Hindawi
Security and Communication Networks
Volume 2018, Article ID 1809302, 16 pages
https://doi.org/10.1155/2018/1809302

Research Article

Energy-Aware Key Exchange for Securing Implantable Medical Devices

Wonsuk Choi, Youngkyung Lee, Duhyeong Lee, Hyoseung Kim, Jin Hyung Park, In Seok Kim, and Dong Hoon Lee

Graduate School of Information Security, Korea University, Seoul, Republic of Korea

Correspondence should be addressed to Dong Hoon Lee; donghlee@korea.ac.kr

Received 11 September 2017; Revised 1 March 2018; Accepted 27 March 2018; Published 14 May 2018

Academic Editor: Vincenzo Conti

Implantable medical devices (IMDs) continuously monitor the condition of a patient and directly apply treatments if considered necessary. Because IMDs are highly effective for patients who frequently visit hospitals (e.g., because of chronic illnesses such as diabetes and heart disease), their use is increasing significantly. However, related security concerns have also come to the fore. It has been demonstrated that IMDs can be hacked—the IMD power can be turned off remotely, and abnormally large doses of drugs can be injected into the body. Thus, IMDs may ultimately threaten a patient's life. In this paper, we propose an energy-aware key exchange protocol for securing IMDs. We utilize synchronous interpulse intervals (IPIs) as the source of a secret key. These IPIs enable IMDs to agree upon a secret key with an external programmer in an authenticated and transparent manner without any key material being exposed either before distribution or during initialization. We demonstrate that it is difficult for adversaries to guess the keys established using our method. In addition, we show that the reduced communication overhead of our method enhances battery life, making the proposed approach more energy-efficient than previous methods.

1. Introduction

Implantable medical devices (IMDs) enable the continuous monitoring of patients with chronic illnesses and automatically deliver therapies when necessary. Recently, advances in medical technology and a convergence with information technology (IT) have led to the development of high-performance IMDs. As a result, millions of people worldwide are now supported by IMDs [1, 2]. Because IMDs are partially or fully inserted into the body of a patient to monitor his/her health, they carry and handle large amounts of personal data. At least once a year, patients with an IMD are supposed to visit their doctors for treatment. The status of the device is checked by the doctor, and its settings are adjusted according to the functionality of the patient's organs.

Only authorized medical staff should be able to adjust an IMD's settings and access the data stored in the IMD related to the health of a patient. However, current IMDs have limited resources for applying security measures, so they have been commercialized and placed on the market without any preventive method against security threats on IMD systems. In fact, the possibilities for a hacker to break into a device to obtain sensitive health-related data and intentionally cause the device to malfunction have been reported over several years [3, 4]. This implies that these devices have the potential to lead to deaths, although they are intended to save lives.

To resolve the security problems of medical devices including IMD systems, relevant policy regulations have been presented. The United States Government Accountability Office (GAO) issued a report in 2012 entitled "Medical Devices: FDA Should Expand Its Consideration of Information Security for Certain Types of Devices" [5]. In this report, the GAO identifies the potential security risks of IMDs and determines how the Food and Drug Administration (FDA) should protect IMDs against information security risks that affect their safety and effectiveness by examining pre- and postmarket activities. The key to reducing the security risks faced by patients using IMDs lies in the authentication technology, because the underlying cause of the IMD vulnerabilities is that external programmers can access the system without any authentication. However, unlike security technologies in other areas, it is difficult

to directly apply security protocols to an IMD, because of its limited resources and constraining requirements. The following four limitations need to be resolved in order to apply effective security to IMD systems: (i) there are high energy overheads of authentication and encryption protocols, (ii) the use of preshared credentials deployed during the manufacturing process remains unchangeable after device implantation, (iii) secure access cannot be deployed during emergency situations, and (iv) protection against resource depletion attacks and denial of IMD functions is insufficient.

This study focuses on the first limitation and suggests a corresponding solution for IMD systems. As IMDs operate on a nonrechargeable battery inside the body, the ultimate depletion of the battery inevitably means that the old IMD needs to be replaced with a new one through surgery. This means that the lifespan of an IMD is mainly determined by the battery's capacity. Therefore, energy-inefficient security mechanisms cannot be applied to an IMD system, even if they would guarantee a high level of security. In this study, we designed a key-exchange method between an IMD and an external programmer, which minimizes the communication overhead of the IMD. The external programmer is the device used by the medical staff to communicate with a patient's IMD. Our basic premise involves utilizing an error correction code (ECC) to adjust the physiological values measured by an IMD and an external programmer. The ECC enables the error correction of redundant data without additional communication between the two entities. Because wireless communication consumes more energy than other processes, such as computation, it is possible to dramatically reduce the total energy consumption of IMDs. This implies that our method not only establishes a secure channel between the IMD and an external programmer but also allows for longer use compared with previous methods [6–15]. In addition, we provide a security analysis by showing that our method satisfies the properties of Secure Sketch, which means that our method is secure against random guessing. The following points summarize the detailed contributions of our method:

(i) Our method minimizes the communication overhead to significantly reduce the IMD's battery consumption.

(ii) We propose a self-recovery method that does not rely on mutual communication between the IMD and an external programmer at the peak misdetection of physiological signals (e.g., ECG or PPG).

2. Related Work

In this section, we introduce related work focusing on security and privacy issues related to IMDs. We classify these studies into several groups presented in the following subsections.

2.1. Alarm-Based Methods. Halperin et al. [3] suggested that an alarm should sound as a warning whenever an attempt is made to access a patient's IMD. Upon hearing the alarm, patients are able to distinguish between valid and invalid attempts. If an invalid attempt by a malicious attacker occurs, the patient takes appropriate action, such as moving from their current location to avoid the attack. However, there are several limitations that prevent this method from being applied to IMD systems. When the patient is in a noisy area, it may be difficult to hear the alarm, and disabled patients could find it difficult to avoid an attack or take appropriate action.

2.2. Distance Bounding. Rasmussen et al. [16] suggested employing proximity-based access control. They assumed that malicious attacks cannot be launched from within a certain distance, as the patient would notice such an attack. Based on this assumption, the IMD authenticates an external programmer by checking that the distance between them is below a certain threshold. The distance is estimated using the relation between the speed of ultrasound and its arrival time. However, their method requires an additional module that enables ultrasonic communication, which results in an additional cost. This module may also incur a significant burden on the battery.

2.3. Communication Cloaking. To reduce battery consumption in IMDs, methods have been suggested that use an external device (called a cloaker) [17–19]. A cloaker is a device, such as a smartphone, that operates on behalf of the IMD. The IMD obtains computational results from the cloaker, thus saving its battery resources. As a result, several cryptographic methods can be applied to IMDs, even though they require heavy computation (i.e., they require significant battery consumption). However, an additional complex security method must be designed to establish a secure channel between an IMD and a cloaker.

2.4. Jamming or Body-Coupled Communication. To achieve security without an additional module or heavy computational burden on the IMD, jamming techniques can be employed [19, 20]. When a hacker attempts to access a patient's IMD, a wireless signal is generated. Accordingly, a jamming technique interrupts the malicious signal to block access to the IMD. However, jamming techniques can affect other valid signals, implying that authorized electronic devices may also not work properly. Although Shen et al. [21] proposed an approach for jamming an attack signal and simultaneously maintaining valid wireless connectivity, channel information must be known in advance to separate the jammed and normal channels. In terms of security, this information should only be shared with authenticated devices. It is impossible to securely share information without using an additional method, such as a secret key exchange. Thus, the jamming technique is not a suitable security method for application to body sensors.

Body-coupled communication [4, 22] is assumed to be secure because an attacker eavesdropping on a communication must be close to or even touching the target's body. However, to apply this method to a body area network (BAN), body sensors require an additional module that enables body-coupled communication.

2.5. Physiological Value/IPI-Based Key Exchange. The concept of physiological value- (PV-) based key exchange (or key agreement) was first introduced by Cherukuri et al. [23]. As PV-based key exchange does not require the exchange of pre-shared secret information between body sensors, it is highly effective from a key management perspective. In particular, this can resolve the problem of emergency access, which is an important requirement in the IMD setting. For this reason, there have been many recent studies conducted in this field [6–15]. Interpulse intervals (IPIs) are the most common metric used in PV-based key exchange. They can be measured noninvasively and easily using low-cost equipment. Studies concerning IPI-based key exchange generally consider three aspects. First, IPIs are derived from different types of heart-related biometric information (e.g., ECG or PPG) [9, 11]. For example, even if an IMD and an external programmer measure different biometric information, the same IPI information can be extracted for key exchange. Second, peak misdetection must be handled effectively [11, 24]. In general, to measure IPIs from heart-related biometric information, a peak detection algorithm is employed. In the real world, all peak detection algorithms have imperfections, which can cause a significant drop in the security performance. If a security method uses an inaccurate peak detection algorithm, then there is a decrease in the security performance compared with the case of using a perfect peak detection algorithm [24]. The third aspect concerns extracting the bit sequence with the highest entropy from one IPI. A 4-bit sequence is extracted from the most common IPI [10, 11, 19]. This implies that at least 32 IPIs are required if a 128-bit key is required. In general, measuring one IPI takes about 0.85 s, and it would take approximately 27.2 s to obtain 32 IPIs. Therefore, to reduce the overall time required for key exchange, high entropy should be retained from one IPI and long bit sequences extracted [25].

