sensors

MDPI

Article

A Secure and Efficient Group Key Management Protocol with Cooperative Sensor Association in WBANs

Haowen Tan and Ilyong Chung *

Department of Computer Engineering, Chosun University, Gwangju 61452, Korea; tan_halloween@foxmail.com
* Correspondence: iyc@chosun.ac.kr; Tel.: +82-62-230-7712

Received: 22 October 2018; Accepted: 10 November 2018; Published: 14 November 2018

Abstract: The wireless body area network (WBAN) is considered as one of the emerging wireless techniques in the healthcare system. Typical WBAN sensors, especially implantable sensors, have limited power capability, which restricts their wide applications in the medical environment. In addition, it is necessary for the healthcare center (HC) to broadcast significant notifications to different patient groups. Considering the above issues, in this paper, the novel practical WBAN system model with group message broadcasting is built. Subsequently, a secure and efficient group key management protocol with cooperative sensor association is proposed. In the proposed protocol, the Chinese remainder theorem (CRT) is employed for group key management between HC and the personal controller (PC), which also supports batch key updating. The proposed sensor association scheme is motivated by coded cooperative data exchange (CCDE). The formal security proofs are presented, indicating that the proposed protocol can achieve the desired security properties. Moreover, performance analysis demonstrates that the proposed protocol is efficient compared with state-of-the-art group key management protocols.

Keywords: wireless body area networks; group key management; authentication; Chinese remainder theorem (CRT); coded cooperative data exchange (CCDE)

1. Introduction

Development of wireless communication and sensor technologies has enabled remarkable improvement in both academic research and practical applications of wireless body area networks (WBAN), which offer ubiquitous wireless communication services to users [1]. In the medical field, WBAN is used to monitor patients' real-time health status and seamlessly transmit physiological data to medical institutions including hospitals, community clinics and emergency centers. Consequently, the doctor could conduct remote diagnostics on the patients and provide timely medical assistance. Additionally, with necessary symptom detection, early warnings, as well as precautionary measurements for certain diseases including asthma, AIDS, cancer and influenza can be provided [2].

Nowadays, as a crucial part of the Internet of Thing (IoT), WBANs have continuously attracted much attention. Its architecture varies greatly, so as to adjust to diverse requirements of different practical scenarios. In general, a typical WBAN designed for the healthcare system mainly consists of the healthcare center (HC), personal controller (PC) and many low-power wireless medical sensors implanted inside or attached to the patient's body [1]. Through these sensors, vital biomedical information such as heartbeat and blood pressure can be measured and then transmitted to the healthcare center (HC) through the personal controller (PC). Therefore, the doctor or physician could be aware of a patient's real-time physical parameters by analyzing the acquired biomedical information. According to these analysis results, appropriate remote diagnostics and timely medical assistance are provided. Note that HC here refers to the healthcare service provider such as a hospital or clinic. PC is a mobile device responsible for both biomedical information gathering from sensors and communication with HC. It is assumed that each patient is combined with one PC. Particularly, in numerous scenarios, the role of PC is normally played by PDAs or smartphones. The applications and implementations of WBANs offer a better choice of receiving healthcare services for patients or other people who need to be taken good care of, for example aged or disabled people in need of long-term physiological monitoring.

The sensitive biomedical data transmission is through an open wireless channel, where the patients' private physical condition may be revealed to an unauthorized entity. Consequently, appropriate strategies are urgently required to guarantee enough security properties and privacy protections.

In practical WBAN scenarios, the HC is responsible for providing medical services to large numbers of patients simultaneously [3,4]. Meanwhile, the patients with different diseases are allocated to different departments. For instance, patients with coronary disease are arranged with the cardiovascular department, while patients with skin disease such as allergic dermatitis are arranged with the dermatological department. In consideration of this, it is essential for the HC to provide push notification service to patients of different departments, respectively. Patients of the same department could also exchange their information on certain diseases. Therefore, a specific group communication channel between HC and patients is indispensable, which enables notification broadcast and information exchange between HC and patients. Moreover, the patients may frequently join or leave the healthcare system. In this way, efficient group key management employing join and revocation operation is of great significance [5]. With the generated group key, HC could broadcast notifications to a specific patient group, and patients of this group could communicate with each other as well [6].

WBAN sensors, including implantable sensors and wearable sensors, are low-power wireless devices with limitations on computation, communication, power and storage [1,2]. In particular, for implantable sensors, which are designed to be implanted in the human body in order to monitor essential physiological parameters, it is impractical to frequently recharge or change the built-in battery. Meanwhile, an increased computation and transmission load on the sensor side can dissipate more power into heat and eventually do harm to human organs [7]. In this assumption, the transmission passes, as well as the computation load, should be considerably optimized for the purpose of prolonging the working time period and preventing the human body from the thermal effect [8]. As a result, an efficient group key generation and management for implantable sensors is of great significance [3,9].

In this paper, we propose an efficient cooperative sensor association and group key management protocol with the Chinese remainder theorem (CRT) in wireless body area networks. Our nontrivial efforts can be summarized as follows:

1. A novel WBAN model with message broadcasting: In practical medical WBAN scenarios, patients who receive services from HC are allocated to different departments according to their physical conditions and diseases. As a result, it is necessary for HC to provide a notification service to different patient groups. To the best of our knowledge, we are the first to propose the system model providing a specific group communication channel for message broadcasting between HC and patients. Moreover, the medical data transmission channel from sensors to PC is also taken into consideration in our design.
2. Group key management between HC and PC with CRT: The Chinese remainder theorem is employed for the group key management between HC and PC, which also supports batch key updating. In this case, HC is capable of broadcasting messages to different patient groups. Moreover, patients in the same group are capable of exchanging information about their physical conditions.
3. Group key management between PC and sensors with CCDE: In our design, the group key management between PC and sensors is motivated by coded cooperative data exchange

for the purpose of minimizing the communication rounds for group key generation. Hence, the communication and computation complexity can be drastically reduced, which is efficient for resource-limited wireless sensors in WBAN.

The remainder of this paper is organized as follows. Section 2 briefly surveys the relevant research achievements. Section 3 introduces some necessary preliminary works and the designed system model in order to allow the reader to obtain a better understanding of the topic. Section 4 presents the proposed sensor association and group key management protocol in detail. Section 5 demonstrates the security analysis. Section 6 displays the performance analysis. The conclusion is drawn in Section 7.

2. Related Works

To the best of our knowledge, many research achievements have been made on group key management for wireless body area networks. Theoretically, the traditional public key cryptosystem (TPKC) had been implemented in wireless body area networks previously [10–15]. A certificate generated by a third party is required to combine the identity of the user and the associated public key. However, in TPKC-based schemes, complex modular exponentiation is calculated so that more computation and storage are required in resource-limited wireless sensor devices. Therefore, these TPKC-based group key management schemes cannot meet the practical requirements. In order to alleviate the computation and storage burden on the sensor side, several authentication and group management schemes [4,16–18] based on elliptic curve cryptography (ECC) have been proposed, which provide the same security with a smaller key size compared to TPKC-based schemes.

Many researchers applied the idea of identity-based public key cryptography (ID-PKC) [19], which was a cryptography technique first proposed by Shamir [20] in order to address the certificate management problem in TPKC. In ID-PKC, the public key of the user can be calculated from his/her publicly-known identity, while the secret key of each user is generated by a fully-trusted key generation center (KGC). In 2009, Yang et al. [21] proposed an ID-PKC-based key management scheme for mobile devices. However, Yoon and Chang [22] proved that the proposed scheme was vulnerable to impersonation attacks. Subsequently, several ID-based key agreement protocols were proposed [23–25].

Certificateless public key cryptography (CL-PKC) was first introduced by Al-Riyami and Paterson [26] in 2003. In CL-PKC, the private key of the user consists of two parts, which are respectively generated by a semi-trusted key generation center (KGC) and by the user himself/herself. Hence, the key escrow problem, as well as the certificate management problem can be addressed. Liu et al. [2] proposed two certificateless authentication protocol in the WBAN environment. However, Xiong [27] demonstrated that Liu et al.'s protocols could not provide forward security and scalability. Additionally, a new certificateless encryption scheme and the signature scheme with efficient revocation against short-term key exposure were proposed in [28]. Thereafter, He et al. [3] proposed an efficient certificateless public auditing (CLPA) scheme with the purpose of addressing integrity issues in cloud-assisted WBANs.

Furthermore, the Chinese remainder theorem (CRT) has been applied in many existing group key distribution schemes [29–32]. Zheng et al. proposed two centralized group key management protocols based on CRT [29]. The main contribution of this work is that the transmission passes for group key distribution are minimized, which is available in wireless networks with sourced restriction. After that, Zhou et al. proposed a key tree and CRT-based group key distribution scheme [30]. Note that in this scheme, the key server uses the root keys of the group member subtrees and CRT for group key distribution. Moreover, the computation on the user side is minimized. Based on this, Vijayakumar et al. proposed CRT-based centralized group key management for secure multicast communication [33]. The proposed key management scheme could prominently reduce the computation complexity of the key server.

Coded cooperative data exchange (CCDE) as first introduced by Rouayeb et al. [34] in 2010 and has drawn increasing attention [35–37]. Milosavljevic et al. proposed a deterministic algorithm for CCDE [38], where a novel divide and conquer-based architecture was presented in order to determine the number of bits each node should transmit in the public channel. Subsequently, Sprinston et al. [39] presented a randomized algorithm with a high probability to minimize the number of transmissions over the public channel. In 2016, Courtade et al. characterized the minimum number of public transmissions for key agreement [40] with an arbitrary key distribution.

The aforementioned group key management schemes vary greatly with different security techniques. The existing research emphasizes the secure data transmission between sensors and PC, while the communication and access control for patients remain to be enhanced. In this paper, we design an integral system model involving both HC-PC and PC-sensor communication. In practical scenarios, a high turnover of patients brings frequent key updating in the hospital environment. In this case, we adopt the CRT to PC group key distribution, which could provide fast and effective key updating. Additionally, the CCDE is adopted in sensor group key distribution. Note that the decentralized cooperative key generation strategy drastically decreases the communication cost, which is suitable for resource-limited WBAN sensors. The corresponding security and performance analysis demonstrates that the proposed protocol could provide adequate security assurance and efficiency.

3. Preliminaries and Model Definitions

This section introduces some necessary preliminaries to facilitate the reader's understanding, including bilinear pairing, the coded cooperative data exchange problem (CCDE) and the Chinese remainder theorem (CRT). Meanwhile, the system model and network assumption are presented.

3.1. Bilinear Pairing

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\mathcal{S}$ be multiplicative cyclic groups of a large prime order $\mathcal{P}$. A map function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathcal{S}$ is a bilinear pairing if it satisfies the three properties below:

1. Bilinearity: For $\forall g_1 \in \mathbb{G}_1, \forall g_2 \in \mathbb{G}_2$ and $\forall a, b \in \mathbb{Z}$, there is $\hat{e}({g_1}^a, {g_2}^b) = \hat{e}(g_1, g_2)^{ab}$.
2. Non-degeneracy: For $\exists g_1 \in \mathbb{G}_1$ and $\exists g_2 \in \mathbb{G}_2$, there is $\hat{e}(g_1, g_2) \neq 1$.
3. Computability: For $\forall g_1 \in \mathbb{G}_1$ and $\forall g_2 \in \mathbb{G}_2$, there exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$.

