Secure Key-Exchange Protocol for Implants Using Heartbeats

Robert M. Seepers[1*], Jos H. Weber[2], Zekeriya Erkin[2], Ioannis Sourdis[3] and Christos Strydis[1]
[1]Dept. of Neuroscience, Erasmus Medical Center, The Netherlands
[2]Cyber Security Group, Delft University of Technology, The Netherlands
[3]Dept. of Computer Science & Engineering, Chalmers University of Technology, Sweden
Corresponding authors: {r.seepers, c.strydis}@erasmusmc.nl

ABSTRACT

The cardiac interpulse interval (IPI) has recently been proposed to facilitate key exchange for implantable medical devices (IMDs) using a patient's own heartbeats as a source of trust. While this form of key exchange holds promise for IMD security, its feasibility is not fully understood due to the simplified approaches found in related works. For example, previously proposed protocols have been designed without considering the limited randomness available per IPI, or have overlooked aspects pertinent to a realistic system, such as imperfect heartbeat detection or the energy overheads imposed on an IMD. In this paper, we propose a new IPI-based key-exchange protocol and evaluate its use during medical emergencies. Our protocol employs fuzzy commitment to tolerate the expected disparity between IPIs obtained by an external reader and an IMD, as well as a novel way of tackling heartbeat misdetection through IPI classification. Using our protocol, the expected time for securely exchanging an 80-bit key with high probability $(1 - 10^{-6})$ is roughly one minute, while consuming only 88 μJ from an IMD.

CCS Concepts

•**Security and privacy → Key management; Biometrics; Access control;** *Mobile and wireless security;*

1. INTRODUCTION

Modern implantable medical devices (IMDs) are lightweight embedded devices equipped with wireless capabilities to support non-invasive treatment updates and maintainability [20]. Given the need for protecting medical data and the life-critical function of an IMD (I), they face stringent security requirements and require an external reader (R) to connect to it securely. At the same time, I should always be accessible during emergencies, as patient safety severely outweighs IMD security [8]. For example, an emergency medical technician may need to obtain recent IMD data logs to expedite patient diagnosis. In most cases, however, I and R are unknown to each other, i.e., they do not share a secret key which is required for secure communication.

To be able to communicate, I and R need to exchange a secret key K_{RI} in a trusted manner. One promising way to facilitate this exchange uses the cardiac interpulse interval (IPI), that is, the time difference between two consecutive heartbeats [14, 15, 17, 23]. Each IPI contains a number of random bits, which may only be obtained (with some consistency) by R and I if they simultaneously measure a cardiac signal from the same person. These characteristics essentially make IPIs time- and person-specific random numbers, which allows it to be used for entity authentication [15, 17] or, as is targeted in this work, key exchange [1, 4, 12]. In the latter case, I (generates and) commits K_{RI} using its own IPIs; R may subsequently decommit K_{RI} using R's IPIs.

While various key-exchange protocols using IPIs have been proposed [1, 4, 12], they have several shortcomings that limit their applicability in real systems. For example, some of these studies either do not take the (limited) randomness available per IPI into account, have overlooked IPI-related practicalities (e.g., imperfect heartbeat detection) or do not consider the requirements of IMDs, such as minimal energy consumption. In this paper, we propose and evaluate a new security protocol which uses IPIs for key exchange through using the fuzzy-commitment security primitive [10]. Our protocol distinguishes itself from related work through a novel way of overcoming heartbeat misdetection, based on heartbeat classification and ignoring any IPIs which have been affected by misdetection. We demonstrate that our protocol is suitable for IMDs by evaluating both how fast and reliably a key can be exchanged, as well as its computational and communication overheads.

The remainder of this paper is structured as follows: First, we detail the fuzzy-commitment scheme in Section 2. In Section 3 we review studies related to the randomness and reliable measurement of IPIs as well as other key-exchange protocols based on IPIs. Our protocol is detailed in Section 4 and is subsequently evaluated in Section 5. Finally, our obtained results are put into perspective in our discussion in Section 6, after which concluding remarks are provided in Section 7.

2. BACKGROUND

During trust establishment (detailed in Section 4.2), external reader R and IMD I use IPIs to derive witnesses w_R and w_I, respectively, where $w_R \approx w_I$. These witnesses are

*This work has been supported by the EU-funded project SHARCS (Grant agreement no: 644571).

CF'16, May 16-19, 2016, Como, Italy

DOI: http://dx.doi.org/10.1145/2903150.2903165

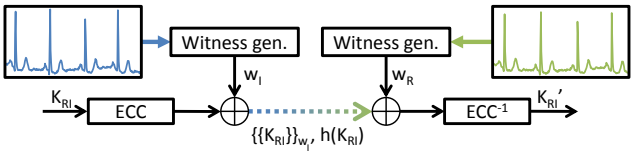


Figure 1: Key exchange using fuzzy commitment and heartbeats.

freshly generated, random numbers: As such, a secret (in our case, symmetric key) K_{RI} can be encrypted using w_I in a similar fashion to a one-time pad (OTP) [23], i.e., using a simple xor operation ($K_{RI} \oplus w_I$). If $w_R = w_I$, R may subsequently decrypt K_{RI} using w_R (as $K_{RI} \oplus w_I \oplus w_R = K_{RI}$).

