement protocol over WHMS," *Sensors*, vol. 14, no. 12, pp. 23742–23757, 2014.

[9] D.-C. Lou, T.-F. Lee, and T.-H. Lin, "Efficient biometric authenticated key agreements based on extended chaotic maps for telecare medicine information systems," *Journal of Medical Systems*, vol. 39, no. 5, pp. 58–68, 2015.

[10] S. H. Erfani, H. H. S. Javadi, and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," *Security and Communication Networks*, vol. 8, no. 6, pp. 1040–1049, 2015.

[11] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt, and S. D. Wolthusen, "Strongly-resilient and non-interactive hierarchical key-agreement in MANETs," in *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS '08)*, pp. 49–65, Málaga, Spain, October 2008.

[12] R. Dupont and A. Enge, "Provably secure non-interactive key distribution based on pairings," *Discrete Applied Mathematics*, vol. 154, no. 2, pp. 270–276, 2006.

[13] Y. Yang, "Broadcast encryption based non-interactive key distribution in MANETs," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 533–545, 2014.

[14] H. Guo, Y. Mu, Z. Li, and X. Zhang, "An efficient and non-interactive hierarchical key agreement protocol," *Computers and Security*, vol. 30, no. 1, pp. 28–34, 2011.

[15] Chad Garland. Cedars-Sinai reports possible breach of patients' medical data, June 2015, http://www.latimes.com/business/la-fi-cedars-breach-20140823-story.html.

[16] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology— CRYPTO '93*, D. R. Stinson, Ed., vol. 773 of *Lecture Notes in Computer Science*, pp. 232–249, Springer, New York, NY, USA, 1994.

[17] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[18] D. Hankerson and A. Menezes, "Elliptic curve discrete logarithm problem," in *Encyclopedia of Cryptography and Security*, C. A. Van Tilborg and S. Jajodia, Eds., pp. 397–400, Springer US, 2011.

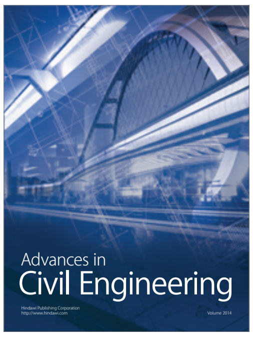
[19] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 850–855, IEEE, Kerkyra, Greece, July 2011.

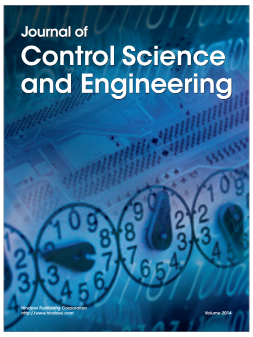
Journal of
Engineering

The Scientific World Journal

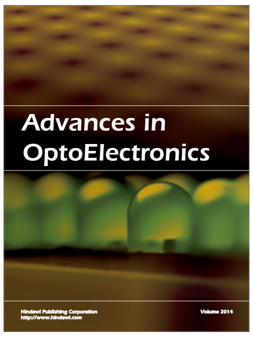
International Journal of
Rotating Machinery

Journal of
Sensors

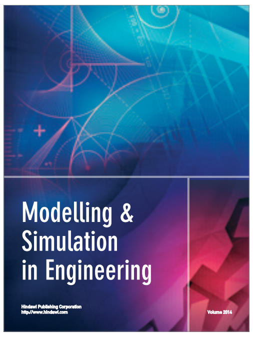
International Journal of
Distributed Sensor Networks

Advances in
Civil Engineering

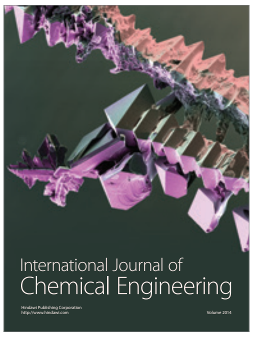
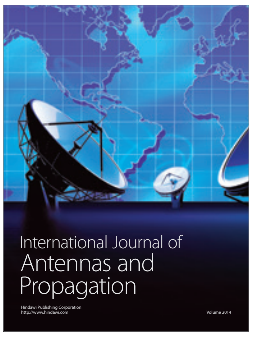

Journal of
Control Science and Engineering

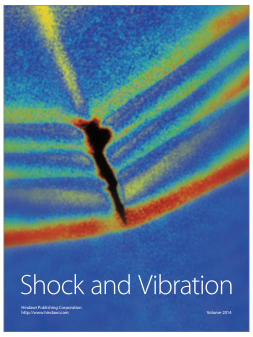
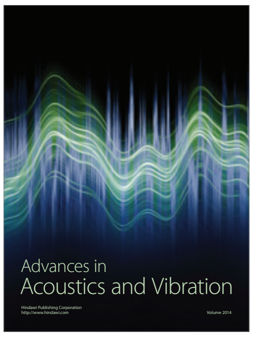
Journal of
Robotics

Journal of
Electrical and Computer Engineering

Advances in
OptoElectronics

VLSI Design

International Journal of
Navigation and Observation

Modelling & Simulation in Engineering

International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering

International Journal of
Antennas and Propagation

Active and Passive Electronic Components

Shock and Vibration

Advances in
Acoustics and Vibration