3.2. Coded Cooperative Data Exchange Problem

A set $X = \{x_1, ..., x_n\}$ of n packets each belonging to a finite alphabet $\mathcal{A}$ needs to be delivered to a set of k clients $C = \{c_1, ..., c_k\}$. Each client $c_i \in C$ initially holds a subset X_i of packets denoted by $X_i \subseteq X$. We denote by $n_i = |X_i|$ the number of packets initially available to client c_i and by $\overline{X_i} = X \backslash X_i$ the set of packets required by c_i. We assume that the clients collectively know all packets in X ($\cup_{c_i \in C} X_i = X$). Each client can communicate to all its peers through an error-free broadcast channel capable of transmitting a single packet in $\mathcal{A}$. The data are transmitted in communication rounds. For example, in round i, one of the clients c_j broadcasts a packet x to all its outgoing neighbors in C. The transmitted information x may be one of the original packets in X_j or some encoding of packets in X_j and the information previously transmitted to c_j [34,37].

The problem is to find a scheme that enables each client $c_i \in C$ to obtain all packets in $\overline{X_i}$ (and thus, in X) while minimizing the total number of broadcasts [35].

3.3. Chinese Remainder Theorem

Let k_1, ..., k_n be positive integers that are relatively prime in pairs. Then, for any given integers a_1, ..., a_n, the system of congruences:

$$X \equiv \{a_i \bmod k_i\}_{i \in [1,n]}$$

has a unique solution modulo $\partial_g = \prod_{i=1}^{n} k_i$. The solution is given by:

$$\mathcal{C} \equiv \sum_{i=1}^{n} \alpha_i \beta_i \gamma_i \bmod k_i,$$

where $\beta_i = \frac{\partial_g}{k_i}$ and $\beta_i \times \gamma_i \equiv 1 \bmod k_i$.

3.4. System Model

As shown in Figure 1, the entire system model consists of three entities: the healthcare center (HC), the personal controller (PC) and the sensors. The description of these three entities is given below.

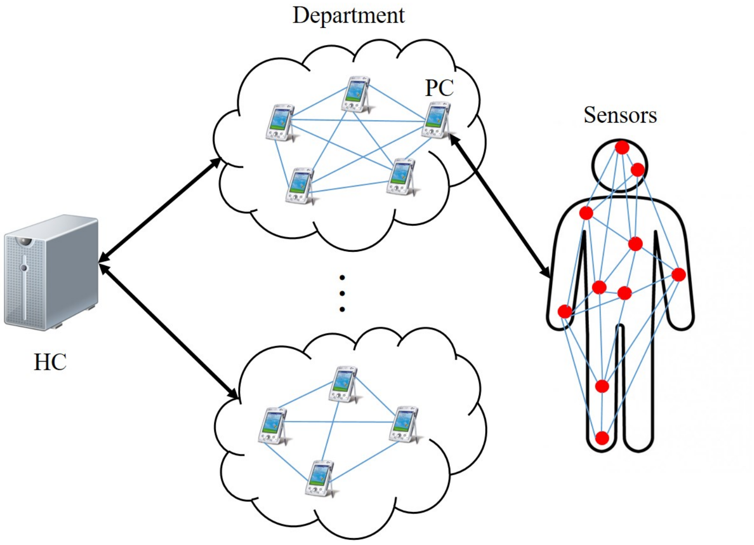

Figure 1. System model.

The healthcare center (HC) is a trustworthy authority providing medical service to the patients. HC is assumed to have adequate storage and computation power. In our system model, HC communicates with PCs to obtain physiology data of patients. Hence, the patient's physical condition can be remotely monitored.

The personal controller (PC) is a mobile device responsible for both biomedical information gathering from sensors and communication with HC. Note that each patient is combined with one PC. The PC employed in this paper is assumed to be professional equipment designed specifically for medical purposes.

Sensors are low-power wireless medical devices either implanted inside or attached to a patient's body. These sensors have limited computation ability and restricted battery capacity. The sensors are responsible for real-time measurement of various physiological parameters of patients.

3.5. Network Assumption

According to Figure 1, there are several departments in the healthcare center. Patients with different diseases are assigned to different departments. In each department, the patients are arranged to be one patient group. HC is assumed to provide service to all the departments (patient groups).

A secure communication channel for data transmission between HC and PC is essential. Furthermore, as mentioned above, a specific group communication channel between HC and a particular patient group is indispensable. As for individual patient, the secure association between PC and multiple sensors is crucial so that the vital physical data from sensors can be safely transmitted.

In our system model, the PC is designed to communicate directly with HC through a wireless channel, which is different from other existing WBAN models using Internet communication between

PC and HC [3,28]. PC is designed as a professional medical device with appropriate treatment units. As part of the medical facility, it is assumed that PC works within the effective range of HC [41,42]. After the patient fully recovers from the disease, his/her PC will be removed and arranged with other new patients.

4. Proposed Schemes

In this section, we explain our cooperative sensor association and group key management protocol, which can be generally divided into two parts: the group key management between HC and PCs affiliated with the same patient group and the cooperative association between sensors and the related PC. According to Figure 1, we assume that HC is in charge of r departments in total. Each department consists of multiple PCs. In this case, one PC is combined with one patient. Consequently, the patient and the relevant PC in this paper are considered as one entity. In department j ($j \in [1, r]$) with n PCs (patients) in total, PC$_i$ ($i \in [1, n]$) is in contact with the corresponding patient P$_i$. As for P$_i$, m sensors are arranged in or on different parts of P$_i$′ body in order to monitor various physiological parameters.

In our design, we are motivated to build a group key management scheme between HC and all the n PCs in department j. At the same time, group key agreement between PC$_i$ and the m sensors is provided accordingly. We introduce our protocol based on department j. Meanwhile, the design for the multiple department situation is similar. The notations used in our protocol are described in the following subsection. Thereafter, the detailed description of our protocol is given, which contains four parts: group key generation for HC and PCs, PC join and leave operation, group key generation for PC and sensors and sensor join and leave operation.

4.1. Notations

The notations used in our protocol and a brief description are listed in Table 1.

Table 1. Notations.

Notation	Description
HC, PC	Healthcare center, personal controller
P$_i$	Patient
hsk, nsk	Symmetric secret key
PSK_i	Secret key of PC$_i$
SSK	HC master key
HID, PID_i	HC and PC$_i$ temporary identity
g, u	Generators of $\mathbb{G}$ and $\mathbb{G}_T$
TS	Time stamp
n	Number of patients in department j
PGK_j	Group key for HC and PCs in department j
$S\_ENC_x(M)$	Symmetric encryption on M with x
$S\_DEC_x(M)$	Symmetric decryption on M with x
$SIG_x(TS\|\|M)$	Signature on M
m	Number of sensors attached to P$_i$
$H()$	One-way hash function
B_v	Master key subset preloaded to SN$_v$
k^i_Ψ	Shared master key
Sk^i_Ψ	Session key
Λ_i	Sensors preloaded with k^i_Ψ
SGK_i	Sensor group key of P$_i$
ρ	Transmission times on the PC$_i$ side
Θ_i	Number of sensors in Λ_i
Φ_i	Number of sensors in $\Lambda_i \cap \Lambda_{i+1}$

4.2. Group Key Generation for HC and PCs

In this section, the group key generation for HC and PCs affiliated with department j is described. It is worth noting that the generation procedures for multiple departments are similar. The proposed group key generation for HC and PCs can be divided into three phases. The first phase is the registration phase, which is responsible for secret key allocation to each PC and other necessary precomputation. The second phase is the group key computation phase, where the group key is generated and distributed to PCs. At last, in the group key derivation phase, each PC derives the group key from the received keying message. The detailed descriptions of these three phases are as follows.

4.2.1. Registration Phase

Before the group key generation procedure, some essential operations should be previously conducted by HC in the registration phase [43]. Initially, let P_1, ..., P_n be n patients who are assigned to department j ($j \in [1, r]$). First, patient $\mathrm{P}_{i\in[1,n]}$ registers to HC so that HC could acquire P_i's personal information including name, age, gender, phone number, and so on. Thereafter, HC allocates PC_i to P_i. Next, HC generates the symmetric key hsk and the secret key PSK_i for $\mathrm{PC}_{i\in[1,n]}$ by conducting **SecKeGen**. Subsequently, HC executes **PreCom** for necessary precomputation. The design of **SecKeGen** and **PreCom** is presented below.

- **SecKeGen**: The HC conducts **SecKeGen** to generate information for $\mathrm{PC}_{i\in[1,n]}$. $\mathbb{Z}_p^*$ and $\mathbb{Z}_s^*$ are defined as two nonnegative integers sets less than p and s, respectively, where p and s are two large prime numbers. Additionally, $\mathbb{G}$ is defined as a multiplicative group of p, and g is a generator of $\mathbb{G}$. HC randomly chooses SSK and $PSK_{i\in[1,n]}$ from $\mathbb{Z}_p^*$, where PSK_i is the secret key of PC_i and SSK is the HC master key. Moreover, HC chooses $hsk \in \mathbb{Z}_s^*$ for symmetric encryption. As a result, the HC temporary identity HID is generated as:

$$HID = g^{SSK||TS}, \tag{1}$$

where TS is the current time stamp.

During the registration phase, HC assigns $\langle PSK_i, HID, hsk\rangle$ to $\mathrm{PC}_{i\in[1,n]}$ of department j and keeps the master key SSK only in its memory. In other words, HC maintains a key list for each department, where SSK, hsk, HID and PSK_i of n PCs are stored. Each PC_i possesses $\langle PSK_i, HID, hsk\rangle$. Note that SSK is the confidential information only known to HC, while HID and hsk are assumed to be known to all PCs in department j.

- **PreCom**: The HC conducts **PreCom** to compute the essential intermediate values [44]. First, HC selects PSK_i from the key list and computes:

$$\partial_g = PSK_1 \times ... \times PSK_n = \prod_{i=1}^{n} PSK_i \tag{2}$$

involving n registered PCs of department j. Then, for each $\mathrm{PC}_{i\in[1,n]}$, HC computes:

$$x_i = PSK_1 \times ...PSK_{i-1} \times PSK_{i+1} \times ...PSK_n = \frac{\partial_g}{PSK_i} \tag{3}$$

and obtains $\{x_1, ..., x_n\}$. That is, x_i for $\mathrm{PC}_{i\in[1,n]}$ is the multiplication of all the remaining PSK_i. Subsequently, HC computes y_i for each x_i ($i \in [1, n]$), which satisfies:

$$x_i \times y_i \bmod PSK_i \equiv 1. \tag{4}$$

That is, y_i is the modular multiplicative inverse of x_i to the modulus PSK_i. Hereafter, HC acquires the variables $var_{i\in[1,n]}$ according to:

$$var_i = x_i \times y_i. \tag{5}$$

Thus, the intermediate value μ can be computed as:

$$\mu = var_1 \times ... \times var_n = \sum_{i=1}^{n} var_i. \tag{6}$$

Upon completion, HC stores the value of μ for the following group key computation. At this point, the precomputation based on CRT is completed.