Unfortunately, it is common in IPIs (and biometrics, in general) that $w_R \approx w_I$, i.e., $w_R \neq w_I$, which prohibits R from successfully decrypting K_{RI} using OTPs. The fuzzy commitment scheme (illustrated in Figure 1) overcomes this limitation by applying error-correcting codes (ECC) to K_{RI} prior to encryption [10]. That is, I commits K_{RI} as $\{\{K_{RI}\}\}_{w_I} = ECC(K_{RI}) \oplus w_I$, where $\{\{x\}\}$ is the commitment of x. As part of fuzzy commitment, I also calculates the hash of K_{RI}, $h(K_{RI})$, and sends this data to R ($I \rightarrow R :< \{\{K_{RI}\}\}_{w_I}, h(K_{RI}) >$). R may subsequently obtain K'_{RI} through the inverse process of commitment, where $K'_{RI} = K_{RI}$ iff $w_R \approx w_I$. To validate the correct exchange of K_{RI}, R calculates its own hash $h(K'_{RI})$ and compares it to $h(K_{RI})$, which match iff $K'_{RI} = K_{RI}$.

3. RELATED WORK

The feasibility of key exchange using IPIs depends both on the randomness of IPIs (as only random bits can be used for fuzzy commitment) and the disparity between IPIs obtained by different entities (inter-witness disparity). In this Section, we therefore first discuss a number of studies which evaluate these IPI characteristics, after which we consider other IPI-based key exchange protocols.

The randomness of each IPI has been evaluated in several studies, considering both healthy subjects during rest and patients with cardiovascular diseases such as hypertension and arrhythmias [15, 17–19, 22]. In all these studies, it is concluded that the four least-significant bits (LSBs) of each IPI may be considered to be independently distributed random variables. In this work, we therefore consider the four LSBs of each IPI usable for fuzzy commitment. It should be noted that cases exist where the entropy per IPI is severely limited (e.g., subjects during exercise [18]) or that no heartbeats may be obtained from a patient at all (e.g., cardiac arrest). We discuss the application of our key-exchange protocol for these cases in Section 6.

The inter-witness disparity is determined both by the inter-sensor variability[1] (VAR_{IS}) and the probability p_{det} of each entity (I or R) detecting each heartbeat correctly[2]. VAR_{IS} is affected by how R and I obtain their IPIs: As such, various models for VAR_{IS} have been proposed, including the difference in measurements between two ECG leads [5, 15, 23], an ECG lead and PPG (Photoplethysmography) [2, 14] and an ECG lead and BP (blood pressure) [17–19]. In this work, we evaluate our key-exchange protocol using two of these models which we consider representative for typical IMD-emergency scenarios.

While most studies related to IPI-based security have been evaluated with VAR_{IS} in mind, most do not consider heartbeat misdetection. However, a recent, preliminary study has shown that it may have a significant effect on inter-witness disparity [17]. In this paper, we demonstrate that fuzzy commitment (on its own) cannot tolerate the disparity resulting from heartbeat misdetection. We subsequently tackle this problem by ignoring IPI blocks which have been affected by misdetection during witness generation.

IPI-based key exchange has previously been proposed for device pairing in body-area networks [1, 4, 12]. More recently, an IPI-based data exchange protocol has been proposed to facilitate reader-IMD communication during emergencies [23]. All of these protocols rely on either the fuzzy commitment [10] or the (closely related) fuzzy vault [9] security primitive to provide secure key exchange. These existing studies, however, have either not taken the limited randomness available per IPI into account [4, 12], or have overlooked one or more practicalities, such as heartbeat misdetection [1,4,23] or the (energy) requirements of IMDs [12]. As a result, we consider these protocols too simplistic for practical use. In this work, we present a new IPI-based key-exchange protocol for IMDs which, similar to existing protocols, achieves secure key exchange through the fuzzy commitment security primitive. Ours, however, overcomes its primary limitation (heartbeat misdetection) by excluding IPIs which have been affected by misdetection prior to commitment. Furthermore, our protocol considers both the limited randomness available per IPI and is tailored to the constraints and requirements of IMDs during emergencies.

4. KEY EXCHANGE USING HEARTBEATS

In typical IMD security, IMD I employs a symmetric (private key) cipher to facilitate lightweight, secured communication [20]. An external reader R can therefore only communicate to I if a key K_{RI} is shared between the two. During emergencies, however, R and I are likely unknown to each other and therefore do not share a secret key. Our goal is, then, to transfer K_{RI} from I to R in a secure and trustworthy manner, which subsequently allows R to communicate to I using its regular security protocol. To do so, R and I first need a way to establish trust between them, after which key exchange ensues under concealment of this trust.

In the following Sections, we first introduce our adversarial model, after which we detail how IPIs are used to generate trust between R and I by generating witnesses w_R and w_I, respectively. We subsequently present our key-exchange protocol in Section 4.3, which exchanges K_{RI} using fuzzy commitment and witnesses w_R and w_I.

4.1 Adversarial model

The goal of an adversary A is to obtain either K_{RI}, w_R or w_I such that he can either gain access to I or obtain private information from the (secure) communication between R and I. Our protocol is designed for an active adversary who has full control of the channel and may eavesdrop, mod-

[1]Assuming precise and non-drifting sensors, VAR_{IS} is the variance between two different sensor measurements of cardiac biosignals, caused by the variable pulse-transition time of ventricular contraction (heartbeats) to the rest of the body due to, for example, motion and pressure differences.
[2]Incorrect heartbeat detection (the detection of a non-existing heartbeat or not-detecting an actual heartbeat) results from, among others, sensor-movement artifacts or imperfections in the detection algorithm.