3. Motivation

As mentioned in Section 1, security attacks on IMDs have become a critical issue, as researchers have demonstrated that security attacks on commercial IMDs are a reality [3, 4]. In 2008, Halperin et al. [3] described several examples of attacks on a commercial implantable cardioverter defibrillator (ICD). They fully analyzed the communication protocol between a commercial ICD and an external programmer using an oscilloscope and a universal software defined radio (USRP). Because the communication channel they analyzed was not encrypted, they could capture the transmitted data without any difficulty. Using this method, they were able to read and modify the patient's name in the ICD. Moreover, the attackers could even access the patient's ECG data as measured by the ICD. Because this information is related to the patient's health, it should be well protected. Furthermore, because the ICD accepts commands that are used by an external programmer to modify its configuration without any authentication process, the attackers were able to regenerate a certain command using the USRP device. In this manner, the attackers could intentionally deactivate the ICD and induce fibrillation.

Another security problem related to IMDs was demonstrated by Li et al. [4], who were able to attack a popular glucose monitoring and insulin delivery system. The system they targeted used a personal identification number (PIN) for secure access. They explained how to discover PIN information by reverse-engineering the communication protocol and packet format. Moreover, because they could discover the information in a legitimate packet format, it was possible to regenerate a legitimate data packet containing misleading information, which was accepted by the insulin pump, for example, incorrect reading of the glucose level, a control command for stopping/resuming an insulin injection, or a control command for immediately injecting a dose of insulin into the human body. It should be noted that misconfigured insulin therapy may cause hyperglycemia (high blood glucose) or hypoglycemia (low blood glucose) and endanger the patient's life [26].

Fundamentally, the reason for such security problems in the IMD system is that IMDs are not able to authenticate external programmers for secure communication. This lack of authentication makes IMDs vulnerable to a variety of potential attacks, thus compromising their reliable functioning.

4. System Model

In this section, we describe the overall system model for our proposed method. Figure 1 illustrates an example of an IMD system. It is possible to extend the domain to which our method can be applied from IMD systems to body area networks (BANs). As body sensors handle health-related personal information, an appropriate security method for protecting this information is necessary. In particular, because IMDs typically have very limited resources, it is difficult to apply an effective security method. Therefore, in this paper, we focus on the development of a security method that can be applied to IMDs. Moreover, if a security method can be applied to IMDs, this generally implies that the same security method could be used with other body sensors. We propose a method that can perform an authentication protocol and establish a secure channel before the IMD and external programmer communicate with each other. To clarify our proposed method, we describe the IMD system, the main requirements for a security method, the threat model, and our underlying assumptions below.

4.1. IMD System. The IMD system consists of two components: an IMD and an external programmer. Because we are designing a security method for IMDs with resource constraints, only the characteristics of IMDs will be explained in this paper. As IMDs are surgically implanted, wireless communication with an external programmer should be established to access the IMD configuration, especially when the doctor decides to change the therapy delivered by the IMD. Access is also required for diagnosing problems with the equipment, extracting historic information related to the patient's vital signs, or updating the IMD firmware. Traditionally, the IMD and the programmer communicate using inductive telemetry, which is based on inductive

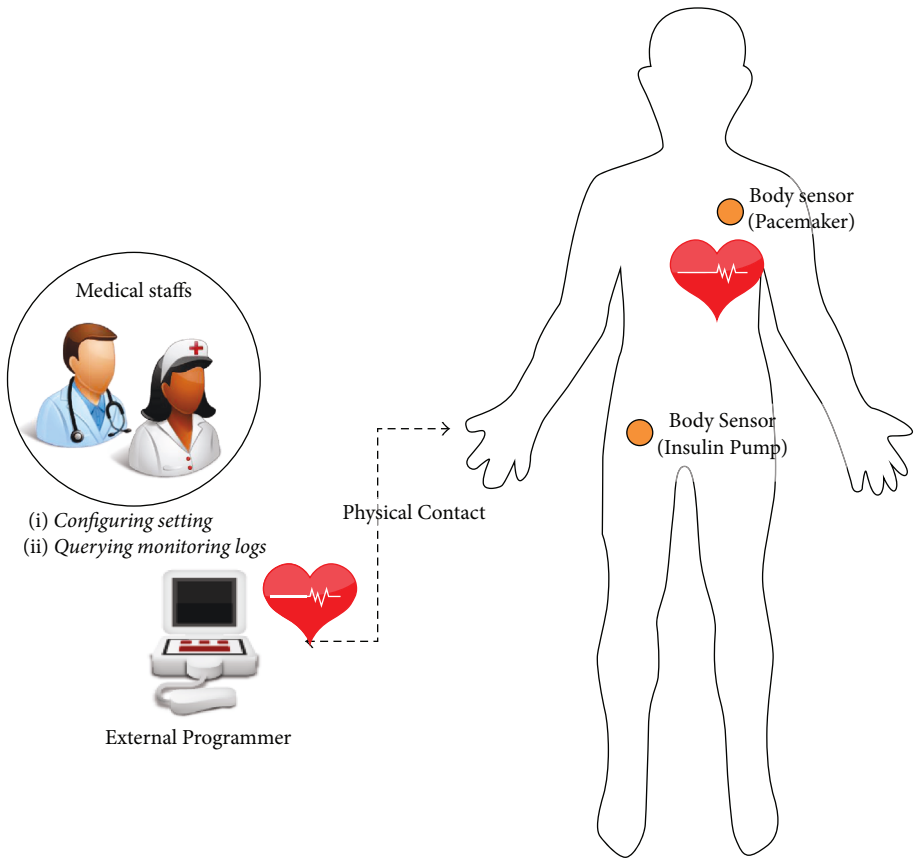

Figure 1: An example of an IMD system.

coupling between coils in the IMD and coils in the programmer. However, this type of communication involves several limitations, including a short communication range and a limited data rate (less than 50 kbps). However, modern IMDs communicate wirelessly with programmers using radio frequency (RF) telemetry through the 402–405 MHz Medical Implant Communication Service (MICS) band, which was established in 1999 by the U.S. Federal Communications Commission. The introduction of MICS has enabled greater communication ranges and higher data rates [27].

4.2. Requirements. In our system model, there are two underlying requirements for the security method to be applied to an IMD system.

4.2.1. Efficient Energy Consumption (Efficiency). Once an IMD is implanted, its battery can last for up to 8 years (in the case of neurostimulators [28]) or up to 10 years (in the case of pacemakers [29]). The exact period is highly dependent on the patient's health (i.e., the more the patient exhibits abnormal physiological conditions over time, the more energy will be consumed by the IMD to react and apply therapy). Three ongoing trends suggest that energy consumption will remain a challenge for IMDs [30]. First, the devices are becoming increasingly complex and power-hungry, because of demands for new and sophisticated therapeutic and monitoring functionalities. Their power requirements are outstripping the benefits of Moore's Law and low-power design techniques that have enabled progress in the area of smartphones. Second, IMDs are collecting more data as new sensors are added to monitor patient health. The transmission of sensor data from an IMD involves wireless communication, which is power-intensive. Third, well-designed security protocols, including authentication and code verification, require the use of cryptography primitives. Even though the overall IMD energy consumption does not stem from a key-exchange protocol, it is known that key-exchange protocols are notoriously computation- and power-intensive. Moreover, the minimal energy consumption in a key-exchange protocol has rarely been considered. Battery usage has a direct impact on the lifetime of an IMD. Once the battery has been depleted, the entire device has to be replaced, which requires a surgical procedure along with the associated risks. Some designs support batteries that can be charged wirelessly using magnetic fields [31–33], but this incurs the risk of damaging the organs close to the IMD [27]. The only realistic alternative is to perform surgery to replace the old battery with a new one. Accordingly, we assume that IMDs use nonrechargeable batteries, meaning that the battery issue is critical when a security method is applied to an IMD system. Therefore, the energy consumption should be minimized.

4.2.2. Emergency Access (Usability). In IMD systems, the balance between usability and security is very important.

Because an IMD is a life-support machine for a patient, its usability has a direct effect on that patient's life. In other words, if the usability of the device is affected by its security features, life-threatening problems can arise. When a security method is applied to an IMD, one typical requirement is emergency access. When a patient loses consciousness, the IMD should be automatically turned off to enable a proper examination of the patient, without errors being introduced by the operating IMD [34, 35]. For wearable devices, this can be achieved by simply removing the device from the patient's body. However, this does not apply to IMDs, implying that emergency access should also be considered when designing a security method. More specifically, if the patient requires an operation in a case of emergency or a scan with magnetic resonance imaging (MRI) when they are fitted with a pacemaker, the pacemaker must be deactivated before the operation in order to prevent unintentional shocks. However, because a hacker may attempt to access the IMD by pretending that there is an emergency, there must be a clear distinction between normal and emergency situations. Therefore, an appropriate security method should define the criteria to distinguish between these situations and perform the appropriate operations.

4.3. Threat Model/Assumption. To clarify the purpose of our method, we first define the threat model that formally identifies the adversaries who may attack an IMD in our system model. The goal of adversaries is to compromise the confidentiality of communications between an IMD and the external programmer. Adversaries' abilities are to eavesdrop on communications, replay old messages, and inject messages. Because our method is a kind of IPI- (or PV-) based key agreement, adversaries may attempt to break the key-exchange process by using PVs from another person or old physiological values from the victim.

We assume that adversaries are unable to obtain the valid PVs to be used as the source of a secret key. Recently, remote photoplethysmography has been suggested, which measures subtle color variations in a human skin surface using a regular RGB camera [36, 37], where heart-related PVs can be inferred. This method could represent one of the most serious threats to PV-based key agreements, including our method. However, it can only be employed to remotely measure such PVs within a short distance (e.g., 50 cm) and thus is not yet a practical threat. We expect that PV-based key agreement will have to be improved as threats that remotely measure PVs of a human body emerge in the future. In addition, denial-of-service (DoS) attacks, such as jamming or battery depletion attacks, are beyond the scope of this paper. Such attacks should be considered separately.