4.2.2. Group Key Computation Phase

In this phase, the group key of department j is generated by HC. Let q be a large prime number where $q \leq \lceil p/2 \rceil$. First, HC chooses a random value from $\mathbb{Z}_p^*$ as the group key PGK_j. Then, **PGKCom** is conducted by HC in order to obtain the keying message. Finally, HC conducts **SecHtoP** to distribute the keying message to all $\mathrm{PC}_{i \in [1,n]}$. The design of **PGKCom** and **SecHtoP** is described in detail below.

- **PGKCom**: In our design, the HC conducts **PGKCom** to get the keying message γ_j for department j, which is illustrated as:

$$\gamma_j = PGK_j \times \mu. \tag{7}$$

Particularly, for department j, only one PGK_j and one μ are effective in the same time interval. Furthermore, the keying message γ_j is available for all PCs.

- **SecHtoP**: The HC conducts **SecHtoP** to distribute the keying message γ_j to department j. First, HC encrypts the keying message, illustrated as:

$$E(\gamma_j) = S\_ENC_{hsk}(TS||\gamma_j), \tag{8}$$

where $S\_ENC_x(M)$ denotes the symmetric encryption using x. Next, HC computes the certificate $SIG_{SSK}(TS||HID||E(\gamma_j))$ according to:

$$SIG_x(TS||M) = H(M)^{x||TS}. \tag{9}$$

In this way, the certificate can be obtained as:

$$SIG_{SSK}(TS||HID||E(\gamma_j)) = H(HID||E(\gamma_j))^{SSK||TS}. \tag{10}$$

Following the above calculation, the message:

$$\langle TS||HID||E(\gamma_j)||SIG_{SSK}(TS||HID||E(\gamma_j))\rangle$$

is finally broadcast to $\mathrm{PC}_{i \in [1,n]}$ of department j.

4.2.3. Group Key Derivation Phase

In this phase, the main task for PC_i is to verify the validity of the received message by employing **AuthMess**. Subsequently, PC_i derives the group key PGK_j using **GrKeCom**. The design of **AuthMess** and **GrKeCom** is described in detail below.

- **AuthMess**: PC_i conducts **AuthMess** to verify the received message from HC. First, PC_i checks the time stamp TS from the broadcast message. If TS matches the current time, PC_i checks whether:

$$\hat{e}(SIG_{SSK}(TS||HID||E(\gamma_j)), g) \overset{?}{=} \hat{e}(H(HID||E(\gamma_j)), HID)$$

holds. The correctness is elaborated as follows:

$$\begin{aligned} & \hat{e}(SIG_{SSK}(TS||HID||E(\gamma_j)), g) \\ = & \hat{e}(H(HID||E(\gamma_j))^{SSK||TS}, g) \\ = & \hat{e}(H(HID||E(\gamma_j)), g^{SSK||TS}) \\ = & \hat{e}(H(HID||E(\gamma_j)), HID) \end{aligned} . \tag{11}$$

If the certificate is valid, PC$_i$ derives $E(\gamma_j)$ from the message and decrypts the message illustrated as:

$$\begin{aligned} & D(E(\gamma_j)) \\ = & S\_DEC_{hsk}(S\_ENC_{hsk}(TS||\gamma_j)), \\ = & TS||\gamma_j \end{aligned} \tag{12}$$

where $S\_DEC_{hsk}(M)$ denotes symmetric decryption using hsk. At this point, the keying message γ_j is securely transmitted.

- **GrKeCom**: This algorithm is designed for group key derivation from the received keying message γ_j. In **GrKeCom**, a modulo division on the PC$_i$ side is conducted as:

$$PGK_j \equiv \gamma_j \bmod PSK_i, \tag{13}$$

where PSK_i is the allocated secret key. As defined above,

$$\begin{cases} PGK_j < q < PSK_i < p \\ \mu \bmod PSK_i \equiv 1 \end{cases}$$

holds, which guarantees that the derived group key PGK_j is equal to the original one. At this point, the group key generation is finished. All the PC$_i$ of department j share PGK_j with HC.

4.3. PC Join and Leave Operations

In the practical scenario, patients frequently join or leave the department [4,45]. Assume patient P$_i$ of department j is restored to health after receiving the treatment. PC$_i$ is not allowed to obtain the broadcast message after revocation for the purpose of privacy protection towards the remaining patients. Moreover, the newly joined patient needs to be allocated the group key. Consequently, the group key should always be updated when join or leave operations happen.

In this section, the key updating scheme is illustrated respectively from two aspects, namely the PC join operation and the PC leave operation. Note that we demonstrate the join and leave operations in the single-PC case. That is, only one PC is to join or leave the department at the same time. Subsequently, the scenario of multiple PCs joining and leaving the same department is studied in the batch updating operation phase. The detailed description of the join and leave operations, as well as the batch updating operation is as follows.

4.3.1. PC Join Operation Phase

As mentioned above, the PC join operation in department j is considered in this section. It is obvious that the HC should update the group key PGK_j as soon as a specific patient, named P$_{join}$, joins department j. We would like to emphasize that P$_{join}$ should register to HC first, which is in accordance with the actual situations. Then, P$_{join}$ is assigned PC$_{join}$ and obtains its own necessary secret key set $\langle PSK_{join}, HID_{join}, hsk \rangle$ from HC. Subsequently, **JoKeUpdate** is conducted by HC to generate the rekeying message of PC$_{join}$ and other n PCs of department j. Finally, by conducting **JoKeDerive**, the updated group key is distributed to all the $n+1$ PCs of department j. The design of **JoKeUpdate** and **JoKeDerive** is described in detail below.

- **JoKeUpdate**: The HC conducts **JoKeUpdate** to generate the rekeying message for both PC$_{join}$ and the current n PCs. A few steps are necessary as introduced below: First, for PC$_{join}$, HC computes its corresponding x_{join} and y_{join} according to the **PreCom** algorithm in Section 4.2. Hence, the variable var_{join} can be computed as:

$$var_{join} = x_{join} \times y_{join}. \tag{14}$$

In this way, HC computes the intermediate value μ_{join} defined as:

$$\mu_{join} = \mu + var_{join}. \tag{15}$$

Consequently, HC selects a new group key PGK_{j-join} and generates the rekeying message γ_{j-join} by computing:

$$\gamma_{j-join} = PGK_{j-join} \times \mu_{join}. \tag{16}$$

Thereafter, by conducting the **SecHtoP** algorithm introduced in Section 4.2, the rekeying message γ_{j-join} can be securely transmitted to the $n+1$ PCs, which includes one new joining PC$_{join}$ and existing n PCs of department j.

- **JoKeDerive**: This algorithm is designed for the aforementioned $n+1$ PCs to derive the updated group key PGK_{j-join} from γ_{j-join}. After the verification process through **AuthMess** in Section 4.2, the PC$_{i\in[1,n]\cup\{join\}}$ conducts a modulo division, illustrated as:

$$PGK_{j-join} \equiv \gamma_{j-join} \bmod PSK_i. \tag{17}$$

Note that the secret key PSK_{join} of PC$_{join}$ is included in μ_{join} so that the derived new group key PGK_{j-join} is equal to the original one. The process of **JoKeDerive** is similar to the group key derivation phase presented in Section 4.2.

4.3.2. PC Leave Operation Phase

In this section, we assume that the patient P$_{leave}$ is restored to health. Hence, HC deletes this patient and the corresponding PC$_{leave}$ from department j. Moreover, if some PCs in department j were compromised, HC would delete the compromised PC in the same way. In this case, the effective compromised detection strategy is necessary. As for this paper, some existing schemes can be applied in order to detect the compromised PCs periodically [46,47].

In this phase, HC conducts the **LeKeUpdate** algorithm first to generate the rekeying message $\mu_{j-leave}$ and transmits it to the remaining $n-1$ PC$_{i\in[1,n]\setminus\{leave\}}$ securely. Then, **LeKeDerive** is adapted on the PC$_i$ side. Hence, the updated group key $PGK_{j-leave}$ is derived by HC and the rest of the $n-1$ PCs. The design of **LeKeUpdate** and **LeKeDerive** is described in detail below.

- **LeKeUpdate**: The HC conducts **LeKeUpdate** to generate the rekeying message concerning the remaining $n-1$ PCs. A few steps are necessary as introduced below: First, HC obtains μ_{leave} of PC$_{leave}$ demonstrated as:

$$\mu_{leave} = \mu - var_{leave}, \tag{18}$$

where var_{leave} is stored in HC's memory. Consequently, HC selects a new group key $PGK_{j-leave}$ and computes the rekeying message $\gamma_{j-leave}$ according to:

$$\gamma_{j-leave} = PGK_{j-leave} \times \mu_{leave}. \tag{19}$$

Thereafter, by conducting the **SecHtoP** algorithm introduced in Section 4.2, the rekeying message $\gamma_{j-leave}$ can be securely transmitted.

- **LeKeDerive**: After the verification process with the **AuthMess** algorithm in Section 4.2, $\mathrm{PC}_{i\in[1,n]\backslash\{leave\}}$ conducts **LeKeDerive** to derive the updated group key $PGK_{j-leave}$, illustrated as:

$$PGK_{j-leave} \equiv \gamma_{j-leave} \bmod PSK_i. \tag{20}$$

Note that the secret key PSK_{leave} of PC_{leave} is excluded in μ_{leave} so that the removed patient P_{leave} cannot derive the correct group key. The process of **LeKeDerive** is similar to the group key derivation phase presented in Section 4.2.

4.3.3. Batch Updating Phase

With the particular feature of CRT, batch updating for multiple PCs can be achieved accordingly, which meets the practical requirements for medical WBAN. In this section, we present the batch updating involving the join and leave operations of multiple PCs at the same time. Suppose that $\mathrm{P}_{bj\in[1,w]}$ delegate w joining patients in department j. Similarly, $\mathrm{P}_{bl\in[1,z]}$ denote z leaving patients at the same time. P_{bj} and P_{bl} are respectively combined with PC_{bj} and PC_{bl}. Hence, after updating, the number of PCs in department j is $n+w-z$.

In our design, first, HC conducts the **BaKeUpdate** algorithm to generate the batch rekeying message $\gamma_{j-batch}$ and uses **SecHtoP** to distribute it to all the $n+w$ PCs. Afterwards, **AuthMess** is conducted for verification on the PC side. Finally, **BaKeDerive** is conducted so that the updated group key $PGK_{j-batch}$ is obtained by $n+w-z$ PCs in department j. It is noteworthy that the **SecHtoP** and **AuthMess** algorithms are the same as the ones presented in Section 4.2. The design of **BaKeUpdate** and **BaKeDerive** is described in detail below.