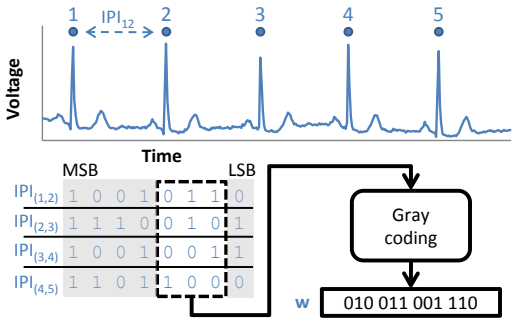


Figure 2: Witness generation.

ify, drop and replay messages sent from R to I and vice versa, in addition to forging his own messages. While this is a rather strong assumption (it is unlikely that A is present during an actual emergency), we consider IMD security important enough to assume such a worst-case model. Despite his capabilities, A is not able to measure the IPI bits which are used for witness generation: This assumption is supported by related work which has shown that remote-measuring techniques are not able to provide a statistical advantage over mere guessing [15].

4.2 Trust establishment

We aim to establish trust between R and I through using IPIs, a cardiovascular biometric which is defined as the time difference between two consecutive heartbeats. Most studies conclude that the 4 least-significant bits of each IPI can be considered random [15, 17–19, 22]. Despite this randomness, R and I may both obtain the same IPI bits (with some consistency – minor disparities are common in biometrics) iff they simultaneously measure the same heartbeats on the same body. These characteristics allow R and I to use IPIs for generating random witnesses w_R and w_I (where $w_R \approx w_I$), providing a basis for trust establishment. The most common way of generating these witnesses (used in [2, 14, 17–19]) is depicted in Figure 2: First, a number of heartbeats are detected from a cardiac biosignal and the time interval between consecutive heartbeats is calculated to form IPIs, i.e., $IPI_{(i,i+1)} = beat_{i+1} - beat_i$. A predefined set of (random) bits is selected from each IPI: The most significant bits (MSBs) are typically discarded due to their inherent low entropy, while the least significant bits (LSBs) may be discarded due to inter-sensor variability (VAR_{IS}[1]). Gray coding is applied to the selected IPI bits in order to strengthen them against VAR_{IS} (reducing the number of bits affected by a small disparity between IPIs). Finally, the Gray-coded bits from consecutive IPIs are concatenated to form a witness.

The trust formed between R and I hinges on the assumption that $w_R \approx w_I$ iff R is physically proximal to I (i.e., capable of touching the patient), which is a common assumption for emergency-trust establishment [16]. Logically, an adversary A *could* try to gain access to the IMD by generating a witness $w_A \approx w_I$ using the same method as R. However, we expect that the risk of abusing this mechanism is minimal, as: 1) It is unlikely that a patient would not notice (or allow) A attaching a heartbeat sensor to him/her; 2) Sensors have to be fastened steadily to the patient for successful trust establishment, as there would otherwise be a significant disparity between the generated witnesses (see

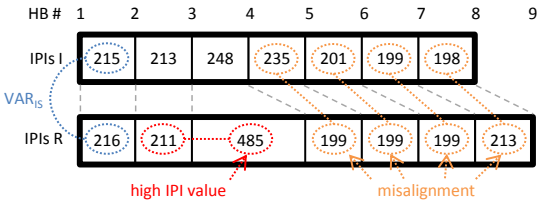


Figure 3: Illustration of inter-witness disparity, showing the IPIs obtained by R and I (in decimals) and annotations highlighting the disparity. In this example, R does not detect heartbeat 4.

Section 5.2); and 3) In the unlikely event that A can satisfy both (1) and (2), it is reasonable to expect that A would have easier methods of harming the patient (e.g., physically attack the patient or use social engineering to obtain his password). That is, IMD security may not be deemed as the most crucial for the patient's well-being in such cases.

The performance and overheads of our key-exchange protocol and the size of w_R and w_I depends largely on the expected inter-witness disparity, as a larger disparity requires stronger error-correcting codes for fuzzy commitment. To understand some of our design choices, we first exemplify the two causes of inter-witness disparity using Figure 3, which depicts the IPIs generated by R and I. Also highlighted in Figure 3 are the following: 1) VAR_{IS}: When R and I obtain an IPI from the same heartbeats, a minor disparity between these IPIs occurs as R and I detect each heartbeat with slight variations due to inter-sensor variability VAR_{IS}; and 2) *Heartbeat misdetection*: It can happen that R or I fails to detect a heartbeat (or detects a fake heartbeat) due to, for example, movement artifacts or imperfections in the detection algorithm. In the example of Figure 3, R has failed to detect heartbeat 4, resulting in the following effects: i) R incorrectly bases $IPI_{(3,4)}$ on heartbeats 3-5, resulting in a value considerably different than its other IPIs and I's $IPI_{(3,4)}$; and ii) The remaining IPIs used for witnesses generation are misaligned as R has generated one less IPI using heartbeats 3-5 than I. That is, heartbeat misdetection introduces *order variance* between w_R and w_I.

Fuzzy commitment can tolerate the random bit permutations stemming from VAR_{IS} effectively using error correcting codes, such as BCH codes [10]. A known limitation of the scheme, however, is that it is not capable of dealing with order variance which, in our case, is introduced by heartbeat misdetection[3]. As such, we have to ensure that R and I use the same IPIs for witness generation. To do so, we opt for having R and I first determine if a misdetection has occurred in a block of IPIs during witness generation and, if so, both entities replace the entire block for freshly obtained IPIs. While this process is further explained in Section 4.3, we first introduce here a proof-of-concept classification algorithm used by R and I to identify misdetections.