5. The Proposed Method

In this section, we describe our method, which enables efficient key exchanges between an IMD and an external programmer. For ease of understanding, we first explain IPI-based key exchange and ECCs, which are the underlying methods of our approach. We then describe our method in detail.

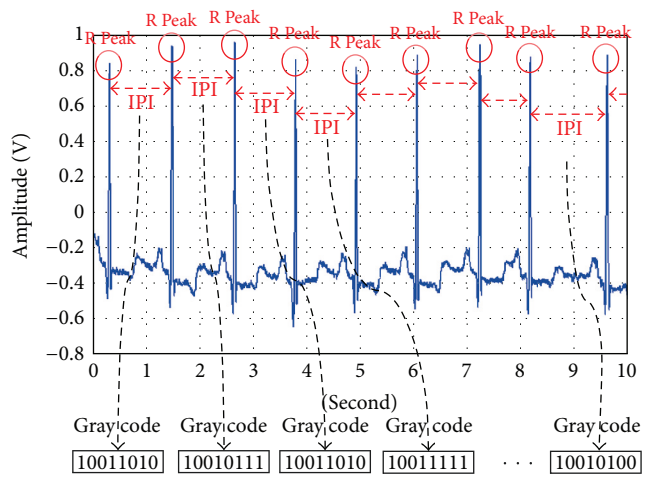

Figure 2: Example of IPIs and their bit strings from ECG signals.

5.1. IPI-Based Key Exchange. There have been many studies concerning the authentication of external programmers by IMDs using IPI-based key exchange, in which measured IPIs are converted to a bit sequence [6–15]. In these methods, IPIs should be simultaneously measured at different parts of a body so that they can be converted to the same bit sequence. These bit sequences are then used as a secret value in a key-exchange method.

An IPI is defined as the elapsed time between two successive pulses (heart rates). The pulse rate changes slightly depending on the condition of the arteries and heart: the rate is around 60–80 beats per minute for an adult and 120–140 beats per minute for an infant. Moreover, the more active the heart is, the faster the blood will be pumped through the arteries, thus leading to a faster pulse rate. Because it is possible to extract randomness from such IPIs, the same random bit sequences can be generated by measuring IPIs at the same time on the same body. Furthermore, two random bit sequences will be different from each other even if they are extracted from two different sets of IPIs that are measured at different times on the same body. Rostami et al. [10] showed that an 8-bit gray-coded sequence from an IPI contains at least 4 bits of entropy. The IPI information is obtained by measuring biosignals of the heart such as ECGs and PPGs. Based on these biosignals, the expansion and contraction intervals of the heart can be measured, thus giving the corresponding IPI values. The expansion and contraction intervals of the heart are calculated from the biosignals using a peak detection algorithm. Figure 2 shows an example of the calculation of IPIs and their conversion to bit sequences using gray encoding, based on ECG measurements.

In the real world, the measurement data includes noise, meaning that the IPI values measured at two different locations may be slightly different. Accordingly, every IPI-based key-exchange method requires a step to make these the same. Figure 3 shows an example of a procedure in IPI-based key-exchange method as a diagram. Step 4 of this figure is the step for the error correction, which usually requires wireless communication between the IMD and the external

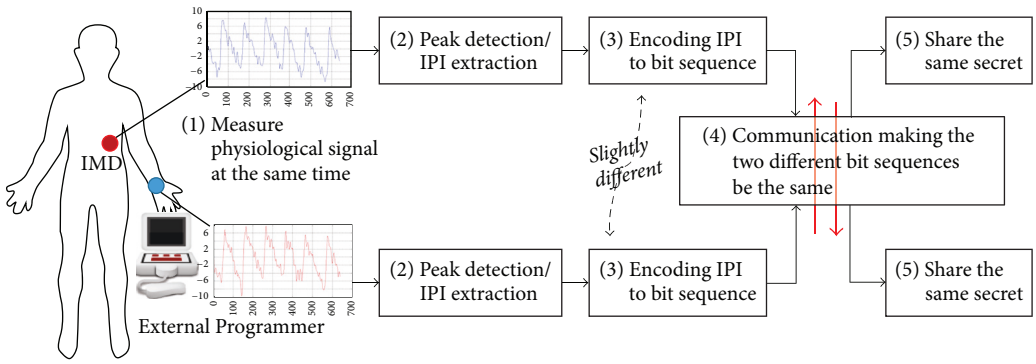

Figure 3: Example of IPI-based key exchange.

programmer. Because wireless communications require a lot of battery power, this step is key saving battery power in an IPI-based key-exchange method.

5.2. Error Correction Code. In general, most communication channels are subject to channel noise, and thus errors can be introduced during transmission from the source to the receiver. To handle such errors, error detection or error correction techniques are often employed. Error detection identifies errors caused by noise or other impairments during transmission from the transmitter to the receiver. Cyclic redundancy checks (CRCs) and checksums are typical examples of error detection techniques. In the case of error correction, more redundant data is added to the original data than in error detection, because error correction aims to reconstruct the original data as well as detect errors. In a simple example known as a repetition code, each data bit is transmitted three times. When the bit sequence 001 is transmitted through a noisy channel, if the third bit contains an error, then the bit sequence 001 is interpreted as being 0.

Formally, ECC is an injective mapping of the form

$$C : \{0,1\}^k \longrightarrow \{0,1\}^n, \tag{1}$$

where $k < n$. Here, $k \in Z^+$ is the message length and $n \in Z^+$ is the block length.

An ECC with Hamming distance $2t + 1$ is denoted by $(n, k, 2t + 1)$. For an ECC $(n, k, 2t + 1)$, the original message should be correctly decoded if no more than t errors occur.

5.3. Our Method. We describe the two steps of the proposed method: (i) the self-recovery of peak misdetection and (ii) the key-exchange protocol. Before describing our method, we list the notations used in our method in the "Notation" section.

5.3.1. Self-Recovery of Peak Misdetection. A peak detection algorithm is used to calculate IPIs from PVs such as ECGs or PPGs. Using a peak detection algorithm, the R peaks of an ECG can be detected, and the time differences between two adjacent R peaks can be calculated to obtain n IPIs. We note that the QRS complex is a name for the combination of three of the graphical deflections seen on a typical ECG. Unfortunately, peak detection algorithms are not 100% accurate, leading to peak misdetections. Although such misdetections degrade the performance of IPI-based key exchange, most previous studies have not attempted to resolve this problem [7, 10, 14, 15]. For such methods, the only available method is to restart the whole procedure for obtaining a set of IPIs whenever a peak misdetection occurs, which drains the battery.

For the first time, Seepers et al. [11, 24] pointed out this inefficiency and proposed a method that tolerates peak misdetection. Their method detects any missed peaks based on a threshold, and the detected results are exchanged via a 1-bit flag. If peak misdetection occurs in one result, then both IPIs are dropped and remeasured.

Unlike the method devised by Seepers et al., we suggest a new approach that can perform a self-recovery procedure when peak misdetection occurs without any communication between the IMD and external programmer. Figure 4 shows two types of peak misdetection, namely, failure of peak detection and fake peak detection, where in the latter the IPIs are misaligned. As peak detection algorithms have reported detection rates of over 99%, we assume that peak misdetection occurs at most once every time our method is performed [38–40].

For a given set of n IPIs, we calculate the sample mean $\overline{x}$ and sample variance s^2 as follows:

$$\begin{aligned} \overline{x} &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{IPI}_i \\ s^2 &= \frac{1}{n-1}\sum_{i=1}^{n} \left(\mathrm{IPI}_i - \overline{x}\right)^2. \end{aligned} \tag{2}$$

To verify IPI_i, we measure the one-dimensional Mahalanobis distance between IPI_i and a distribution of IPIs as follows:

$$d = \sqrt{\left(\mathrm{IPI}_i - \overline{x}\right) s^{-2} \left(\mathrm{IPI}_i - \overline{x}\right)} = \left|\frac{\mathrm{IPI}_i - \overline{x}}{s}\right|. \tag{3}$$

Because we assume that IPIs are normally distributed, 99% of IPIs are separated by less than 2.575 from the standard normal (Z) table. Accordingly, we consider any IPI_i with a Mahalanobis distance larger than 2.575 to be incorrect because of peak misdetection. If the value of $(\mathrm{IPI}_i - \overline{x})/s$ is positive and its distance is larger than 2.575, then it can be

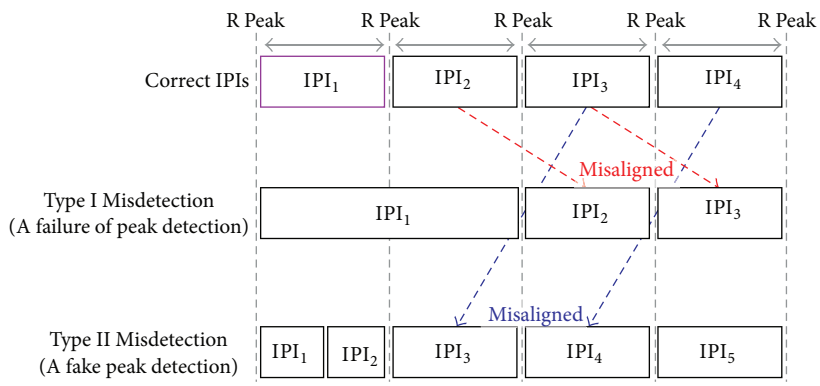

FIGURE 4: Two types of peak misdetection.

considered as a Type I Misdetection. If the value of $(\mathrm{IPI}_i - \overline{x})/s$ is negative and its distance is larger than 2.575, then it is considered as a Type II Misdetection. In case of Type I error, we add a new peak to half of IPI_i. In our method, an IMD and an external programmer need to have the same number of IPI blocks even if their values of IPI blocks are not equal. By this addition, IMD does not have to additionally communicate with an external programmer for peak misdetection. The difference that is caused by the simple addition of a new peak would be corrected by the error correction code. In case of Type II error, we discard the corresponding peak of IPI_i.