- **BaKeUpdate**: The HC conducts **BaKeUpdate** to generate the batch rekeying message for the $n+w-z$ PCs. A few steps are necessary as introduced below: First, with the aforementioned **PreCom** algorithm described in Section 4.2, HC computes the corresponding x_{bj} and y_{bj} of w $\mathrm{PC}_{bj\in[1,w]}$. Hence, the variable for PC_{bj} is obtained as:

$$var_{bj} = x_{bj} \times y_{bj}. \tag{21}$$

Consequently, the sum var_b^+ involving all the w joining PCs can be computed as:

$$\begin{aligned} var_b^+ &= \sum_{bj=1}^{w} var_{bj} \\ &= \sum_{bj=1}^{w} (x_{bj} \times y_{bj}) \end{aligned}. \tag{22}$$

Similarly, the sum var_b^- involving all the z leaving PCs can be computed as:

$$\begin{aligned} var_b^- &= \sum_{bl=1}^{z} var_{bl} \\ &= \sum_{bl=1}^{z} (x_{bl} \times y_{bl}) \end{aligned}. \tag{23}$$

Hence, the intermediate value including w joining PCs and z leaving PCs is defined as follows:

$$\mu_{j-ba} = \mu + var_b^+ - var_b^-. \tag{24}$$

As a result, HC chooses a new group key $PGK_{j-batch}$ and generates the batch rekeying message $\gamma_{j-batch}$, demonstrated as:

$$\gamma_{j-batch} = PGK_{j-batch} \times \mu_{j-ba}. \tag{25}$$

Afterwards, by conducting the **SecHtoP** algorithm introduced in Section 4.2, the batch rekeying message $\gamma_{j-batch}$ can be distributed to all the $n+w$ PCs.

- **BaKeDerive**: After the verification process using the **AuthMess** algorithm in Section 4.2, PC$_{i\in[1,n+w]}$ derives the updated group key $PGK_{j-batch}$ from $\gamma_{j-batch}$ using **BaKeDerive**. The PC$_{i\in[1,n+w-z]}$ conducts a modulo division, illustrated as:

$$PGK_{j-batch} \equiv \gamma_{j-batch} \bmod PSK_i. \tag{26}$$

Note that the w secret keys PSK_{bj} of new join PC$_{bj}$ are included in μ_{j-ba} so that the derived PGK_{j-join} is equal to the original one. Additionally, the secret keys of PC$_{bl\in[1,z]}$ are excluded in μ_{j-ba} so that the removed patient P$_{bl\in[1,z]}$ cannot get the correct group key.

At this point, the batch updating procedure interrelated with w joining patients and z leaving patients is completed. The group key for all the $n+w-z$ PCs in department j is updated securely.

4.4. Group Key Generation for PC and Sensors

In this section, our design is motivated by coded cooperative data exchange (CCDE). Assume that k packages are loaded to t clients previously. In simple terms, the goal of CCDE is to recover the k packages for t clients in minimal transmission. Upon completion, each client obtains all the k packages. So far, many research achievements have been made on solving the CCDE problem. According to [38] and [48], if the t clients are fully connected, the CCDE problem can be solved in polynomial time. Inspired by the group key agreement designed in [5], we consider assigning in total k master keys to all the sensors in department j. The master key distribution follows the rule that every two sensors share at least one master key. Hence, the sensors of department j are fully connected with each other. With the assistance of the corresponding PC, the sensors can build the group key cooperatively. Based on Definition 1 in [5], the CCDE-based scheme is feasible for efficient sensor association for the purpose of achieving optimal transmission passes.

For a better description, we take a patient P$_i$ with PC$_i$, for instance, where P$_i$ is in department j. Let $C_i = \{\mathrm{SN}_v | v \in [1,m], m \in \mathbb{N}^*\}$ be a set of m wireless sensors allocated to P$_i$. The association of these m sensors will be conducted after PC$_i$ successful registers to HC. The proposed sensor association scheme can be divided into two phases: the setup phase and the key generation phase. The setup phase is responsible for secret key allocation and some necessary preparation. Thereafter, the group key is generated in the next key generation phase. The detailed descriptions of these two phases are presented as follows.

4.4.1. Setup Phase

In this phase, PC$_i$ assigns necessary secret information to the m sensors. First, the PC$_i$ conducts **SecKeDis** to generate temporary identity PID_i and symmetric secret key nsk. Thereafter, PC$_i$ conducts **MasKeDis** to distribute the predefined master keys to sensor SN$_{v\in[1,m]}$. The design of **SecKeDis** and **MasKeDis** is described in detail below.

- **SecKeDis**: The PC$_i$ conducts **SecKeDis** to generate nsk and PID_i. Let $\mathbb{Z}_h^*$ be a nonnegative integer set less than h, where h is assumed to be a large prime number. Additionally, $\mathbb{G}_T$ is defined as a multiplicative group of h, and u is the generator of $\mathbb{G}_T$. First, PC$_i$ randomly chooses nsk from $\mathbb{Z}_h^*$. Hence, PID_i is generated, illustrated as follows:

$$PID_i = u^{PSK_i||TS}, \tag{27}$$

where PSK_i is the confidential information of PC$_i$. Thereafter, PC$_i$ stores $\langle PSK_i, PID_i, nsk\rangle$ in its memory.

- **MasKeDis**: The PC$_i$ conducts **MasKeDis** to distribute a set of master keys among the m sensors. Let $Q_i = \{k_h | h \in [1,c], c > m \wedge c \in \mathbb{N}^*\}$ be the c master keys to be allocated. According to our design, a master key subset B_v denoted by $B_v \subseteq Q_i$ is distributed to $\mathrm{SN}_v \in C_i$. Hence, $\forall v_1, v_2 \in \{1, ..., m\}$ $(v_1 \neq v_2)$, $B_{v_1} \cap B_{v_2} \neq \varnothing$ and $B_{v_1} \cup B_{v_2} \subseteq Q_i$ hold. In this way, each sensor $\mathrm{SN}_v \in C_i$ shares at least one master key with each remaining sensor. Upon completion, PC_i assigns $\langle PID_i, nsk, B_v\rangle$ to sensor SN_v.

4.4.2. Key Generation Phase

In this phase, PC_i is responsible for distributing the keying message to all the sensors securely. First, PC_i conducts **MasKeSel**$_1$ to select the most widely-shared master key $k^1_{\Psi} \in Q_i$ in all the m subsets $B_{v\in[1,m]}$ and computes the session key Sk^1_{Ψ}. Afterwards, PC_i transmits the session key Sk^1_{Ψ} to sensors with **SecPtoS**. Subsequently, **AuthSess** is conducted by sensor $\mathrm{SN}_v \in C_i$ so as to guarantee the validity of the received session key and to compare it with B_v. Hence, the sensors preloaded with k^1_{Ψ} are classified as one subset $\Lambda_1 \subseteq C_i$. Other sensors without k^1_{Ψ} abandon the received message.

Thereafter, PC_i runs **MasKeSel**$_2$ to select the second master key k^2_{Ψ}. Similarly, the sensors preloaded with k^2_{Ψ} are classified as the second subset $\Lambda_2 \subseteq C_i$. According to our design, $\Lambda_1 \cap \Lambda_2 \neq \varnothing$. In other words, at least one sensor is preloaded with both k^1_{Ψ} and k^2_{Ψ}. Let $\mathrm{SN}^{\Lambda_1\cap\Lambda_2}_{\hbar}$ be the sensors such that $\mathrm{SN}^{\Lambda_1\cap\Lambda_2}_{\hbar} \in \Lambda_1 \cap \Lambda_2(\hbar \in [1, \Phi_1])$, assuming that there are in total Φ_1 elements in $\Lambda_1 \cap \Lambda_2$. Subsequently, $\mathrm{SN}^{\Lambda_1\cap\Lambda_2}_{\hbar\in[1,\Phi_1]}$ with both k^1_{Ψ} and k^2_{Ψ} conducts **GrKeEnc** so that the sensors in $\complement_{\Lambda_2}(\Lambda_1 \cap \Lambda_2)$ can derive the session key Sk^1_{Ψ}. Note that Sk^1_{Ψ} is considered as the group key SGK_i.

Now, Sk^1_{Ψ} is distributed to the sensors in $\Lambda_1 \cap \Lambda_2$. Subsequently, PC_i repeatedly conducts the above process in order to distribute Sk^1_{Ψ} to the remaining sensors in $\complement_{C_i}(\Lambda_1 \cup \Lambda_2)$. In this way, after several broadcast transmission passes, all the $\mathrm{SN}_v \in C_i$ can finally get Sk^1_{Ψ} as the group key. Hence, the key generation phase is completed. The design of **MasKeSel**$_1$, **SecPtoS**, **AuthSess**, **MasKeSel**$_2$ and **GrKeEnc** is respectively described in detail below.

- **MasKeSel**$_1$: This algorithm is designed for PC_i to select the master key k^1_{Ψ}. It is assumed that PC_i primarily chooses the master key involving more sensors. As a result, the corresponding session key Sk^1_{Ψ} is generated, illustrated as:

$$Sk^1_{\Psi} = H(k^1_{\Psi}||TS). \tag{28}$$

- **SecPtoS**: After the computation of session key Sk^1_{Ψ}, PC_i conducts **SecPtoS** for session key distribution. First, Sk^1_{Ψ} is encrypted by PC_i following:

$$E_1(Sk^1_{\Psi}) = S\_ENC_{nsk}(TS||Sk^1_{\Psi}). \tag{29}$$

As illustrated before, $S\_ENC_x(M)$ denotes the symmetric encryption. Next, PC_i computes the certificate $SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_{\Psi})$ according to Equation (9). Hence, the certificate $SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_{\Psi}))$ can be obtained by computing:

$$SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_{\Psi})) = H(PID_i||E_1(Sk^1_{\Psi}))^{PSK_i||TS}. \tag{30}$$

After the above calculation, the message:

$$\left\langle TS||PID_i||E_1(Sk^1_{\Psi})||SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_{\Psi}))\right\rangle$$

is finally broadcast to $\mathrm{SN}_v \in C_i$. It is noteworthy that the entire process of **SecPtoS** is similar to the aforementioned **SecHtoP**.

- **AuthSess**: This algorithm is designed for sensors to verify the received certificate from PC_i. The whole process is similar to the aforementioned **AuthMess** algorithm. PC_i checks whether:

$$\begin{aligned}&\hat{e}(SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_{\Psi})), u)\\ &\overset{?}{=} \hat{e}(H(PID_i||E_1(Sk^1_{\Psi})), PID_i)\end{aligned}$$

holds. The correctness is elaborated as follows:

$$\begin{aligned} & \hat{e}(SIG_{PSK_i}(TS||PID_i||E_1(Sk^1_\Psi)), u) \\ = & \hat{e}(H(PID_i||E_1(Sk^1_\Psi))^{PSK_i||TS}, u) \\ = & \hat{e}(H(PID_i||E_1(Sk^1_\Psi)), u^{PSK_i||TS}) \\ = & \hat{e}(H(PID_i||E_1(Sk^1_\Psi)), PID_i) \end{aligned} . \tag{31}$$

If the certificate is valid, SN_v derives $E_1(Sk^1_\Psi)$ from the message and decrypts the message as:

$$\begin{aligned} & D_1(E_1(Sk^1_\Psi)) \\ = & S\_DEC_{nsk}(S\_ENC_{nsk}(TS||Sk^1_\Psi), \\ = & TS||Sk^1_\Psi \end{aligned} \tag{32}$$

where $S\_DEC_{nsk}(M)$ denotes symmetric decryption using nsk. As a result, the keying message Sk^1_Ψ is securely transmitted.