Our classification algorithm in Listing 1 essentially employs a double-thresholding mechanism to distinguish correct and misdetected IPIs by considering the substantially higher (or lower) IPI values resulting from it. The algorithm first calculates the mean of a block of IPIs and subsequently

[3]We could opt to use an order-invariant derivative of fuzzy commitment (called fuzzy vault [9]). However, we argue (in Section 6) that this scheme would be too resource-heavy for IMD application.

Listing 1: Heartbeat-classification algorithm (pseudo code).

```
Input: IPIb #block of IPIs
       Th_l, Th_u #classification thresholds
Output: m #misdetection flag

m = 0;
IPIm = mean(IPIb);
for i = 0: len(IPIb):
  if IPIb[i] < IPIm * Th_l or IPIb[i] > IPIm * Th_u:
    m = 1;
    return;
```

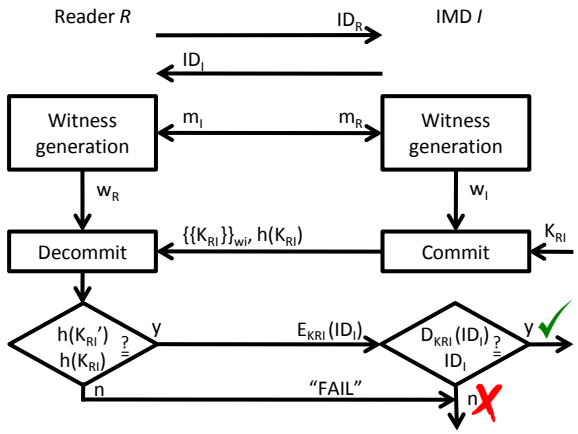


Figure 4: Key exchange protocol.

compares each IPI to this mean value and two thresholds, lower threshold Th_l ($Th_l < 1$) and upper threshold Th_u ($Th_u > 1$), to distinguish between detecting a fake and missing a real heartbeat, respectively. If a block of IPIs contains a misdetection for either R or I, they respectively set their misdetection flag (m_R or m_I, resp.) to 1. We expect that the simplicity of our algorithm favors the tight energy budget of IMDs. It should be noted that our algorithm – as with any classification algorithm – may result in both a number of false positives (i.e., a regular heartbeat is considered misdetected) and false negatives (i.e., a misdetection is not identified). These classification rates are a function of the IPI-block size, Th_l and Th_u and are further considered in our evaluation in Section 5.2.

4.3 Key-exchange protocol

Based on our adversarial model in Section 4.1, we next detail our security protocol in a generic fashion, after which we discuss specific implementation details.

4.3.1 Protocol design

Our key-exchange protocol is depicted in Figure 4 and comprises the following four steps:

S1. *Initialization*: R and I initiate the key-exchange session. R first sends its identifier to start the session with I ($R \rightarrow I : ID_R$) after which I replies with its own identifier ($I \rightarrow R : ID_I$). These identifiers are used for binding the key K_{RI} to identifiers ID_I and ID_R upon successful exchange, allowing R and I to communicate using Is regular security protocol.

S2. *Witness generation*: R and I generate witnesses w_R and w_I from their respective heartbeat measurements following the methodology described in Section 4.2. To generate these witnesses, R and I simultaneously obtain a block of IPIs and classify if a misdetection has occurred: If so, they set their respective misdetection flags (m_R and m_I) to 1. These flags are subsequently exchanged ($R \rightarrow I : m_R$ and $I \rightarrow R : m_I$) and, if a misdetection has occurred ($m_R \vee m_I = 1$), both entities replace the block with fresh IPIs. R and I resynchronize after each block (using m_R and m_I) to prevent a misdetection from affecting multiple blocks [17]. This process is repeated until enough IPIs are obtained to generate w_R and w_I.

While w_R and w_I are transmitted in plain text, this does not provide an adversary A with an advantage. First, overhearing m_I or m_R does not provide any useful information, except that a certain block of IPIs is not used. As this block is replaced with fresh and random IPI bits, the secrecy of the fuzzy commitment is not threatened. Secondly, if A were to modify, drop, replay, delay or insert its own m_R and/or m_I, the net effect would be that R and I either do not agree on which IPIs to use (i.e., introduce order variance between w_R and w_I) or continuously drop all IPIs. The security of our protocol hinges on the randomness of w_I, which is not affected by using different IPIs from I and/or R [17]. As such, malicious m_R or m_I do not provide A with any insights into w_R, w_I or K_{RI} (exchanged in the following protocol step).

S3. *Fuzzy commitment*: I generates a random secret key K_{RI} and commits it using w_I and fuzzy commitment. As both K_{RI} and w_I are fresh and random, fuzzy commitment facilitates perfect secrecy [23]. This commitment is sent to R ($I \rightarrow R :< \{\{K_{RI}\}\}_{w_I}, h(K_{RI}) >$). R subsequently decommits K'_{RI} using w_I and the inverse process of commitment. As in the original fuzzy commitment scheme, we employ BCH codes as they provide strong error-correcting capabilities to random the bit flips resulting from VAR_{IS} [10].

S4. *Key validation*: To validate the correct decommitment of K_{RI}, R computes its own hash $h(K'_{RI})$ and compares it to the hash received from I in its fuzzy commitment ($h(K_{RI}) \stackrel{?}{=} h(K'_{RI})$). If these hashes match, R encrypts ID_I using K_{RI} and its regular cipher and sends this to I ($R \rightarrow I :< E_{K_{RI}}(ID_I) >$). The security of this step assumes that A cannot obtain K_{RI} from this encryption even if ID_I is known, which is the case for modern ciphers. I subsequently decrypts ID_I using its cipher and K_{RI} and, if the decrypted $D_{K_{RI}}(ID_I) = ID_I$, key exchange is a success. Alternatively, if either $h(K_{RI}) \neq h(K'_{RI})$ or $D_{K_{RI}}(ID_I) \neq ID_I$, the key exchange has failed. This step guarantees mutual trust, as both R and I (implicitly) validate if $w_R \approx w_I$.