Because this method for peak misdetection recovery does not require communication between the IMD and the external programmer, less battery power is consumed. Moreover, because extra IPIs are not measured, the overall key-exchange time for our method is shorter than in the technique devised by Seepers et al.

5.3.2. Key-Exchange Protocol. Because we focus on the energy efficiency of IMDs under a secure key exchange, our method is designed to minimize the communication overhead. We describe the key-exchange procedure between the IMD (I) and external programmer (P) in three steps. The bit sequences from IPIs of I and P are denoted by α and $\beta \in \{0,1\}^n$, respectively.

(1) P sends identifiers ID_I and ID_P to I to initiate the key-exchange procedure. We note that both I and P work on the same body and simultaneously measure IPIs. Given the measured IPIs, the "self-recovery of peak misdetection" procedure is performed.

(2) P randomly chooses $\gamma \in \{0,1\}^k$. Using the $(n, k, 2t+1)$ ECC, $\mathrm{SS}(\beta;\gamma) \in \{0,1\}^n$ is calculated as follows:

$$\mathrm{SS}(\beta;\gamma) = \beta \oplus \mathrm{ECC.encode}(\gamma). \quad (4)$$

A secret key sk is then calculated using the cryptographic hash function $H : \{0,1\}^k \longrightarrow \{0,1\}^l$ as

$$\mathrm{sk} = H(\gamma). \quad (5)$$

Using sk, a message authentication code (MAC) is then calculated for $(1, \mathrm{ID}_P, \mathrm{ID}_I)$, and P transmits $\mathrm{SS}(\beta;\gamma)$ and $\mathrm{MAC}_{\mathrm{sk}}(1, \mathrm{ID}_P, \mathrm{ID}_I)$ to I. We note that the message authentication code is used for key confirmation.

(3) With the $(n, k, 2t+1)$ ECC, I recovers γ' from $\mathrm{SS}(\beta;\gamma)$ as follows:

$$\gamma' = \mathrm{ECC.decode}(\mathrm{SS}(\beta;\gamma) \oplus \alpha). \quad (6)$$

Since ECC.encode() and ECC.decode are inverse to each other, $\mathrm{ECC.decode}(\mathrm{SS}(\beta;\gamma) \oplus \beta)$ is naturally decoded to γ. In addition, the values that are encoded by SS() and have smaller difference than a threshold can be also decoded to γ. Accordingly, $\mathrm{ECC.decode}(\mathrm{SS}(\beta;\gamma) \oplus \alpha)$ can be successfully decoded if α and β are within the threshold.

Using the calculated γ' and $H()$, the secret key sk' can be calculated as

$$\mathrm{sk}' = H(\gamma'). \quad (7)$$

Using sk' for key confirmation, the MAC value transmitted by P is verified as follows:

$$\mathrm{MAC}_{\mathrm{sk}'}(1, \mathrm{ID}_P, \mathrm{ID}_I) \stackrel{?}{=} \mathrm{MAC}_{\mathrm{sk}}(1, \mathrm{ID}_P, \mathrm{ID}_I). \quad (8)$$

Once the verification is complete, the IMD checks whether it is sharing the same key as the programmer and calculates $\mathrm{MAC}_{\mathrm{sk}'}(2, \mathrm{ID}_P, \mathrm{ID}_I)$ to send to P. If this fails, then the session will be aborted.

P also uses sk to verify the MAC values from P for the key confirmation, as follows:

$$\mathrm{MAC}_{\mathrm{sk}'}(2, \mathrm{ID}_P, \mathrm{ID}_I) \stackrel{?}{=} \mathrm{MAC}_{\mathrm{sk}}(2, \mathrm{ID}_P, \mathrm{ID}_I). \quad (9)$$

If the verification fails, then the session will be aborted.

If $\mathrm{dist}(\alpha, \beta) \leq t$, then I obtains $r' = r$. From this, the secret key can be exchanged as in P (i.e., $\mathrm{sk}' = \mathrm{sk}$). Figure 5 illustrates the key-exchange protocol between I and P. We note that a hash operation would

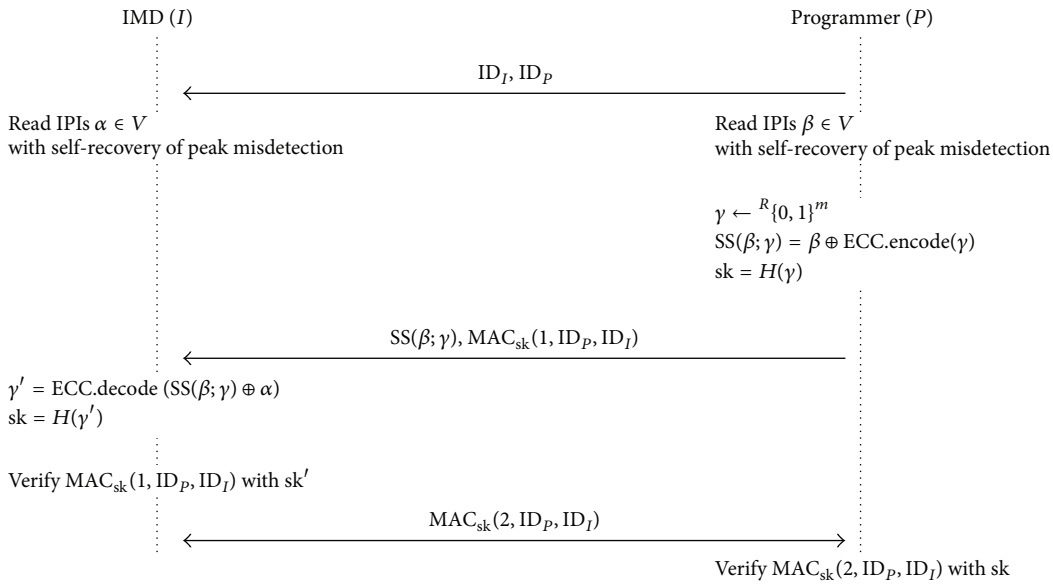

Figure 5: Key-exchange protocol.

be cheaper than the MAC operation in our method in terms of the computational overhead. However, the MAC operation enables explicit key confirmation, which provides stronger assurances for the IMD than implicit key confirmation [41]. The MAC operation in our method can be made optional to reduce computational overhead. In addition, our method is designed as an authenticated key-exchange protocol approach that provides authentication before the key establishment. Because external programmers are authenticated by IMDs in our method, the symmetric key generated by the programmer can be trusted.

6. Security Analysis

To verify the security of our method, we show that it satisfies the requirements of Secure Sketch on the metric space $M = \{0,1\}^n$ under the Hamming distance metric. If a function satisfies the properties of Secure Sketch, we can analyze its security in terms of the entropy. That is, we show that the encoding function of our method satisfies the requirements of Secure Sketch.

Before the detailed analysis, we describe the concept of how our method is securely designed based on the Secure Sketch. It is verified that the random value γ encoded with biometric information (i.e., $\mathrm{SS}(\beta;\gamma)$) leaks only $n-k$ bits about γ, where n is the length of the encoded bit sequence by the Secure Sketch and k is the entropy of the decoded message by error correction code (i.e., a random secret in our case). With respect to a biometric value whose entropy is m, at most $(n-k)$ bits of information are leaked from $\mathrm{SS}(\beta;\gamma)$, and the remaining $m-(n-k)$ bits of information are still preserved. Accordingly, it is said that β is of high entropy when the value of $m-(n-k)$ is larger than the security level. In our method, we determined the security level at 128 bits, to conform with current NIST key length recommendations [42].

6.1. Secure Sketch. The Secure Sketch concept was introduced by Dodis et al. [43, 44] for correcting errors in noisy secrets (e.g., biometrics) by releasing a helper string S that does not reveal any information about the secret. An (M, m, m', t)-Secure Sketch is a randomized map $\mathrm{SS} : M \longrightarrow \{0,1\}^*$ with the following properties, where M is a metric space with distance function $\mathrm{dist}(\cdot)$.

(1) There is a deterministic recovery function $\mathrm{Rec}(\cdot)$ that recovers the original ω from its sketch $\mathrm{SS}(\omega)$, as follows:

$$\mathrm{Rec}\left(\omega', \mathrm{SS}(\omega)\right) = \omega, \quad \text{if } \mathrm{dist}\left(\omega, \omega'\right) \le t, \tag{10}$$

for all $\omega, \omega' \in M$.

(2) For all random variables W over M with min-entropy m, the average min-entropy variable with the min-entropy of W given $\mathrm{SS}(\omega)$ is at least m'. That is, $\widetilde{H}_\infty(W \mid \mathrm{SS}(\omega)) \ge m'$.

The Secure Sketch is efficient if SS and Rec run in polynomial time in the representation size of a point in M. Secure Sketches have been constructed for various different types of metric spaces with defined distance functions. The security of a Secure Sketch is evaluated in terms of the entropy of W when releasing the sketch $\mathrm{SS}(\omega)$, that is, the entropy loss $m - m'$ associated with making $\mathrm{SS}(\omega)$ public. To satisfy the properties of Secure Sketches, most Secure Sketch constructions are designed using the ECC mentioned in previous sections. Furthermore, in this paper, we mainly focus on the second property of Secure Sketches for the security analysis. In other words, we show that the secret key is secure by proving that our method satisfies the second property of Secure Sketches. Regarding the first property, our method does not use the recovery function as it is. In our method, only random numbers are recovered to generate the

Secure Sketches, whereas the conventional recovery function recovers biosignals as well as random numbers.