- **MasKeSel$_2$**: This algorithm is designed for PC_i to select the second master key k^2_Ψ. It is required that at least one sensor in Λ_1 stores master key k^2_Ψ in its master key subset. That is, $\exists \mathrm{SN}_\Omega \in \Lambda_1$, $k^2_\Psi \in B_\Omega$ holds. Following this rule, PC_i chooses the master key involving more sensors in $\complement_{C_i}\Lambda_1$. After that, session key Sk^2_Ψ is generated according to:

$$Sk^2_\Psi = H(k^2_\Psi||TS). \tag{33}$$

- **GrKeEnc**: After PC_i broadcasts the session keys two times, sensors $\mathrm{SN}_\hbar^{\Lambda_1\cap\Lambda_2}$ have both Sk^1_Ψ and Sk^2_Ψ. Consequently, $\mathrm{SN}_{\hbar\in[1,\Phi_1]}^{\Lambda_1\cap\Lambda_2}$ encrypts Sk^1_Ψ using Sk^2_Ψ as follows:

$$E_2(Sk^1_\Psi) = S\_ENC_{Sk^2_\Psi}(TS||Sk^1_\Psi). \tag{34}$$

Next, $E_2(Sk^1_\Psi)$ is broadcast by $\mathrm{SN}_{\hbar\in[1,\Phi_1]}^{\Lambda_1\cap\Lambda_2}$. It is noteworthy that the transmission process is similar to the aforementioned **SecPtoS**. At last,

$$\left\langle TS||PID_i||E_2(Sk^1_\Psi)||SIG_{PSK_i}(TS||PID_i||E_2(Sk^1_\Psi))\right\rangle$$

is distributed.

After the message checking process employing **AuthSess**, sensors in $\complement_{\Lambda_2}(\Lambda_1 \cap \Lambda_2)$ derive the session key Sk^1_Ψ. Hence, Sk^1_Ψ is distributed as the group key SGK_i.

The above process repeats until:

$$\bigcup_{\dagger=1}^{\rho} \Lambda_\dagger = C_i \tag{35}$$

holds, where ρ denotes the transmission times on the PC_i side. At this point, the group key generation for PC and sensors is completed.

4.5. Sensor Join and Leave Operations

In this section, the occasions of sensor joining and leaving C_i are considered respectively.

4.5.1. Sensor Join Operation

In our system model, the sensor join operation should be available in order to offer continuous treatment for the current patient. Assume patient P_i is equipped with m wireless sensors in department j. SN_{join} denotes the new sensor to be assigned. It is worth emphasizing that the existing m sensors have already been associated with PC_i through the generated group key SGK_i. In this case, the joining sensor SN_{join} first registers to PC_i and obtains $\langle PID_i, nsk, B_{join}\rangle$. Additionally, B_{join} denotes the master key subset allocated to SN_{join}. For $\forall\bar{v} \in [1, m]$, $B_{\bar{v}} \cap \ B_{join} \neq \varnothing$ and $B_{\bar{v}} \cup \ B_{join} \subseteq Q_i$ hold. After that, PC_i selects the master key k_{Ψ}^{join}. Note that k_{Ψ}^{join} is preloaded to B_{join} and at least one existing sensor in C_i simultaneously. That is, for $k_{\Psi}^{join} \in B_{join}$, $\exists\bar{v} \in [1, m]$, $k_{\Psi}^{join} \in B_{\bar{v}} \cap B_{join}$ holds. The next process for the joining sensor is similar to Section 4.4. As a result, all the $m + 1$ sensors obtain the group key SGK_i. The sensor join operation is completed. Furthermore, the occasion with multiple sensors joining the group is similar to the above single-sensor case.

In conclusion, the above sensor join scheme emphasizes allocating the existing group key SGK_i to the new join sensor. However, in order to enhance the security properties, the existing group key should be updated whenever a new sensor joins C_i, which is supported by the aforementioned group key generation process.

4.5.2. Sensor Leave Operation

According to our system model, the sensors are assigned to each patient by the healthcare center and will not be frequently removed from the patient's body. In most cases, the allocated sensors are combined with the related patient and keep working till the patient leaves the department. However, if the sensor is compromised or disabled, the current group key should be refreshed in timely manner. It is notable that in our design, the sensors are closely attached to or in the patient's body so that the sensors are fully controlled by the patient, where the patient is assumed to be a benign user. Hence, for security consideration, PC_i should assign a new secret message and conduct the group key generation process again.

5. Security Analysis

In this section, we analyze the security properties of the proposed protocol. The security theorems, as well as the corresponding proofs are given below.

5.1. Resistance to Replay Attack

The adversary can conduct a replay attack by reusing the previous messages [49,50]. We analyze the resistance to replay attack in the proposed protocol.

Theorem 1. *During the authentication process in the group key management between HC and PCs, replay attack can be prevented. That is, the reuse of the previous message sent from HC cannot pass the current authentication process on the PC side.*

Proof of Theorem 1. The security of replay attack resistance is formally defined through game $\mathcal{G}_1$. Let $\mathcal{A}_1$ be a probabilistic time adversary. $\mathcal{C}_1$ denotes the challenger, and h and H denote the random oracles. It is worth emphasizing that $\mathcal{C}_1$ has the ability to simulate all the oracles and to output the signing message as a real signer [2,3]. In $\mathcal{G}_1$, it is assumed that $\mathcal{A}_1$ can conduct the following corresponding queries to $\mathcal{C}_1$:

h query:$\mathcal{A}_1$ can query the random oracle h at any time. $\mathcal{C}_1$ simulates this random oracle by maintaining a list L_h of tuple $\{j, \mathrm{PC}_i\}$, where L_h is initialized to be empty. When the oracle is queried with input j, if the query j is already in L_h, $\mathcal{C}_1$ outputs PC_i to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ generates a random PC_i and returns it to $\mathcal{A}_1$. Note that $\{j, \mathrm{PC}_i\}$ is added to L_h.

Extract query: Upon receiving the query from $\mathcal{A}_1$, $\mathcal{C}_1$ executes the **SecKeGen** algorithm to generate relevant secret information $\{TS, PSK_i, SSK, g, hsk\}$. It is notable that TS denotes the current time stamp. After that, $\mathcal{C}_1$ computes HID and $E(\gamma_j)$. Finally, $\{\mathrm{PC}_i, HID, TS, g, E(\gamma_j)\}$ is returned to $\mathcal{A}_1$.

H query: $\mathcal{A}_1$ can query the random oracle H at any time. $\mathcal{C}_1$ simulates this random oracle H by maintaining a list L_H of tuple $\{\mathrm{PC}_i, Y_i\}$, where L_h is initialized to be empty. When the oracle is queried with input PC$_i$, if the PC$_i$ is already in L_h, $\mathcal{C}_1$ outputs PC$_i$ to $\mathcal{A}_1$. Otherwise, $\mathcal{C}_1$ generates a random number Y_i and returns it to $\mathcal{A}_1$. Meanwhile, $\{\mathrm{PC}_i, Y_i\}$ is added to L_h.

SigGen query: $\mathcal{C}_1$ simulates the signature oracle by responding to the signature query of message $E(\gamma_j)$. $\mathcal{C}_1$ executes the **SecHtoP** algorithm to generate the signature $SIG(TS||HID||E(\gamma_j))$ and return it to $\mathcal{A}_1$.

Replay query: Upon receiving the signature from $\mathcal{A}_1$, $\mathcal{C}_1$ simulates the replay operation by conducting the **AuthMess** algorithm to check the validity of the received signature. The received signature is compared with the newly-generated signature after a certain time interval Δt by replaying the process.

As a result, $\mathcal{A}_1$ obtains the signature $SIG(TS||HID||E(\gamma_j))$, where the generated signature is valid and the following equation:

$$\begin{aligned}
&\hat{e}(SIG(TS||HID||E(\gamma_j)), g)\\
&= \hat{e}(H(HID||E(\gamma_j))^{SSK||TS}, g)\\
&= \hat{e}(H(HID||E(\gamma_j)), g^{SSK||TS})\\
&= \hat{e}(Y_i, HID)
\end{aligned} \tag{36}$$

holds. At this point, $\langle TS||HID||E(\gamma_j)||SIG(TS||HID||E(\gamma_j))\rangle$ is obtained by $\mathcal{A}_1$, while the newly-generated signature $SIG(TS||g^{SSK||TS_{\Delta t}}||E(\gamma_j))$ involving the corresponding information satisfies:

$$\begin{aligned}
&\hat{e}(SIG(TS||g^{SSK||TS_{\Delta t}}||E(\gamma_j)), g)\\
&= \hat{e}(H(g^{SSK||TS_{\Delta t}}||E(\gamma_j))^{SSK||TS_{\Delta t}}, g),\\
&= \hat{e}(Y_i', HID')
\end{aligned} \tag{37}$$

where $TS_{\Delta t}$ is the time stamp at time Δt generated by $\mathcal{C}_1$ ($TS_{\Delta t} > TS$). Accordingly, by conducting the replay query, $\mathcal{C}_1$ runs the **AuthMess** algorithm as follows:

$$\begin{aligned}
&\hat{e}(SIG(TS_{\Delta t}||HID||E(\gamma_j)), g)\\
&= \hat{e}(H(g^{SSK||TS}||E(\gamma_j))^{SSK||TS_{\Delta t}}, g).\\
&= \hat{e}(Y_i, HID')
\end{aligned} \tag{38}$$

It is obvious that the reused previous signature can pass the authentication only when $Y_i = Y_i'$ and $HID = HID'$. That is, $TS_{\Delta t} = TS$, which contradicts the aforementioned definition. Hence, the replay attack is not available in the proposed group key management scheme between HC and PCs. □

Theorem 2. *During the authentication process in the group key management between PC and sensors, the replay attack can be prevented. That is, the reuse of the previous message sent from PC cannot pass the current authentication process on the sensor side.*

Proof of Theorem 2. The proof of Theorem 2 is similar to the above proof of Theorem 1. □

5.2. Resistance to Forgery Attack

In this section, we analyze the resistance to the forgery attack of the proposed protocol.

Theorem 3. *The proposed group key management scheme between HC and PCs is existentially unforgeable in the random oracle model.*

Proof of Theorem 3. Similarly, the proof of forgery attack resistance is formally defined through game $\mathcal{G}_2$. Let $\mathcal{A}_2$ be a probabilistic time adversary. $\mathcal{C}_2$ denotes the challenger, and h and H denote the random oracles. It is worth noting that $\mathcal{C}_2$ has the ability to simulate all the oracles and to output the signing message as a real signer. In $\mathcal{G}_2$, it is assumed that $\mathcal{A}_2$ can conduct the following corresponding queries to $\mathcal{C}_2$:

h query: This is the same as the definition in Theorem 1.