4.3.2 Implementation aspects

In line with a recently proposed security protocol for IMDs, ID_R and ID_I are chosen to be 96 bits long [20]. As K_{RI} is used in subsequent communication between R and I using I's regular protocol, it has to adhere to the key-length requirements of I's regular cipher. We consider the PRESENT-80 cipher well-suited for IMD cryptography, given its minimal energy footprint [3] and, as such, K_{RI} is chosen to be 80

bits long. Moreover, as PRESENT-80 has a block size of 64 bits, the encryption $E_{K_{RI}}(ID_I)$ uses the 64 least-significant bits of ID_I. The expected disparity between the IPIs obtained by R and I determines the length of the witnesses BCH codes in fuzzy commitment. While the particular design choices for obtaining these parameters is left for our evaluation in Section 5.2, we briefly state that our protocol uses 3 bits per IPI to generate 204-bit long witnesses and (204, 80, 37) BCH codes (a shortened version of the (255, 131, 37) BCH code). That is, the 80-bit long K_{RI} is encoded using 204 bits, which creates a Hamming-distance of 37 bits between code words. Finally, for hashing K_{RI} we rely on SHA-3, a recommended, collusion-resistant hash function [13], which emits a 224-bit long hash. This hash sufficiently protects the secrecy of K_{RI}, given its size and randomness.

5. EVALUATION

In this Section we evaluate how fast and reliably our protocol may exchange a key from I to R, as well as its computational and communication overheads. We start in Section 5.1 by introducing our evaluation methodology, figures of merit and input datasets and proceed in Section 5.2 with reporting our results.

5.1 Experimental setup

Our protocol is quantified in terms of the probability of successful exchange p_{ex} (i.e., the probability that $K'_{RI} = K_{RI}$), the key-exchange time T_{ex} and I's computational and communication overheads. During emergency situations, it is crucial that the key exchange between I and R is successful, given the (life-) critical situation of a patient. In line with related works [15, 17, 19], we therefore target a high key-exchange rate with $p_{ex} = 1 - 10^{-6}$ and where we try to obtain $T_{ex} < 60$ seconds (commonly reported values for T_{ex} vary between 30-60 seconds). p_{ex} and T_{ex} are dependent on the fuzzy-commitment size (n_c): An increased commitment size allows for the use of stronger error-correcting codes (increasing p_{ex}) while also requiring more heartbeats to form w_R and w_I (increasing T_{ex}). In turn, n_c depends on inter-witness disparity due to VAR_{IS}, the rate at which heartbeats are detected correctly (p_{det}) and how well misdetections are classified correctly.

We evaluate p_{ex} and T_{ex} using two models of VAR_{IS}: *ECG-ECG* [15] and *ECG-BP* [17]. In both models, I is modeled as an ECG recording, which we consider realistic for I as IMDs are typically implanted close to the heart and, therefore, likely have access to some form of accurate heartbeat measurements (using ECG). The IPIs of I are obtained from the *MIT-BIH* arrhythmia dataset, which contains 30-minute ambulatory ECG recordings of 48 subjects (average heart rate: 68 beats per minute (BPM)). From this dataset, we have selected recordings where subjects do not experience severe episodes of arrhythmia (these cases are discussed in Section 6). R is subsequently modeled by adding VAR_{IS} to these recordings: In *ECG-ECG* [15], R is modeled as an ECG recording (different from I), whereas it is modeled as a blood-pressure (BP) recording in *ECG-BP* [17]. We consider these models representative for R as both ECG and BP are commonly recorded during medical emergencies. Table 1 presents the average bit-error rates of these models: Notice that the less significant bits tend to suffer more from VAR_{IS} compared to the more significant bits, and that the *ECG-BP* model is considerably noisier than the *ECG-ECG* one. To find the best settings for p_{ex} and T_{ex} we, therefore, vary which of the IPI bits are selected for witness generation.

Table 1: Average bit-error rate dataset due to VAR_{IS}.

Bit #	0	1	2	3
***ECG-ECG* [15]**	0.08	0.04	0.02	0.01
***ECG-BP* [17]**	0.46	0.29	0.15	0.08

The probability of correctly detecting a heartbeat p_{det} is modeled using a random process with a uniform distribution, where heartbeats are randomly deleted (or inserted) for either of the two entities with probability $1 - p_{det}$. This generic model allows us to investigate the performance of our protocol without relying on a specific heartbeat-detection algorithm. As several detection algorithms report a detection rate of over 99% [6,11], we vary p_{det} from 1.00 down to 0.99.

The performance of our misdetection-classification algorithm is expected to vary based on its parameters, i.e., the number of input IPIs considered per block and thresholds Th_l and Th_u. We, therefore, vary these parameters and quantify the algorithm's performance in terms of sensitivity (the accuracy of correctly classifying misdetections) and specificity (the accuracy of correctly classifying heartbeats). Moreover, we consider the key-exchange time overhead due to ignoring misdetected IPI blocks for witness generation.