6.2. Security of the Proposed Method. We show that the function SS(·) in our method, which is based on the $(n, k, 2t+1)$ ECC, satisfies the requirements of the $(M, m, m+k-n, t)$ Secure Sketch. Function SS(·) is expressed as follows:

$$\mathrm{SS}(\beta;\gamma) = \beta \oplus \mathrm{ECC.encode}(\gamma), \tag{11}$$

where $\beta \in W \subset M$ and $\gamma \in_R X \subset \{0,1\}^k$.

To be a Secure Sketch, this function must satisfy the following properties.

(1) For any $\alpha, \beta \in W$ such that $\mathrm{dist}(\alpha,\beta) \leq t$ and $\gamma \in X$, when $\mathrm{SS}(\beta;\gamma)$ and α are given, β needs to be recovered. Here, because recovering β is equivalent to recovering γ, we will show how to recover γ.

(2) $\widetilde{H}_\infty(\omega \mid \mathrm{SS}(\omega)) \geq m+k-n$ should hold.

We can prove that SS(·) satisfies the above two properties, using Lemma 3 of [44], as follows.

Proof. Let ECC.decode(·) be the decoding procedure of ECC.encode(·). As ECC.decode(·) can correct up to t errors, if $v = \beta \oplus \mathrm{ECC.encode}(\gamma)$ and $\mathrm{dist}(\alpha,\beta) \leq t$, then $\mathrm{ECC.decode}(\alpha \oplus v) = \gamma$.

Let A be the joint variable (X, W). These have minimum entropy $m+k$ when $\widetilde{H}_\infty(W) = m$ and X is independent of W. Because $\mathrm{SS}(W) \in \{0,1\}^n$ and $\mathrm{SS}(W)$ is dependent on (X, W), we have that $\widetilde{H}_\infty(A \mid \mathrm{SS}(W)) \geq m+k-n$. If the information exposure of A resulting from $\mathrm{SS}(W)$ is the maximum (i.e., the entropy reduction caused by information exposure is the maximum), then the minimum value of $\widetilde{H}_\infty(A \mid \mathrm{SS}(W))$ is $m+k-n$. In other words, because $\mathrm{SS}(W)$ exposes at most n bits of information, $\widetilde{H}_\infty(A \mid \mathrm{SS}(W))$ cannot be smaller than $m+k-n$. Now, given $\mathrm{SS}(W)$, W and X determine each other uniquely, and so it also holds that $\widetilde{H}_\infty(W \mid \mathrm{SS}(W)) \geq m+k-n$.

As a result, our method provides $(l = m+k-n)$-bit security. □

7. Experimental Results

We evaluated our method by performing a series of experiments. First, we calculated the entropy of the IPIs used for the secret key, to demonstrate the security level of our method. We set the security level to 128 bits, to conform with current NIST key length recommendations [45]. We estimated the parameters that yield 128-bit security and used these to evaluate the performance of our method. Second, we evaluated the energy consumed by our method in terms of computation and communication and compared this with the energy consumption of state-of-the-art secure IMD systems [7, 10, 11].

7.1. Experimental Setup. We extracted IPIs from ECG signals provided by PhysioBank, which is a large archive of well-characterized digital recordings of physiological signals [46]. Among their many data types, we used the MIT-BIH [47], PTB [48], and MGH/MF datasets to ensure a fair comparison between our method and previous methods [10, 11]. In addition, we evaluated our method on the EUROPEAN ST-T and LONG-TERM ST datasets, in order to demonstrate how IPI-based key agreements work on IPIs for patients with heart-related diseases. With the extracted IPIs, we adopted a quantization method that uses the cumulative distribution function (CDF) transformation, known as dynamic quantization. The quantized values were then encoded as 8-bit unsigned integers, and their gray code representations were calculated. The Bose-Chaudhuri-Hocquenghem (BCH) code was used as the ECC in our method.

Table 1: Average entropy.

#lowest bits	Entropy	# lowest bits	Entropy
H(8IPI)	6.51	H(7IPI′)	5.55
H(7IPI)	6.23	H(6IPI′)	5.26
H(6IPI)	5.87	H(5IPI′)	4.73
H(5IPI)	4.95	H(4IPI′)	3.96
H(4IPI)	3.99	H(3IPI′)	2.99

We let $k\mathrm{IPI}'_i$ denote the k lowest bits other than the least significant bit. In other words, if a least significant bit is removed from the $(k+1)$th IPI, then this becomes $k\mathrm{IPI}'_i$.

7.2. Entropy of IPIs. Before evaluating the security level of the secret key derived from our method, we first calculate the entropy of the IPIs, which is the source of the secret key. In comparison to [10, 11], in which three or four least-significant bits were used from an 8-bit gray-coded IPI, our method is designed to acquire the maximum entropy from a single IPI. For most human bodies, the IPI value would be about 0.85 s [7], and so reducing the number of IPIs saves time for the key exchange. We first extracted IPIs from ECG signals in the MIT-BIH, PTB, and MGH/MF datasets, and then gray codes were converted from the IPIs. Using the gray-coded sequences, we performed the MatLab function called Entropy(·) [49] to obtain the entropies of the sequences. In this function, a probability density function (PDF) is estimated from the normalized histogram of a sequence. Using the PDF function, the entropy is calculated as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i), \tag{12}$$

where x_i is a bit sequence. Table 1 shows the average entropy when n least-significant bits are selected from the 8-bit gray-coded bit sequence of a single IPI. We will evaluate the security level of our method based on this result.

7.3. Parameter Estimation for BCH Code. We estimate parameters used in the BCH code. We consider the false rejection (FR) and false acceptance (FA) rates. FR refers to the case in which a valid pair of IPIs that were simultaneously measured from the same human body is incorrectly considered as being invalid. FA refers to the case in which an invalid pair of IPIs that were measured from two different human bodies is incorrectly considered as being valid.

Table 2: Two types of error rates.

err_{same}		err_{diff}	
1st MSB	0.001	1st MSB	0.001
2nd MSB	0.003	2nd MSB	0.268
3rd MSB	0.005	3rd MSB	0.426
4th MSB	0.007	4th MSB	0.425
5th MSB	0.010	5th MSB	0.439
6th MSB	0.020	6th MSB	0.453
7th MSB	0.045	7th MSB	0.444
8th MSB	0.092	8th MSB	0.448

For the FR and FA rates, we utilize the cumulative distribution function (CDF) of the binomial distribution. We define two types of error rates: err_{same} denotes the error rate where values of two bit sequences are different at each bit, even if they are from the same body, and err_{diff} denotes the error rate where values of two bit sequences are the same at each bit even if they are from two different bodies. Table 2 shows the two types of error rates at each most significant bit (MSB) of an 8-bit sequence. These error rates were calculated from the MIT-BIH, PTB, and MGH/MF databases. We note that we obtained different results compared with existing studies [10, 11], even if we employed the same dataset. However, because we used a higher error rate to evaluate our method, the difference does not affect the fair comparison with other methods.

The probability in the binominal distribution is calculated using the mean value of error rates. For example, for 8IPI, the average err_{same} is $(0.001 + 0.003 + 0.005 + 0.007 + 0.010 + 0.020 + 0.045 + 0.092)/8 = 0.023$. When 7IPI is used, the average err_{same} is $(0.001+0.003+0.005+0.007+0.010+0.020+0.045)/7 = 0.013$. In our evaluation, we set the objective FR and FA rates as 10^{-3} and 10^{-30}, respectively. These values are considered to be reasonable in terms of usability and security. Furthermore, the FA rate should be lower than the FR rate, because security should be a more important concern.

A threshold t in BCH code should be calculated that satisfies $P(X_{same} > t) \leq 10^{-3}$ and $P(X_{diff} > t) \leq 10^{-30}$, where X_{same} and X_{diff} are binomial distribution with err_{same} and err_{diff}, respectively. Table 3 lists the minimum values of t that achieve these FR and FA rates for each n value.

Once n and t are determined for the BCH code, the remaining parameter k is determined automatically. For example, if we use $n = 255$ and $t = 19$, then $k = 123$ [50]. Details on the BCH code parameters and their relationships are not discussed in this paper.

Subsequently, the security level was calculated for various code parameters. The security level of our method is equal to $l = m + k - n$, where m is the entropy of the entire IPI. Recall that we analyzed the security level of our method in Section 6.2. For example, when 255-bit sequences need to be derived using 5IPI, 51 IPI blocks are needed. Therefore, the entropy of the bit sequence is $m = 51 \times H(5\text{IPI}) = 51 \times 4.95 = 252.45$. Figure 6 shows the security level $l(= m + k - n)$ for different values of n, k, and t when the FR and FA rate constraints are satisfied. These results show that the security level increases with the length of the bit sequence. The security level of our method is 128 bits when a 255-bit sequence is derived using 6IPI with BCH code parameters (n, k, t) of $(255, 131, 18)$. Although a higher security level can be obtained if a larger value of n is used, $[n/q]$ IPI blocks are needed to derive n-bit sequences ($q = |q\text{IPI}|$).