Extract query: Upon receiving the query from $\mathcal{A}_2$, $\mathcal{C}_2$ executes the **SecKeGen** algorithm to generate relevant secret information $\{TS, PSK_i, SSK, g, hsk\}$. Note that TS denotes the current time stamp. $\{\mathrm{PC}_i, HID, TS, g\}$ is returned to $\mathcal{A}_2$.

SyEnc query: $\mathcal{C}_2$ maintains a list L_S of tuple $\{\mathrm{PC}_i, \gamma_j, E(\gamma_j)\}$, where L_S is initialized to be empty. When queried by $\mathcal{A}_2$, $\mathcal{C}_2$ generates a random number as γ_j and checks the list L_S. If $\{\mathrm{PC}_i, \gamma_j\}$ is already in L_S, $\mathcal{C}_2$ randomly chooses another value again. Otherwise, $\mathcal{C}_2$ computes $E(\gamma_j)$ with hsk. Finally, $\{\mathrm{PC}_i, \gamma_j, E(\gamma_j)\}$ is returned to $\mathcal{A}_2$ and also added to L_S.

H query: This is the same as the definition in Theorem 1.

SigGen query: This is the same as the definition in Theorem 1.

Replay query: Upon receiving the signature from $\mathcal{A}_2$, $\mathcal{C}_2$ simulates the replay operation by conducting the **AuthMess** algorithm to check the validity of the received signature. The received signature is compared with the newly-generated signature of $E(\gamma'_j)$ ($\gamma'_j \neq \gamma_j$).

Finally, the adversary $\mathcal{A}_2$ obtains the signature $SIG(TS||HID||E(\gamma_j))$, as well as $\{TS, HID, E(\gamma_j)\}$ of PC_i by querying $\mathcal{C}_2$. As a result, the equation $\hat{e}(SIG(TS||HID||E(\gamma_j)), g) = \hat{e}(Y_i, HID)$ holds. Furthermore, $\mathcal{C}_2$ outputs another signature $SIG(TS||HID||E(\gamma'_j))$ to $\mathcal{A}_2$. Assume the signature can pass the authentication, illustrated as:

$$\begin{aligned} &\hat{e}(SIG(TS||HID||E(\gamma'_j)), g) \\ &= \hat{e}(H(HID||E(\gamma'_j))^{SSK'||TS}, g) \\ &= \hat{e}(H(HID||E(\gamma'_j)), g^{SSK'||TS}) \\ &= \hat{e}(H(HID||E(\gamma'_j)), HID') \end{aligned}. \tag{39}$$

Thus, the forged signature can pass the authentication only when $Y_i = Y'_i$ and $HID = HID'$. That is, $g^{SSK'||TS} = HID'$, so that $SSK = SSK'$, which contradicts the aforementioned assumption. Hence, the forgery according to the acquired message is not available in the proposed group key management scheme between HC and PCs. □

Theorem 4. *The proposed group key management scheme between PC and sensors is existentially unforgeable in the random oracle model.*

Proof of Theorem 4. The proof of Theorem 4 is similar to the above proof of Theorem 3. □

5.3. Forward Security

In this section, we analyze the forward security property of the proposed protocol.

Theorem 5. *The proposed group key management scheme between HC and PCs provides forward security against an adversary. That is, the revoked PCs (patients) cannot get access to the current communication.*

Proof of Theorem 5. This theorem is analyzed through game $\mathcal{G}_3$. Let $\mathcal{A}_3$ be the adversary by colluding with the revoked PC_i in department j. It is worth noting that $\mathcal{A}_3$ obtains all the secret information stored in PC_{leave} and wants to derive the current group key $PGK_{j-leave}$. After receiving the keying message $\gamma_{j-leave} = PGK_{j-leave} \times \mu_{leave}$ from HC, $\mathcal{A}_3$ conducts the modulo division to derive the group key. However, as described in the aforementioned sections, for the revoked PC_{leave}, HC subtracts var_{leave} from μ so that the $\mu_{leave} = \mu - var_{leave}$. In this case, the rekeying message only involves information of the rest of the $n-1$ PCs. Hence, the revoked PC_{leave} cannot derive the correct group key. That is, $PGK_{j-leave} \neq \gamma_{j-leave} \bmod PSK_{leave}$. Thereafter, the rest of the $n-1$ PCs in department j can update their new group key securely. We assume that the size of PSK_{leave} is ϖ bits. As a result,

$\mathcal{A}_3$ has to perform 2^{ϖ} times in order to obtain one PSK_i of the rest of the $n-1$ PCs. Accordingly, the probability that $\mathcal{A}_3$ can successfully obtain $PGK_{j-leave}$ is $\frac{n-1}{2^{\varpi}}$. Thus, the forward security is provided in our protocol between HC and PCs. □

Theorem 6. *The proposed group key management scheme between PC and sensors provides forward security against an adversary. That is, the revoked sensors cannot get access to the current communication.*

Proof of Theorem 6. As illustrated above, the sensors are closely attached on or in the patient's body and are fully controlled by the patient. Assume a sensor is removed from the patient's body. In this case, PC$_i$ assigns new secret messages including PID_i, nsk and a master key subset to the remaining sensor. The whole group key generation process will be conducted again to refresh the group key. In this way, the revoked sensor cannot derive the new group key since the vital secret information is different. □

5.4. Resistance to Collusion Attack

In this section, we analyze the collusion attack resistance of the proposed protocol.

Theorem 7. *The proposed group key management scheme between PC and sensors provides forward security against an adversary. That is, the revoked sensors cannot get access to the current communication.*

Proof of Theorem 7. We define the collusion attack through game $\mathcal{G}_4$. Let $\mathcal{A}_4^1$ and $\mathcal{A}_4^2$ be the adversaries removed from department j at time t_1 and t_2 ($t_1 < t_2$), respectively. At time t_1, $\mathcal{A}_4^1$ leaves the department with the acquired group key PGK_{t_1-}. Meanwhile, the rekeying message γ_{t_1} is obtained by $\mathcal{A}_4^2$. Additionally, the updated group key PGK_{t_1+} is derived by $\mathcal{A}_4^2$. Similarly, at time t_2, $\mathcal{A}_4^2$ leaves the department with the acquired group key PGK_{t_2-} ($PGK_{t_1+} = PGK_{t_2-}$). The rekeying message γ_{t_2} is obtained by $\mathcal{A}_4^2$. Then, $\mathcal{A}_4^1$ and $\mathcal{A}_4^2$ exchange the obtained information to derive the updated group key PGK_{t_2+}. In this way, $\mathcal{A}_4^1$ and $\mathcal{A}_4^2$ are aware of $\langle PSK_{\mathcal{A}_4^1}, PSK_{\mathcal{A}_4^2}, PGK_{t_1-}, PGK_{t_1+}, \gamma_{t_1}, \gamma_{t_2}\rangle$. With the above information, PGK_{t_2+} is computed according to $PGK_{t_2+} = \gamma_{t_2} \bmod PSK_{\mathcal{A}_4^\eta}$ with $\eta \in \{1,2\}$. Assume the size of $PSK_{\mathcal{A}_4^\eta}$ is ϖ bits. The probability that $\mathcal{A}_4^1$ and $\mathcal{A}_4^2$ can successfully obtain the group key is $\frac{n-2}{2^{\varpi}}$. Hence, the collusion attack is prevented. □

6. Performance Analysis

In this section, we present the performance analysis towards the proposed protocol. As illustrated in the above sections, our scheme consists of two parts: the group key management between HC and PCs and group key management between PC and sensors. The performances of the two schemes are respectively considered. Subsequently, the corresponding simulations and results are presented.

6.1. Group Key Management between HC and PCs

The proposed protocol is compared with two state-of-the-art grouping key management protocols: ESSA [4] and DAKM [44]. The comparison of the computational cost and storage, as well as the communication cost are demonstrated as follows.

6.1.1. Computational Cost and Storage

The computational cost is defined as the total time consumption for group key generation [44]. Additionally, the storage mentioned here refers to the required memory size for the corresponding operations. The comparison result with ESSA and DAKM is given in Table 2. We denote the modulo operation as mod, the exponential operation as Ex and the bilinear pairing as e. Enc and Dec refer to encryption and decryption. Additionally, H, M, D and A represent the one-way hash function, multiplication operation, division operation and addition operation, respectively. Finally, the point multiplication operation is defined as p.

Table 2. Comparison of computational cost and storage.

Protocol	ESSA [4]	DAKM [44]	Our Protocol
Computation of HC	$np + n\text{mod} + 2nA + 2nM + nH$	$3Enc + 2nM + nD + (n-1)A$	$2Ex + 2nM + nD + 1Enc + 1H + (n-1)A$
Computation of PC	$1p + 1\text{mod} + 2A + 2M + 1H$	$1Dec + 1\text{mod} + 1Enc$	$1e + 1H + 1Dec + 1\text{mod}$
Storage of HC	$3n + 10$	$5n + 9$	$3n + 10$
Storage of PC	13	10	8

6.1.2. Communication Cost

The communication cost refers to the time consumption for message transmission. The comparison result on communication cost is given in Table 3. Accordingly, both DAKM and our protocol require one broadcast for the whole process, which is efficient for resource-constrained wireless sensors.

Table 3. Comparison of the communication cost.

Protocol	ESSA [4]	DAKM [44]	Our Protocol
Transmission Type	Unicast	Broadcast	Broadcast
Communication Cost	$3n$	1	1

6.2. Group Key Management between PC and Sensors

In this section, the proposed protocol is analyzed and compared with the ESSA protocol [4]. The comparisons of the computational cost and storage, as well as the communication cost are illustrated as follows.

6.2.1. Computational Cost and Storage

The comparison result with ESSA [4] on the computational cost and storage is given in Table 4. The notations used in the table are the same as those in Table 2. As illustrated above, the sensors in subset $\Lambda_{\mathcal{M}} \subseteq C_i$ get the session key $\mathcal{S}k_{\Psi}^{\mathcal{M}}$. Note that the process repeats for ρ times so that $\mathcal{M} \in [1, \rho]$. For abetter description, we assume that there are $\Theta_{\mathcal{M}}$ sensors in $\Lambda_{\mathcal{M}}$. Meanwhile, there are $\Phi_{\mathcal{M}}$ sensors in subset $\Lambda_{\mathcal{M}} \cap \Lambda_{\mathcal{M}+1}$. In this case, the computational cost on the PC$_i$ side is $(\rho+1)Ex + 2\rho H + \rho Enc$.

Table 4. Comparison of the computational cost and storage.