While our key-agreement protocol adds an extra security feature to an implant, it requires additional resources for execution. These resources are estimated by profiling the protocol on the Smart Implantable Security Core (SISC), a 5-stage low-power ASIP for IMDs with ISA extensions for security applications [20]. We are mainly interested in the memory footprint and energy consumption per session (E_{ses}), given that IMDs are effectively energy-constrained embedded systems. For the memory footprint, we consider both the static memory required for storing the different algorithms executed in the protocol, as well as the dynamic memory required for executing these algorithms.

E_{ses} is a function of both the computations performed by I (such as BCH encoding and hashing) as well as the transmission of data from I to R. To estimate the computational energy consumption (E_{cmp}), we have synthesized the SISC processor for a UMC 90nm CMOS technology in Synopsys Design Compiler using Faraday SP libraries (due to availability). We also execute the protocol on SISC in RTL simulation to extract the processor's switching activity. An accurate estimation of E_{cmp} is subsequently obtained by evaluating the switching activity for the synthesized core in Synopsys PrimeTime. Unfortunately, we cannot directly measure the energy spent on communication (E_{com}) as we lack a functional prototype of an IMD. However, to provide a first-order assessment of this overhead, we estimate E_{com} by assuming that transmitting one bit over the air costs 40 nJ of energy, as reported in related work [21]. Finally, our evaluation assumes that I has access to heartbeats at no additional (energy) costs, which is the case for the majority of IMDs.

5.2 Experimental Results

Our evaluation is organized as follows: First, we consider the effect of VAR_{IS} on p_{ex} and T_{ex}, ignoring peak misdetection (i.e., we set $p_{det} = 1.00$). We subsequently vary p_{det} from 1.00 to 0.99 and demonstrate its detrimental effects

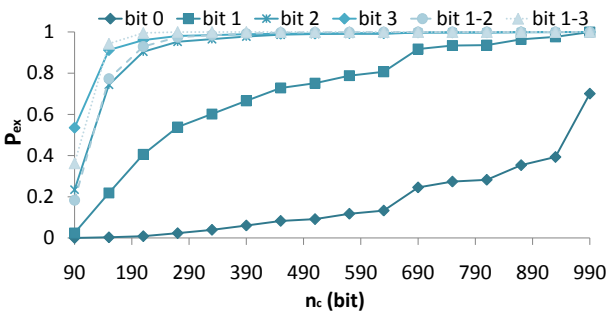


Figure 5: p_{ex} as a function of n_c for *ECG-ECG* noise and various IPI-bit selections.

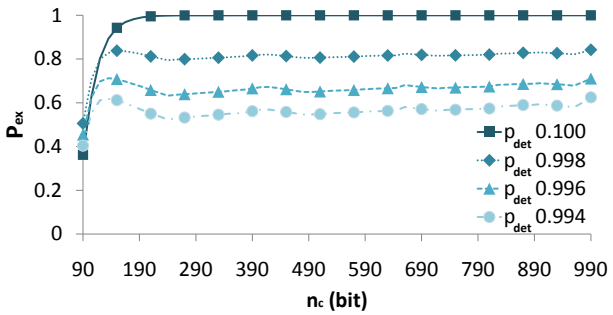
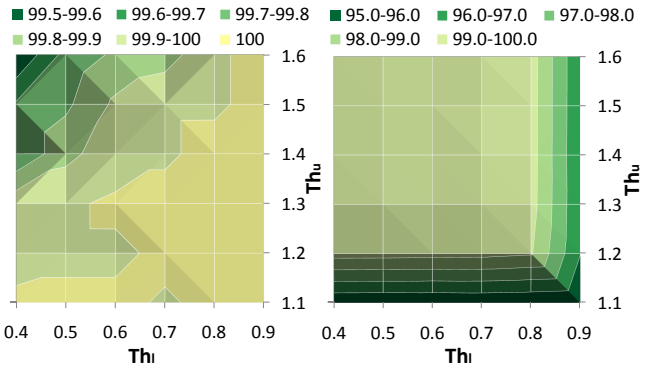


Figure 6: p_{ex} as a function of n_c for various p_{det}.



(a) Sensitivity (%) (b) Specificity (%)

Figure 7: Sensitivity and specificity as a function of Th_l and Th_u using 4 IPIs per block and $p_{det} = 0.99$.

on p_{ex} using fuzzy commitment on its own, after which we describe the results when employing our classification algorithm. Finally, we discuss the computational and communication overheads of the IMD.

First, to understand the effect of VAR_{IS} on p_{ex} and T_{ex}, we vary which bit(s) are selected per IPI and the length of error-correcting codes n_c for both our VAR_{IS} noise models. Let us first consider the results for the *ECG-ECG* model, depicted in Figure 5. Starting with the IPI-bit selection which includes only the noisiest bit (bit 0), we see that p_{ex} increases slowly as a function of n_c up to a maximum $p_{ex} = 69.4\%$ for $n_c = 1000$. It is clear that the high bit-error rate of this least-significant bit prevents us from obtaining our targeted $p_{ex} \geq 1 - 10^{-6}$, even with strong error-correcting codes. Moreover, note that $n_c = 1000$ implies that 1000 heartbeats would be used for fuzzy commitment (as only one bit is used per IPI). Given the average heart rate of 68 BPM, this would also result in an infeasible key-exchange time of $T_{ex} = \frac{1000}{68} \approx 15$ minutes.