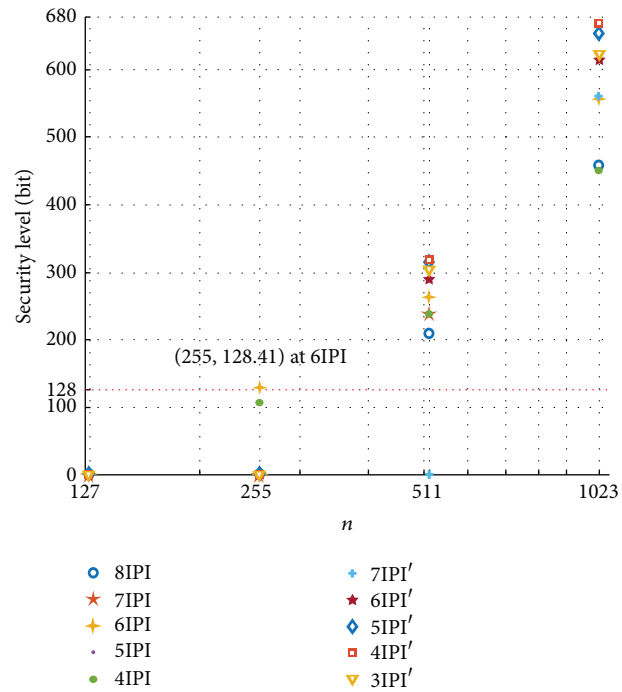

Figure 6: Security level as a function of BCH code parameters (n, k, t).

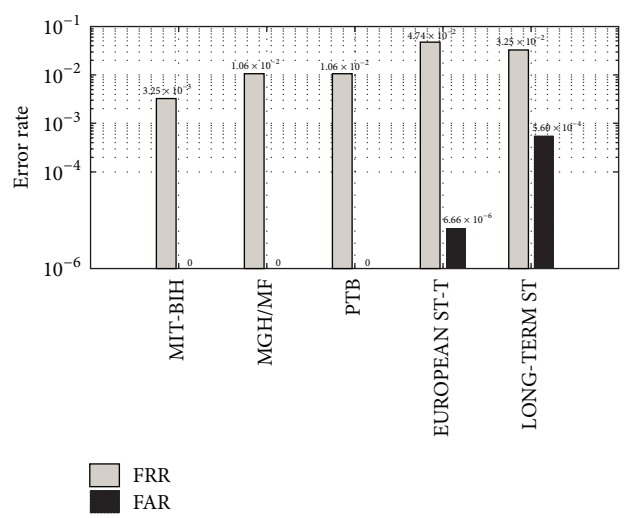

Figure 7: The FR and FA rates with BCH code $(255, 131, 18)$ and 6IPI.

7.4. Performance. We calculated the FR and FA rates for each dataset provided by PhysioBank using the estimated parameters $(n, k, t) = (255, 131, 18)$ and 6IPI. Figure 7 presents the FR and FA rates of five different datasets, including the three

Table 3: BCH code parameter (n, t) for various k.

8IPI		7IPI		6IPI		5IPI		4IPI	
n	t	n	t	n	t	n	t	n	t
127	-	127	-	127	-	127	-	127	-
255	-	255	-	255	18	255	19	255	22
511	23	511	26	511	28	511	32	511	37
1023	39	1023	44	1023	49	1023	55	1023	64
7IPI′		6IPI′		5IPI′		4IPI′		3IPI′	
n	t	n	t	n	t	n	t	n	t
127	-	127	-	127	-	127	-	127	-
255	-	255	-	255	-	255	-	255	-
511	-	511	18	511	19	511	22	511	25
1023	26	1023	29	1023	32	1023	36	1023	42

"-" indicates no (n, t) values available for FN = 10^{-3} and FP = 2.7×10^{-15}.

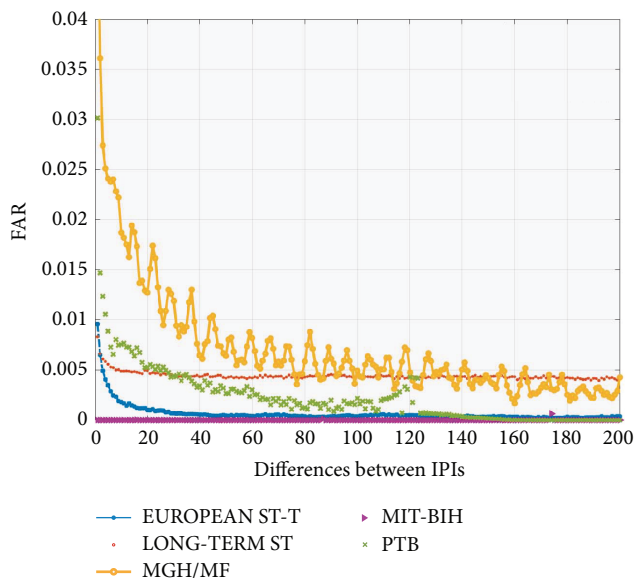

Figure 8: Temporal variance.

mentioned datasets in the previous section. The reason for the high values in Figure 7 (compared with the initial values for the FR rate of 10^{-3} and FA rate of 10^{-30}) is that when we estimated the parameters, there were differences in the average error rate. The higher FA rate for EUROPEAN ST-T and LONG-TERM ST may be because of these datasets being taken from patients diagnosed with heart disease, meaning that the key randomness was relatively lower and the corresponding performance was lower than for the other datasets.

7.5. Temporal Variance. A higher temporal variance implies that an ECG signal has a higher randomness, which reduces the probability of success for attackers employing a replay attack. The bit sequence that is converted from IPIs has sufficient entropy, which means that asynchronous IPIs should not match each other. However, in reality, the probability that asynchronous IPIs match each other is not zero. We examine the temporal variance of our method by employing asynchronous IPIs. If an IMD and an external programmer establish a secret key with the asynchronous IPIs, this can be considered to represent the FA case. Figure 8 illustrates the FA rates with respect to the time differences of IPIs. For example, the FA rate is approximately 0.01 when an IMD and external programmer have a time difference of 3 IPIs from the PTB dataset. We can see that the FAR decreases if the time difference is greater than 130 IPIs, which is around 2 min. We conclude that the IPI information should be protected from attackers for at least 2 min, as attackers can establish a secret key for IMDs within this time.

7.6. Energy Consumption. The effectiveness of IMDs that use nonrechargeable batteries is highly sensitive to the additional energy consumption resulting from new security techniques. We analyzed the energy consumption of our method in terms of communication and computation.

7.6.1. Energy Consumption due to Communication. We used the method proposed in [51] to evaluate the energy consumption of message exchanges. As presented in [52], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 μJ and 59.2 μJ to receive and transmit 1 byte, respectively. Moreover, most of the communication protocol data payloads are set in bytes. For instance, the length of the payload of ZigBee's frame format ranges from 0 to 127 bytes [42]. That is, even when 1 bit of data is transmitted, 1 byte of payload space is required. Therefore, we measured the communication overhead of our method and existing methods in terms of the byte size. To ensure a fair comparison, we measured the communication overhead by modifying each method slightly to enable key confirmation.

For our method, the message sizes to be transmitted and received were 64 and 96 bytes, respectively. Hence, the energy consumption required to transmit and receive the message was 3.79 mJ and 1.83 mJ, respectively. Table 4 lists the communication overheads for our method and the existing methods. The method in [7] was set to 4,000 so that the Coffer size (the number of chaff points) provided 128-bit security. Although 128-bit security could be achieved with a Coffer size

Table 4: Communication overhead.

	Preparation for key exchange	Misdetection tolerance	Secure PV exchange (authentication)	Key confirmation	Total
[7]					
Send	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$	n/a	$\lvert Coffer\rvert + \lvert No\rvert$	$\lvert MAC\rvert$	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert R\rvert + \lvert MAC\rvert = 10{,}080$ bytes
Receive	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$		$\lvert Ind\rvert + \lvert MAC\rvert$	-	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert IND\rvert + \lvert MAC\rvert = 65$ bytes
[10]					
Send	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert Enc_{RSA2014}\rvert$	n/a	$\lvert MAC\rvert$	$\lvert ACK_{Auth}\rvert$	$\lvert Enc_{RSA2014}\rvert + \lvert MAC\rvert + \lvert ACK_{Auth}\rvert = 289$ bytes
Receive	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert Cert\rvert$		$\lvert MAC\rvert$	-	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert Cert\rvert + \lvert MAC\rvert = 1{,}088$ bytes
[11]					
Send	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$	$85 \times \lvert m_{IMD}\rvert$	$\lvert ECC\rvert$	$\lvert MAC\rvert$	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + 85\lvert m_r\rvert + \lvert ECC\rvert + \lvert MAC\rvert = 181$ bytes
Receive	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$	$85 \times \lvert m_P\rvert$	$\lvert Enc_{AES128}\rvert$	-	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + 85\lvert m_s\rvert + \lvert Enc_{AES128}\rvert = 133$ bytes
Ours					
Send	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$	-	-	$\lvert MAC\rvert$	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert MAC\rvert = 64$ bytes
Receive	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert$	-	$\lvert ECC\rvert$	$\lvert MAC\rvert$	$\lvert ID_{IMD}\rvert + \lvert ID_P\rvert + \lvert ECC\rvert + \lvert MAC\rvert = 96$ bytes

(i) References [7, 10] do not support misdetection tolerance. (ii) ID_{IMD} and ID_P denote the identity of an IMD and external programmer, respectively. $\lvert ID_{IMD}\rvert = \lvert ID_P\rvert = 16$ bytes. (iii) Coffer denotes the set that includes both real PVs and chaff points ($\lvert Coffer\rvert = 10{,}000$ bytes). (iv) No denotes a nonce (unique random number) ($\lvert No\rvert = 16$ bytes). (v) Ind denotes the information indicating the index of real measurements out of all elements in Coffer ($\lvert Ind\rvert = 1$ byte). (vi) MAC denotes the message authentication code ($\lvert MAC\rvert = 32$ bytes). (vii) ECC denotes the error correction code ($\lvert ECC\rvert = 32$ bytes). (viii) m_{IMD} and m_P denote the misdetection flag ($\lvert m_{IMD}\rvert = \lvert m_P\rvert = 1$ byte). (ix) Cert denotes the certificate that includes the public key of an external programmer ($\lvert Cert\rvert = 1024$ bytes).