Protocol	ESSA [4]	Our Protocol
Computation of PC	$(2m + 1)p + 6mH + (m-1)A + Enc$	$(\rho + 1)Ex + 2\rho H + \rho Enc$
Computation of Sensor	$2p + 6H + Dec$	$\left[(e + H + Dec)(\Theta_1 + 2\sum\limits_{i=2}^{\rho}\Theta_i)\right] \Big/ m$
Storage of PC	$6m + 9$	$km + 8$
Storage of Sensor	15	$9 + k$

On the sensor side, we consider the average required computation for message authentication and encryption. The detailed procedures are as follows: First, after receiving the first message from PC$_i$, the computation for each sensor in subset Λ_1 is $1e + 1H + 1Dec$ so that the total computation is $\Theta_1(1e + 1H + 1Dec)$. Similarly, in the second round, after receiving the message from PC$_i$, the computation for all the Θ_2 sensors in subset Λ_2 is $\Theta_2(1e + 1H + 1Dec)$. After that, Φ_1 sensors in $\Lambda_1 \cap \Lambda_2$ broadcast the message to others with computation $1Enc + 1H + 1Ex$. Next, the $\Theta_2 - \Phi_1$ sensors in $\complement_{\Lambda_2}(\Lambda_1 \cap \Lambda_2)$ computes for $1e + 1H + 1Dec$. Hence, we can conclude that the total computation in the i-th rounds is:

$$\begin{array}{l} \Theta_i(1e+1H+1Dec)+\Phi_{i-1}(1Enc+1H+1Ex) \\ \quad +(\Theta_i-\Phi_{i-1})(1e+1H+1Dec) \\ =(2\Theta_i-\Phi_{i-1})e+2\Theta_i H+(2\Theta_i-\Phi_{i-1})Dec \\ \quad +\Phi_{i-1}Enc+\Phi_{i-1}Ex \end{array}. \tag{40}$$

In conclusion, the total computational cost for all the sensors is computed according to:

$$\begin{array}{l} \quad \mathrm{Comp}(i) \\ =(\Theta_1 e+\Theta_1 H+\Theta_1 Dec)+\sum\limits_{i=2}^{\rho}[(2\Theta_i-\Phi_{i-1})e+2\Theta_i H \\ \quad +(2\Theta_i-\Phi_{i-1})Dec+\Phi_{i-1}Enc+\Phi_{i-1}Ex] \\ \approx \Theta_1(e+H+Dec)+2(e+H+Dec)\sum\limits_{i=2}^{\rho}\Theta_i \\ =(e+H+Dec)(\Theta_1+2\sum\limits_{i=2}^{\rho}\Theta_i) \end{array}. \tag{41}$$

Consequently, the average computational cost on the sensor side is:

$$\begin{array}{l} \quad \mathrm{AveComp\_Sen}(i) \\ =\dfrac{\mathrm{Comp}(i)}{m} \\ =\left[(e+H+Dec)(\Theta_1+2\sum\limits_{i=2}^{\rho}\Theta_i)\right]/m \end{array}. \tag{42}$$

We consider the extreme situation where PC_i needs to conduct $m-1$ broadcasting. In this assumption, the computational cost reaches the upper limitation. That is,

$$\begin{cases} \rho=m-1 \\ \Theta_i=2, i\in\{1,...,\rho\} \end{cases}.$$

In this way, the maximum average computation cost on the sensor side is:

$$\begin{array}{l} \quad \mathrm{AveComp\_Sen}(i) \\ =\dfrac{\mathrm{Comp}(i)}{m} \\ =(e+H+Dec)(2+4(m-2))/m \\ =(4-\dfrac{6}{m})(e+H+Dec) \end{array}. \tag{43}$$

According to the practical requirement, $m \gg 6$; thus:

$$\mathrm{AveComp\_Sen}(i) \approx 4(e+H+Dec).$$

Subsequently, the storage comparison with ESSA is shown in Table 4. It is notable that the value k_{PSK^i} in the table denotes a certain storage allocated for the preloaded master keys on the both PC and sensor side.

After this comparison with the existing two protocols on the group key management in WBAN, the simulations for the three protocols are presented, so as to prove the efficiency of the proposed protocol.

Subsequently, the storage comparison with ESSA is shown in Table 4. It is notable that the value k in the table denotes a certain storage allocated for the preloaded master keys on both the PC and sensor side. The comparison result shows that our protocol requires less memory size compared with the ESSA protocol.

6.2.2. Communication Cost

The comparison result on the communication cost is given in Table 5. In ESSA [4], the transmission type during the authentication between PC and sensors is unicast. After that, broadcast is used for group key derivation. PC_i communicates with each sensor for four rounds. Hence, the total communication cost is $4m + 1$. As described above, PC_i broadcasts for ρ times to assign necessary messages to sensors. Moreover, each sensor in subset $\Lambda_{\mathcal{M}} \cap \Lambda_{\mathcal{M}+1}$ ($\mathcal{M} \in [1, \rho - 1]$) broadcasts the keying message to other sensors. In this way, the total communication cost is $\rho + \sum_{i=1}^{\rho-1} \Phi_i$. Similar to the above section, we set $\rho = m - 1$ and $\Phi_i = 2, i \in \{1, ..., \rho - 1\}$ to compute the maximum communication cost. That is,

$$\begin{aligned} &\rho + \sum_{i=1}^{\rho-1} \Phi_i \\ &= m - 1 + 2(m - 2) \\ &= 3m - 5 \end{aligned}. \tag{44}$$

In this case, we can get $3m - 5 < 4m + 1$. It is obvious that our protocol requires less communication cost for group key management between PC and sensors.

Table 5. Comparison of communication cost.

Protocol	**ESSA** [4]	**Our Protocol**
Transmission Type	Unicast/Broadcast	Broadcast
Communication Cost	$4m + 1$	$\rho + \sum_{i=1}^{\rho-1} \Phi_i$

6.3. Simulation Experiments and Results

In the previous two sections, adequate performance analysis and comparison emphasizing computational and communication cost are provided, along with a mathematical discussion and estimation for extreme cases. In addition, relevant simulations are presented in order to prove the efficiency of our protocol. It is worth noting that the time consumption for group key generation and distribution is particularly concerned, which is the crucial factor in the performance evaluation of WBANs.

The experiments were conducted on Windows 10 with a 2.70-GHz Intel(R) Core i7-6820HK CPU and 16 GB memory. Two parts of the proposed protocol, namely the group key management between HC and PCs and group key management between PC and sensors, were performed in Visual Studio 2015 with C++ language. Moreover, the Pairing-Based Cryptography (PBC) library was adopted accordingly.

The experiments on group key management between HC and PCs were conducted first. Note that the assignment of necessary secret information was designed to be done before the formal group key generation. Hence, the time consumption for **SecKeGen** was not included. The simulation was performed for several times based on different numbers of PCs. The comparison results with ESSA [4] and DAKM [44] are presented in Figures 2 and 3. As shown in Figure 2, it is obvious that our protocol required less running time.

When the number of PCs increased, the running time for our protocol and DAKM [44] was similar. Additionally, the running time for each PC was affected by the key size, where in Figure 3, our protocol obviously required less running time on the PC side when the key size was set to 512 bits.

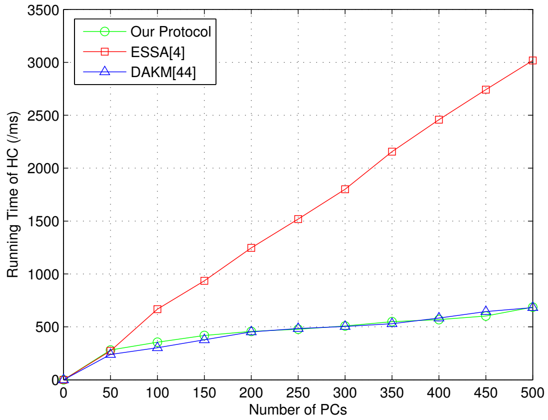

Figure 2. Time consumption of HC for key generation between HC and PCs.

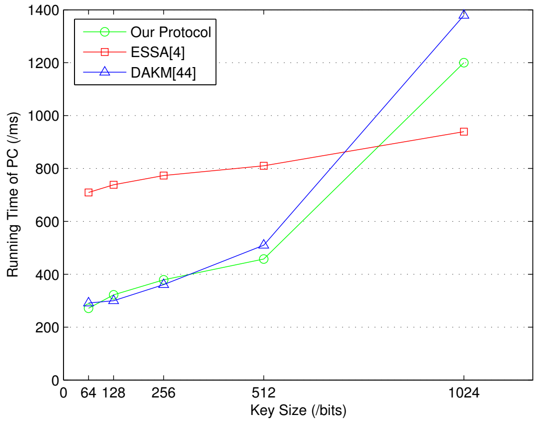

Figure 3. Time consumption of PC for key generation between HC and PCs.

After that, the group key updating time of HC was considered in order to prove the efficiency of our group key updating scheme based on CRT. Note that both the joining and revoked PCs were defined to be the updated PCs. In this way, the key updating time is shown in Figure 4.

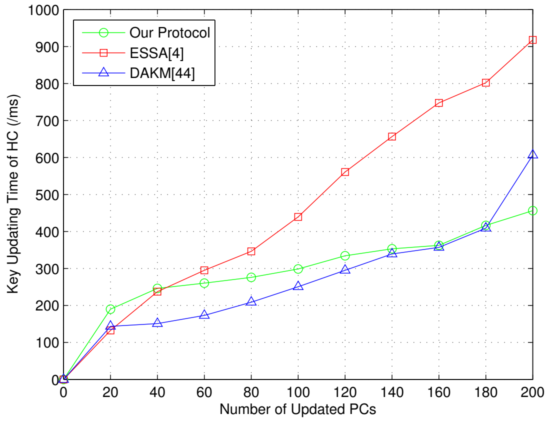

Figure 4. Key updating time of HC.

Similarly, the comparison result with ESSA [4] on group key generation time between PC and sensors is given in Figure 5.

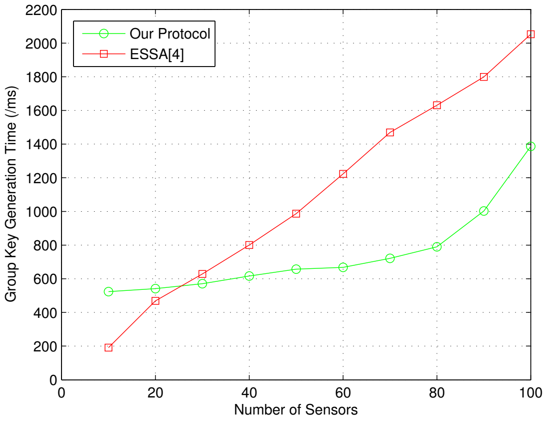

Figure 5. Group key generation time between PC and sensors.

In a word, the above simulation results demonstrate that our protocol could provide better performance than the state-of-the-art group key management protocols.

7. Conclusions

In this paper, first, a novel practical WBAN system model with a notification channel is designed. Moreover, an efficient group key management protocol employing the Chinese remainder theorem (CRT) between HC and PCs is introduced, which supports secure group key updating. In this way, the HC is capable of broadcasting the message to different patient groups. Additionally, the group key scheme between PC and sensors is designed, which is motivated by coded cooperative data exchange (CCDE). Formal security analysis is given, indicating that the proposed protocol can achieve the desired security properties. Furthermore, performance analysis demonstrates that the proposed protocol is efficient compared with the state-of-the-art.

Author Contributions: Conceptualization, H.T. and I.C.; Methodology, H.T.; Formal analysis, H.T.; Investigation, H.T.; Writing—Original Draft Preparation, H.T.; Writing—Review and Editing, H.T. and I.C.; Supervision, I.C.

Funding: This study was supported by the research fund from Chosun University, 2018.