To increase p_{ex}, we try to use a more significant IPI bit as these are less prone to VAR_{IS}. As depicted in Figure 5 (bit 1, 2, 3), we indeed find that these bits result in more favorable values for p_{ex} for a given n_c. While p_{ex} quickly reaches values for $p_{ex} > 90\%$ ($n_c = 140$), a high $n_c \geq 980$ (using bit 3 only) is still required to obtain our target p_{ex}, resulting in an infeasible key-exchange time. Finally, we observe the strongest increase in p_{ex} when the IPI-bit selection includes multiple bits (bit 2-3, bit 1-3), which may be attributed to Gray coding as it minimizes the average bit-error rate. The best selection includes 3 bits per IPI (bits 1-3), which allows for $p_{ex} \geq 1 - 10^{-6}$ for $n_c = 204$ and, given the average heart rate of 68 BPM, results in a $T_{ex} = \frac{204}{3 \cdot 68} = 1$ minute. As the *ECG-BP* model is considerably noisier than *ECG-ECG* (see Table 1) it comes as no surprise that key exchange with a high probability of success is *not* possible using this model, i.e., reliable key exchange is only possible if R obtains its measurements from ECG.

Let us now briefly consider the effect of peak misdetection on p_{ex} using fuzzy commitment *without* support of our classification algorithm. Figure 6 depicts p_{ex} for various heartbeat-detection rates p_{det} using our previously found best IPI-bit selection (bits 1-3). It is clear that heartbeat misdetection is detrimental to p_{ex}, i.e., decreasing p_{det} results in a lower p_{ex} for a given n_c. Moreover, we observe that for a given p_{det}, increasing n_c does not result in an improvement in p_{ex}, i.e., the length of the used error-correcting codes does no longer influence the probability of successful key exchange. As explained in Section 4.2, this is due to the order variance introduced by misdetection which cannot be tolerated using fuzzy commitment.

The results in Figure 6 clearly show that w_R and w_I should be generated without the influence of misdetection. The first step for doing so is classifying whether a block of IPIs contains any misdetections: Figure 7 depicts the sensitivity and specificity of our algorithm as a function of detection thresholds Th_l and Th_u, for a $p_{det} = 0.99$ and blocks of 4 IPIs (other p_{det} and block sizes are discussed later). First, when Th_l and Th_u are close to 1 ($Th_l = 0.9$, $Th_u = 1.1$), an IPI has to differ from the IPI-mean only slightly to be classified as a misdetection. As a result, we find a high sensitivity (of 100%), i.e., all misdetections are correctly classified as such, while resulting in a specificity of 95.1%. By decreasing Th_l or increasing Th_u, the difference between an IPI and the mean value has to be larger to be classified as a misdetection, leading to decreased sensitivity and increased specificity. Given the substantial impact misdetection has on p_{ex}, we aim to ensure that all misdetections are classified as such, i.e., sensitivity=100%. Under this constraint, we find a best-case specificity of 99.6% for $Th_l = 0.7$, $Th_u = 1.3$. These thresholds also hold for different p_{det} and block sizes.

Any IPI block which is classified as containing a misdetection is ignored for witness generation, resulting in an overhead to the key-exchange time T_{ex}. While this overhead (T_{ex}^{mi}) varies between key-exchange attempts (depending on the actual misdetections made), we next discuss the expected T_{ex}^{mi} for most (99.0% of all) cases. Figure 8 depicts T_{ex}^{mi} for various p_{det} and block sizes. As may be expected, T_{ex}^{mi} is increased when more misdetections are made (decrease in p_{det}) as more IPI blocks are (correctly) identified

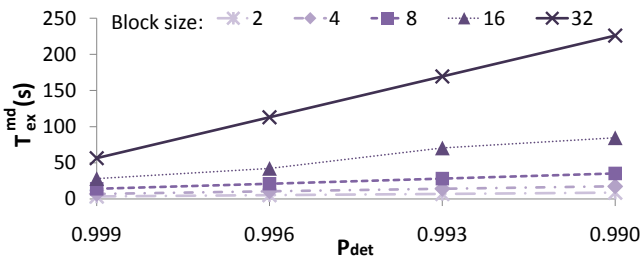


Figure 8: T_{ex}^{mi} as a function of p_{det} for various block sizes.

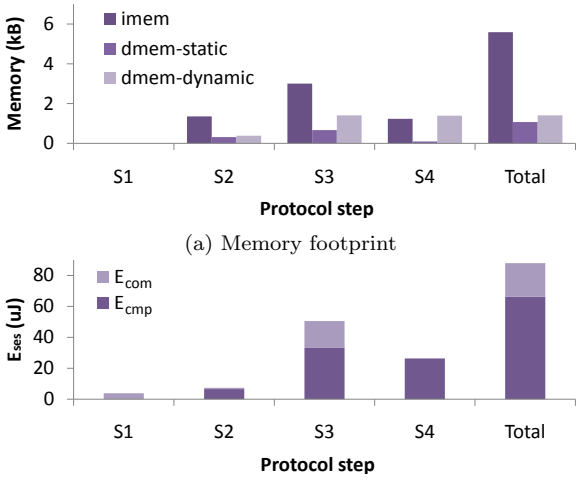
to contain a misdetection. Furthermore, a higher block size has a higher probability of being ignored for witness generation as there is a higher probability of misdetection within the block. This has a substantial impact on T_{ex}^{mi} of up to 225.8 seconds, as Figure 8 reveals. For small block sizes (2 or 4 IPIs), T_{ex}^{mi} varies between 3.5 and 17.6 seconds depending on p_{det}. That is, T_{ex} varies between 60 and 77.6 seconds in 99.0% of all cases.