Table 5: Approximate number of cycles and energy consumption of each component block.

Block name	# cycles	Energy consumption (μJ)	Where
SHA256	5,140	4.74	Every method
HMAC_SHA256	32,710	41.73	Every method
BCH-encoding	500,928	517.99	[11]
BCH-decoding	142,704	110.43	Our method
AES128 encryption	2,740	0.1	[11]
RSA2048 encryption	8,558,400	6551.79	[10]

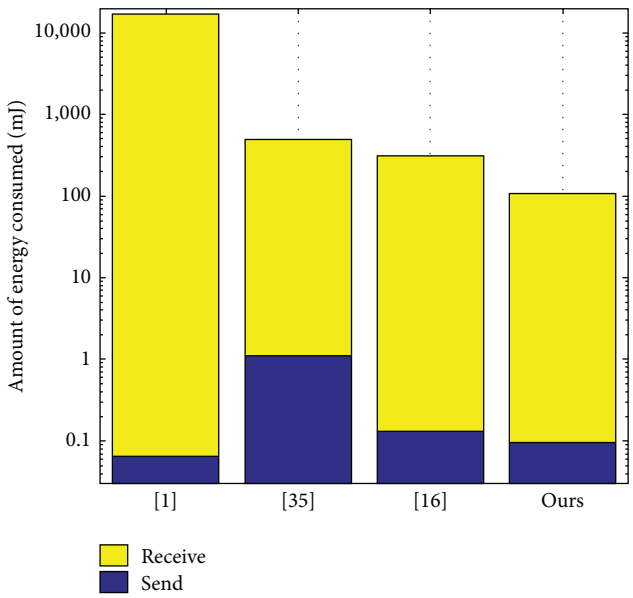

Figure 9: Energy consumption due to communication overhead.

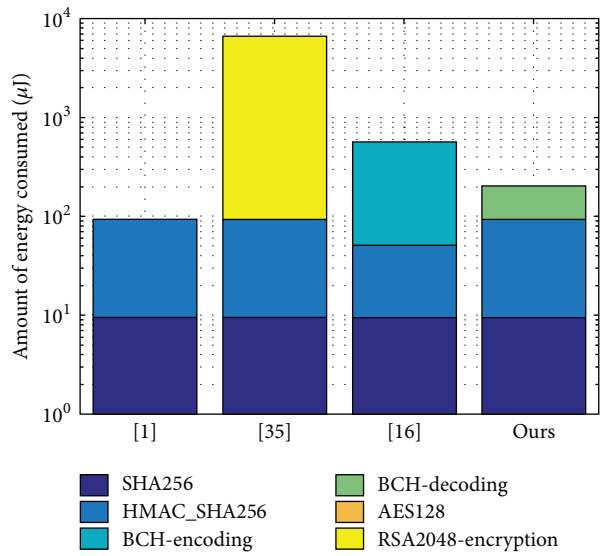

Figure 10: Energy consumption due to computation overhead. It is difficult to identify the amount of energy consumed by SHA256, $HMAC_{SHA256}$, and AES128 because these algorithms consume relatively little energy.

of 2,000, this gave a high FRR. In [11], the authors employ PRESENT, which is an encryption algorithm using an 80-bit symmetric key. However, we evaluate the method in [11] assuming AES128 is employed for a fair comparison with our method, which also employs AES128. Figure 9 shows the energy consumption of each method due to communications under 128-bit security. Because our method has the lowest communication overhead of the techniques compared here, the energy consumption due to communication is also the lowest.

7.6.2. Energy Consumption due to Computation. To measure the energy consumed by computations, we used the EFM32LG-DK3650 Development Kit with a 32-bit ARM Cortex-M3 processor, 256 Kb flash, and 32 Kb SRAM. This development kit provides a power-consumption profiler and power debugging tools. We implemented various block components used by either our method or the comparative techniques in the EFM32LG-DK3650 Development Kit. The code size, number of clock cycles, and approximate power consumption of each block component are listed in Table 5. We note that we assume that the method in [11] employs AES128 instead of PRESENT with an 80-bit symmetric key, for a fair comparison with our method. Because less energy is consumed for BCH encoding than BCH decoding, we designed the IMD to perform BCH decoding. Thus, the energy consumption of our method is lower than that of [11], which performs BCH encoding. Figure 10 shows the energy consumption of our method and existing approaches due to computation.

As public key operations require a considerable amount of energy, the method described in [10] consumes the most energy in the computation stage. For [7], which has the lowest energy consumption for computations, the overall amount of energy consumption is high, because of the communication overhead. In conclusion, our method consumes the least amount of energy when the IMD and external programmer perform the key-exchange protocol.

8. Discussion

Electrogram (EGM) Signal. Our method was evaluated using IPIs that are extracted from an ECG. However, an IMD actually measures electrograms (EGMs), rather than ECG within individual chambers of hearts. On the other hand,

EGM is not measurable outside the body, and hence an external programmer is not able to measure it. Our plans for future work involve additional evaluations using EGM for an IMD and ECG for an external programmer. Because EGM also determines heart-related PVs, like ECG, it is expected that IPIs can be extracted from EGM. In addition, most IMDs have the functionality to record a series of IPIs [53].

Cardiac Arrhythmia. Those who implant IMDs into their body would be patients with heart-related diseases such as arrhythmia. It is known that it is difficult to detect peaks in the ECG signals of such patients [54]. In fact, most PV-based key agreement solutions have the same limitation [7, 34, 51, 52]. Our plans for future work include an improved method to address this issue.

Distribution of IPIs. The normal distribution of IPIs was assumed for self-recovery of peak misdetection. Our method is designed based on the existing research results assuming the normal distribution for IPIs [55, 56]. However, it is expected that the more accurate distribution for IPIs is necessary to improve our self-recovery of peak misdetection. One of our future works will cover analyzing distribution of IPIs.

9. Conclusion

In this paper, we have presented an energy-aware key-exchange protocol that enables secure communication between an IMD and an external programmer. Our method utilizes IPIs to generate a secret key in an authenticated and transparent manner without any keying material being exposed predistribution or during initialization. As the battery consumption of IMDs is a critical issue, we first focused on the energy efficiency of IMDs when using our method. Our method reduces energy consumption, while still enabling secure key exchange. A security analysis showed that our method satisfies the Secure Sketch requirements, meaning that it is difficult for adversaries to guess the secret key. Finally, experiments were conducted to estimate the entropy of IPIs and the parameters for the BCH code. We also analyzed the performance, temporal variance, and key-exchange time. Finally, we demonstrated that our method consumes less energy in communications and computations than comparable techniques. As a result, our method is more feasible and efficient for securing IMD systems than the existing approaches.

Notation

$\overline{x}$: given a sample of size n, consider n independent random variables $X_1, X_2, \ldots, X_n$, each corresponding to one randomly selected observation. The sample mean is defined to be $\overline{x} = (1/n)(x_1, x_2, \ldots, x_n)$, where x_i is the ith observation

$\overline{s^2}$: the sample variance is defined to be $s^2 = (1/n)\sum_{i=1}^{n}(x_i - \overline{x})^2$

α, β: α and $\beta \in \{0,1\}^n$ are denoted by n-bit sequences converted from IPIs measured at the IMD and external programmer, respectively

γ: γ is denoted by a k-bit random sequence $\in \{0,1\}^k$ selected by the external programmer

ECC.encode(·): for $(n, k, 2t+1)$ – ECC, ECC.encode(·) is the encoding mapping

ECC.decode(·): for $(n, k, 2t+1)$ – ECC, ECC.encode(·) is the decoding mapping

$H(\cdot)$: $H(\cdot)$ is denoted by a cryptographic hash function

$\mathrm{MAC}_k(\cdot)$: $\mathrm{MAC}_k(\cdot)$ is denoted by a message authentication code with a secret key k

$\mathrm{ID}_P, \mathrm{ID}_I$: ID_P and ID_I are denoted by identities of the external programmer (P) and IMD (I), respectively.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Samsung Electronics.

References

[1] A. J. Greenspon, J. D. Patel, E. Lau et al., "16-Year trends in the infection burden for pacemakers and implantable cardioverter-defibrillators in the United States: 1993 to 2008," *Journal of the American College of Cardiology*, vol. 58, no. 10, pp. 1001–1006, 2011.

[2] C. Zhan, W. B. Baine, A. Sedrakyan, and C. Steiner, "Cardiac device implantation in the United States from 1997 through 2004: a population-based analysis," *Journal of General Internal Medicine*, vol. 23, no. 1, pp. 13–19, 2008.

[3] D. Halperin, T. S. Heydt-Benjamin, B. Ransford et al., "Pacemakers and implantable cardiac defibrillators: software radio attacks and zero-power defenses," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '08)*, pp. 129–142, Oakland, Calif, USA, May 2008.

[4] C. Li, A. Raghunathan, and N. K. Jha, "Hijacking an insulin pump: security attacks and defenses for a diabetes therapy system," in *Proceedings of the Proceeding of the IEEE 13th International Conference on e-Health Networking, Applications and Services (HEALTHCOM '11)*, pp. 150–156, Columbia, Mo, USA, June 2011.