Acknowledgments: The authors would like to thank the anonymous reviewers for their insightful comments and helpful suggestions, which greatly contributed to improving this paper. Specially, Haowen Tan would like to thank Yuanzhao Song for her continuous encouragement and companionship, which means a lot to him.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alemdar, H.; Ersoy, C. Wireless Sensor Networks for Healthcare: A Survey. *Comput. Netw.* **2010**, *54*, 2688–2710. [CrossRef]
2. Liu, J.; Zhang, Z.; Chen, X.; Kwak, K.S. Certificateless Remote Anonymous Authentication Schemes for Wireless Body Area Networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 332–342. [CrossRef]
3. He, D.; Zeadally, S.; Wu, L. Certificateless Public Auditing Scheme for Cloud-Assisted Wireless Body Area Networks. *IEEE Syst. J.* **2018**, *12*, 64–73. [CrossRef]
4. Shen, J.; Tan, H.; Moh, S.; Chung, I.; Liu, Q.; Sun, X. Enhanced Secure Sensor Association and Key Management in Wireless Body Area Networks. *J. Commun. Netw.* **2015**, *17*, 453–462. [CrossRef]
5. Halford, T.R.; Courtade, T.A.; Chugg, K.M.; Li, X.; Thatte, G. Energy-Efficient Group Key Agreement for Wireless Networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 5552–5564. [CrossRef]
6. Zhang, P.; Ma, J. Channel Characteristic Aware Privacy Protection Mechanism in WBAN. *Sensors* **2018**, *18*, 2703. [CrossRef] [PubMed]
7. Lee, D.; Lee, I. Dynamic Group Authentication and Key Exchange Scheme Based on Threshold Secret Sharing for IoT Smart Metering Environments. *Sensors* **2018**, *18*, 3534. [CrossRef] [PubMed]

8. Tan, H.; Choi, D.; Kim, P.; Pan, S.; Chung, I. Secure Certificateless Authentication and Road Message Dissemination Protocol in VANETs. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 7978027. [CrossRef]
9. Augimeri, A.; Fortino, G.; Galzarano, S.; Gravina, R. Collaborative Body Sensor Networks. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 3427–3432.
10. Horn, G.; Preneel, B. Authentication and Payment in Future Mobile Systems. *J. Comput. Secur.* **2000**, *8*, 183–207. [CrossRef]
11. Zhu, J.; Ma, J. A New Authentication Scheme With Anonymity for Wireless Environments. *IEEE Trans. Consum. Electron.* **2004**, *50*, 231–235.
12. Shacham, H.; Brent, W. Compact Proofs of Retrievability. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, 7–11 December 2008; pp. 90–107.
13. Hao, Z.; Zhong, S.; Yu, N. A Privacy-Preserving Remote Data Integrity Checking Protocol With Data Dynamics and Public Verifiability. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1432–1437.
14. Wang, C.; Wang, Q.; Ren, K.; Cao, N.; Lou, W. Toward Secure and Dependable Storage Services in Cloud Computing. *IEEE Trans. Serv. Comput.* **2012**, *5*, 220–232. [CrossRef]
15. Huang, K.; Xian, M.; Fu, S.; Liu, J. Securing The Cloud Storage Audit Service: Defending Against Frame and Collude Attacks of Third Party Auditor. *IET Commun.* **2014**, *8*, 2106–2113. [CrossRef]
16. Lu, R.; Lin, X.; Zhu, H.; Ho, P.; Shen, X. A Novel Anonymous Mutual Authentication Protocol With Provable Link-Layer Location Privacy. *IEEE Trans. Veh. Technol.* **2009**, *58*, 1454–1466.
17. Teranishi, I.; Furukawa, J.; Sako, K. K-Times Anonymous Authentication. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, 5–9 December 2004; pp. 308–322.
18. Tan, H.; Choi, D.; Kim, P.; Pan, S.; Chung, I. Comments on 'Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks'. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2149–2151. [CrossRef]
19. Cao, X.; Zeng, X.; Kou, W.; Hu, L. Identity-Based Anonymous Remote Authentication for Value-Added Services in Mobile Networks. *IEEE Trans. Veh. Technol.* **2009**, *58*, 3508–3517. [CrossRef]
20. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In Proceedings of the Advances in Cryptology, Santa Barbara, CA, USA, 11–15 August 1984; pp. 47–53.
21. Yang, J.; Chang, C. An ID-based Remote Mutual Authentication With Key Agreement Scheme for Mobile Devices on Elliptic Curve Cryptosystem. *Comput. Secur.* **2009**, *28*, 138–143. [CrossRef]
22. Yoon, E.; Yoo, K. Robust ID-Based Remote Mutual Authentication With Key Agreement Scheme for Mobile Devices on ECC. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; pp. 633–640.
23. Wang, H. Identity-Based Distributed Provable Data Possession in Multicloud Storage. *IEEE Trans. Serv. Comput.* **2015**, *8*, 328–340. [CrossRef]
24. He, D.; Chen, J.; Hu, J. An ID-based Client Authentication With Key Agreement Protocol for Mobile Client–Server Environment on ECC With Provable Security. *Inf. Fusion* **2012**, *13*, 223–230.
25. Wang, Y.; Wu, Q.; Qin, B.; Shi, W.; Deng, R.H.; Hu, J. Identity-Based Data Outsourcing With Comprehensive Auditing in Clouds. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 940–952. [CrossRef]
26. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In Proceedings of the Advances in Cryptology-ASIACRYPT2003, Taipei, Taiwan, 30 November–4 December 2003; pp. 452–473.
27. Xiong, H. Cost-Effective Scalable and Anonymous Certificateless Remote Authentication Protocol. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 2327–2339. [CrossRef]
28. Xiong, H.; Qin, Z. Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1442–1455. [CrossRef]
29. Zheng, X.; Huang, C.; Matthews, M. Chinese Remainder Theorem Based Group Key Management. In Proceedings of the 45th Annual Southeast Regional Conference, Winston-Salem, NC, USA, 23–24 March 2007; pp. 266–271.
30. Zhou, J.; Ou, Y. Key Tree and Chinese Remainder Theorem Based Group Key Distribution Scheme. *J. Chin. Inst. Eng.* **2009**, *32*, 967–974. [CrossRef]

31. Lv, X.; Li, H.; Wanga, B. Group Key Agreement for Secure Group Communication in Dynamic Peer Systems. *J. Parallel Distrib. Comput.* **2012**, *72*, 1195–1200. [CrossRef]
32. Guo, C.; Chang, C. An Authenticated Group Key Distribution Protocol Based on The Generalized Chinese Remainder Theorem. *Int. J. Commun. Syst.* **2014**, *27*, 126–134. [CrossRef]
33. Vijayakumar, P.; Bose, S.; Kannan, A. Chinese Remainder Theorem Based Centralised Group Key Management for Secure Multicast Communication. *IET Inf. Secur.* **2014**, *8*, 179–187. [CrossRef]
34. Rouayheb, S.E.; Sprintson, A.; Sadeghi, P. On Coding for Cooperative Data Exchange. In Proceedings of the 2010 IEEE Information Theory Workshop on Information Theory, Cairo, Egypt, 6–8 Januaray 2010; pp. 1–5.
35. Courtade, T.A.; Wesel, R.D. Coded Cooperative Data Exchange in Multihop Networks. *IEEE Trans. Inf. Theory* **2014**, *60*, 1136–1158. [CrossRef]
36. Gonen, M.; Langberg, M. Coded Cooperative Data Exchange Problem for General Topologies. *IEEE Trans. Inf. Theory* **2015**, *61*, 5656–5669. [CrossRef]
37. Heidarzadeh, A.; Yan, M.; Sprintson, A. Cooperative Data Exchange With Priority Classes. In Proceedings of the 2016 IEEE International Symposium on Information Theory, Barcelona, Spain, 10–15 July 2016; pp. 2324–2328.
38. Milosavljevic, N.; Pawar, S.; Rouayheb, S.E.; Gastpar, M.; Ramchandran, K. Deterministic Algorithm for The Cooperative Data Exchange Problem. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 31 July–5 August 2011; pp. 410–414.
39. Sprintson, A.; Sadeghi, P.; Booker, G.; Rouayheb, S.E. A Randomized Algorithm and Performance Bounds for Coded Cooperative Data Exchange. In Proceedings of the 2010 IEEE International Symposium on Information Theory Proceedings, Austin, TX, USA, 13–18 June 2010; pp. 1888–1892.
40. Courtade, T.A.; Halford, T.R. Coded Cooperative Data Exchange for a Secret Key. *IEEE Trans. Inf. Theory* **2016**, *62*, 3785–3795. [CrossRef]
41. Jiang, Q.; Ma, J.; Wei, F.; Tian, Y.; Shen, J.; Yang, Y. An Untraceable Temporal-Credential-Based Two-Factor Authentication Scheme Using ECC for Wireless Sensor Networks. *J. Netw. Comput. Appl.* **2016**, *76*, 37–48. [CrossRef]
42. Pirbhulal, S.; Zhang, H.; Wu, W.; Mukhopadhyay, S.C.; Zhang, Y. Heart-Beats Based Biometric Random Binary Sequences Generation to Secure Wireless Body Sensor Networks. *IEEE Trans. Biomed. Eng.* **2018**, doi:10.1109/TBME.2018.2815155. [CrossRef]
43. Shen, J.; Tan, H.; Zhang, Y.; Sun, X.; Xiang, Y. A New Lightweight RFID Grouping Authentication Protocol for Multiple Tags in Mobile Environment. *Multimed. Tools Appl.* **2017**, *76*, 22761–22783. [CrossRef]
44. Vijayakumar, P.; Azees, M.; Kannan, A.; Deborah, L.J. Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1015–1028. [CrossRef]
45. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S.A. Efficient End-to-End Authentication Protocol for Wearable Health Monitoring Systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195. [CrossRef]
46. Ho, J.; Wright, M.; Das, S.K. ZoneTrust: Fast Zone-Based Node Compromise Detection and Revocation in Wireless Sensor Networks Using Sequential Hypothesis Testing. *IEEE Trans. Dependable Secur. Comput.* **2012**, *9*, 494–511. [CrossRef]
47. Thaile, M.; Ramanaiah, O. Node Compromise Detection based on NodeTrust in Wireless Sensor Networks. In Proceedings of the International Conference on Computer Communication and Informatics, Coimbatore, India, 7–9 January 2016; pp. 1–5.
48. Courtade, T.A.; Wesel, R.D. Weighted Universal Recovery, Practical Secrecy, and An Efficient Algorithm for Solving Both. In Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 28–30 September 2011; pp. 1349–1357.
49. Tan, H.; Choi, D.; Kim, P.; Pan, S.; Chung, I. An Efficient Hash-based RFID Grouping Authentication Protocol Providing Missing Tags Detection. *J. Internet Technol.* **2018**, *19*, 481–488.
50. Pirbhulal, S.; Zhang, H.; Wu, W.; Mukhopadhyay, S.C.; Zhang, Y. An Efficient Biometric-Based Algorithm Using Heart Rate Variability for Securing Body Sensor Networks. *Sensors* **2015**, *15*, 15067–15089. [CrossRef] [PubMed]