[5] T. U. S. G. A. Office, Medical devices: citation fda should expand its consideration of information security for certain types of device, 2012.

[6] S.-D. Bao, C. C. Y. Poon, Y.-T. Zhang, and L.-F. Shen, "Using the timing information of heartbeats as an entity identifier to secure body sensor network," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 6, pp. 772–779, 2008.

[7] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in

Proceedings of the 32nd IEEE Conference on Computer Communications, IEEE INFOCOM 2013, pp. 2274–2282, Italy, April 2013.

[8] N. Jamali and L. C. Fourati, "SKEP: a secret key exchange protocol using physiological signals in wireless body area networks," in *Proceedings of the International Conference on Wireless Networks and Mobile Communications, WINCOM 2015*, Marrakech, Morocco, October 2015.

[9] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and M-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[10] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-Heart (H2H): authentication for implanted medical devices," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 1099–1111, ACM, Berlin, Germany, November 2013.

[11] R. M. Seepers, J. H. Weber, Z. Erkin, I. Sourdis, and C. Strydis, "Secure key-exchange protocol for implants using heartbeats," in *Proceedings of the ACM International Conference on Computing Frontiers, CF 2016*, pp. 119–126, Como, Italy, May 2016.

[12] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Plethysmogram-based secure inter-sensor communication in body area networks," in *Proceedings of the 2008 IEEE Military Communications Conference, MILCOM 2008 - Assuring Mission Success*, San Diego, Calif, USA, November 2008.

[13] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "EKG-based key agreement in body sensor networks," in *Proceedings of the 2008 IEEE INFOCOM Workshops*, pp. 1–6, Phoenix, Ariz, USA, April 2008.

[14] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "PSKA: usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2010.

[15] J. Zhou, Z. Cao, and X. Dong, "BDK: Secure and efficient biometric based deterministic key agreement in wireless body area networks," in *Proceedings of the 8th International Conference on Body Area Networks, BODYNETS 2013*, pp. 488–494, Boston, Mass, USA, October 2013.

[16] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 410–419, Chicago, Ill, USA, November 2009.

[17] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: new directions for implantable medical device security," in *Proceedings of the 3rd USENIX Workshop on Hot Topics in Security*, San Jose, Calif, USA, January 2008.

[18] X. Hei, X. Du, J. Wu, and F. Hu, "Defending resource depletion attacks on implantable medical devices," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, Seattle, Wash, USA, December 2010.

[19] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "IMDGuard: securing implantable medical devices with the external wearable guardian," in *Proceedings of the 30th IEEE International Conference on Computer Communications*, pp. 1862–1870, Shanghai, China, April 2011.

[20] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: Non-invasive security for implantable medical devices," in *Proceedings of the ACM SIGCOMM 2011 Conference, SIGCOMM '11*, pp. 2–13, Toronto, Ontario, Canada, August 2011.

[21] W. Shen, P. Ning, X. He, and H. Dai, "Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time," in *Proceedings of the 34th IEEE Symposium on Security and Privacy, SP 2013*, pp. 174–188, Berkeley, Calif, USA, May 2013.

[22] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Y. Huang, *Body Area Network Security: Robust Key Establishment Using Human Body Channel*, HealthSec, 2012.

[23] S. Cherukuri, K. K. Venkatasubramanian, and S. K. S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Proceedings of the 32nd International Conference on Parallel Processing (ICPP '03)*, pp. 432–439, Kaohsiung, Taiwan, October 2003.

[24] R. M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, and C. I. De Zeeuw, "Peak misdetection in heart-beat-based security: Characterization and tolerance," in *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, pp. 5401–5405, Chicago, Ill, USA, August 2014.

[25] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. De Zeeuw, "Enhancing heart-beat-based security for mhealth applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 254–262, 2017.

[26] C. H. Raine III, L. E. Schrock, S. V. Edelman et al., "Significant insulin dose errors may occur if blood glucose results are obtained from miscoded meters," *Journal of Diabetes Science and Technology*, vol. 1, no. 2, pp. 205–210, 2007.

[27] C. Camara, P. Peris-Lopez, and J. E. Tapiador, "Security and privacy issues in implantable medical devices: A comprehensive survey," *Journal of Biomedical Informatics*, vol. 55, pp. 272–289, 2015.

[28] Parkisons disease, http://www.medtronic.eu/your-health/parkinsons-disease/device/our-dbs-therapy-products/activaRC/index.htm/.

[29] V. S. Mallela, V. Ilankumaran, and S. N. Rao, "Trends in cardiac pacemaker batteries," *Indian Pacing and Electrophysiology Journal*, vol. 4, no. 4, pp. 201–212, 2004.

[30] M. Rostami, W. Burleson, A. Juels, and F. Koushanfar, "Balancing security and utility in medical devices?" in *Proceedings of the 50th Annual Design Automation Conference, DAC 2013*, Austin, Tex, USA, June 2013.

[31] H. Jiang, J. Zhang, D. Lan et al., "A low-frequency versatile wireless power transfer technology for biomedical implants," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, no. 4, pp. 526–535, 2013.

[32] K. M. Silay, C. Dehollain, and M. Declercq, "A closed-loop remote powering link for wireless cortical implants," *IEEE Sensors Journal*, vol. 13, no. 9, pp. 3226–3235, 2013.

[33] R. F. Xue, K. W. Cheng, and M. Je, "High-efficiency wireless power transfer for biomedical implants by optimal resonant load transformation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 867–874, 2013.

[34] W. Choi, I. S. Kim, and D. H. Lee, "E2PKA: An Energy-Efficient and PV-Based Key Agreement Scheme for Body Area Networks," *Wireless Personal Communications*, vol. 97, no. 1, pp. 977–998, 2017.

[35] D. Pitcher, J. Soar, K. Hogg et al., "Cardiovascular implanted electronic devices in people towards the end of life, during cardiopulmonary resuscitation and after death: guidance from the Resuscitation Council (UK), British Cardiovascular Society

and National Council for Palliative Care," *Heart*, vol. 102, pp. A1–A17, 2016.

[36] J. Kranjec, S. Beguš, G. Geršak, and J. Drnovšek, "Non-contact heart rate and heart rate variability measurements: A review," *Biomedical Signal Processing and Control*, vol. 13, no. 1, pp. 102–112, 2014.

[37] W. Verkruysse, L. O. Svaasand, and J. S. Nelson, "Remote plethysmographic imaging using ambient light," *Optics Express*, vol. 16, no. 26, pp. 21434–21445, 2008.

[38] P. J. M. Fard, M. H. Moradi, and M. R. Tajvidi, "A novel approach in R peak detection using Hybrid Complex Wavelet (HCW)," *International Journal of Cardiology*, vol. 124, no. 2, pp. 250–253, 2008.

[39] A. Ghaffari, H. Golbayani, and M. Ghasemi, "A new mathematical based QRS detector using continuous wavelet transform," *Computers and Electrical Engineering*, vol. 34, no. 2, pp. 81–91, 2008.

[40] J. P. V. Madeiro, P. C. Cortez, J. A. L. Marques, C. R. V. Seisdedos, and C. R. M. R. Sobrinho, "An innovative approach of QRS segmentation based on first-derivative, Hilbert and Wavelet Transforms," *Medical Engineering & Physics*, vol. 34, no. 9, pp. 1236–1246, 2012.

[41] S. Blake-Wilson and A. Menezes, "Authenticated diffe-hellman key agreement protocols," in *International Workshop on Selected Areas in Cryptography*, vol. 1556 of *Lecture Notes in Computer Science*, pp. 339–361, Springer, Berlin, Germany, 1998.

[42] Z. Alliance and et al., "Zigbee specification," 2006.

[43] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.

[44] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 523–540, Springer, Berlin, Germany, 2004.

[45] E. B. Barker and A. L. Roginsky, "Transitions: recommendation for transitioning the use of cryptographic algorithms and key lengths," National Institute of Standards and Technology NIST SP 800-131a, 2011.

[46] A. L. Goldberger, L. A. Amaral, L. Glass et al., "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals.," *Circulation*, vol. 101, no. 23, pp. E215–E220, 2000.

[47] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[48] R. Bousseljot, D. Kreiseler, and A. Schnabel, "Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet," *Biomedizinische Technik. Biomedical Engineering*, vol. 40, no. 1, pp. 317-318, 1995.

[49] Entropy, https://kr.mathworks.com/matlabcentral/fileexchange/28692-entropy.

[50] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, New York, NY, USA, 2003.

[51] K. Ren, W. Lou, K. Zeng, and P. J. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4136–4144, 2007.

[52] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, Kauai Island, Hawaii, USA, March 2005.

[53] N. Wessel, C. Ziehmann, J. Kurths, U. Meyerfeldt, A. Schirdewan, and A. Voss, "Short-term forecasting of life-threatening cardiac arrhythmias based on symbolic dynamics and finite-time growth rates," *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 61, no. 1, pp. 733–739, 2000.

[54] C. Kamath, "Entropy-based algorithm to detect life threatening cardiac arrhythmias using raw electrocardiogram signals," *Middle East Journal of Scientific Research*, vol. 12, no. 10, pp. 1403–1412, 2012.

[55] A. A. Kuznetsov and S. A. Permyakov, "Distribution functions for patient heart rate parameters," *Measurement Techniques*, vol. 58, no. 5, pp. 567–573, 2015.

[56] P. Verbeeten, S. Ray, and M. Peters, "G450(p) the distribution of heart rate in critically ill children," 2017.