Let us now consider the computational and communication overheads when executing the protocol (IMD-side) on the SISC processor. Figure 9 depicts the memory footprint and energy consumption for the different protocol steps (S1-S4) executed by I. Here, we again assume that 3 bits are selected per IPI to form 204-bit witnesses, blocks of 4 IPIs and $p_{det} = 0.99\%$. We first discuss these results for each step individually and conclude with the overheads for the protocol as a whole.

During the initiation step (S1), I responds to the incoming request of R by replying with ID_I (96 bits) and, accordingly, spends only a trivial amount of energy on communication. In the second step, I performs witness generation by classifying and eliminating misdetections for all IPI blocks, followed by concatenating these bits to form w_I. These steps require a minimal amount of computation and communication and E_{ses} is, resultingly, comparable to that of S1. Despite the low complexity of this step, we find a relatively high instruction-memory size comparable to the PRESENT-80 algorithm used in S4. This memory overhead is largely due to the multiplications used in the classification algorithm: As a minimalistic processor, SISC does *not* have a dedicated multiplier and relies on soft multiplication (emulation) instead, resulting in a large instruction binary.

In S3, I performs fuzzy commitment, comprising the commitment of K_{RI}, the hashing of K_{RI} using SHA-3 and the transmission of this commitment. While not depicted in such detail in Figure 9, we find that commitment is substantially less complex than hashing, which is responsible for roughly 75% of the memory and E_{cmp} overheads in this step. Moreover, given the substantial amount of data sent to R (428 bits), a significant part of the energy in S3 is spent on both computation and communication. Finally, the validation of successful key exchange (S4) employs the PRESENT-80 cipher for decrypting ID_R, which exhibits similar computational overheads as SHA-3.

The memory required for storing and executing the entire protocol is 6.7 kB and 1.4 kB, respectively. Note, however, that I is expected to employ hashing and encryption in it's regular security protocol regardless of key exchange, i.e., these algorithms are already stored on I. We can, therefore, exclude these algorithms from our instruction-memory overhead, resulting in a net overhead of only 2.9 kB. Finally, the



(a) Memory footprint

(b) Energy consumption

Figure 9: Memory footprint (b) and energy consumption (c) for I during the different protocol steps, using 3 bits per IPI, 204-bit long witnesses, 4 IPIs per block and $p_{det} = 0.99$.

total energy spent on computation and communication is 88 μJ per protocol session. A typical IMD consumes roughly 2.16 J per day [20] and our protocol, thus, has negligible overheads.

6. DISCUSSION

While our evaluation shows that our protocol can facilitate secure and reliable key exchange, there are a few points of discussion. First, our work has not considered patients who suffer from cardiac arrest (i.e., the patient no longer produces heartbeats) and arrhythmic patients (i.e., highly irregular heartbeat patterns). In the former case, no heartbeats are available to form a witness, while in the latter case heartbeat-patterns are so erratic that they are often classified as misdetections (our preliminary experiments reveal sensitivity scores well below 70%). In such cases of severe cardiac emergencies, our key-exchange protocol cannot be used to facilitate communication between the external reader and IMD. However, it is important to realize that patient safety significantly outweighs device security in these situations, i.e., IMD-reader communication should be established regardless of security. To do so, we believe that our protocol should be complemented using *criticality awareness* [7], where the itself detects if the patient is experiencing an emergency and, if so, allow for fail-open access. While it is generally thought that criticality awareness cannot detect a wide range of medical emergencies reliably, both cardiac arrest and arrhythmias are easy to detect, making it suitable for these specific cases.

We have observed that the expected key-exchange time varies between 60 and 77.6 seconds which, while acceptable for a proof-of-concept system, should be further reduced for practical reasons. Furthermore, it is known that certain subjects (e.g., during exercise) have significantly reduced heart-rate variability [18], which limits the randomness of each IPI (leading to increased key-exchange times or lower successful key-exchange rates). To overcome these limitations, i.e., obtain a more rapid key-exchange time while upholding security, we consider two possible modifications to our scheme: First, our work has targeted a high exchange-reliability con-

straint of $p_{ex} \geq 1 - 10^{-6}$, which could be relaxed so as to achieve more acceptable key-exchange times. Secondly, recent work has focused on entropy-extraction techniques [18, 19] where less-random IPI bits are consolidated to form (quasi-) random bits. We expect that both T_{ex} and subjects with limited heart-rate variability could benefit from such techniques, as more randomness is obtained per IPI compared to the simple bit-concatenation currently employed.

Finally, our protocol provides perfect secrecy by employing the fuzzy commitment security primitive for key exchange. A key limitation of fuzzy commitment, however, is its inability of tolerating the order variance resulting from heartbeat misdetection. To overcome this limitation, we employ a classification algorithm to identify misdetections and discard them prior to commitment. Alternatively, we could have opted to use a fuzzy vault [9], which is an order-invariant derivative of the fuzzy commitment scheme. There are, however, two reasons why we believe that a fuzzy-vault scheme does not provide a good match for our use case: 1) Fuzzy vault places significantly higher overheads than fuzzy commitment, as the 'vault size' is typically several kB in size, which has to be stored on and transmitted by the IMD. We (and others [4]) expect that this introduces non-trivial overheads, likely not suitable for the tight IMD-energy budget; and 2) Fuzzy vault typically employs different error-correcting codes (most often, Reed-Solomon codes [9]) to achieve its order invariance. The main downside of these codes is that they are not as well-suited for tolerating individual bit flips as the BCH codes used in fuzzy commitment, leading to extended code lengths.

7. CONCLUSIONS

In this work, we have proposed a new key-exchange protocol for implantable medical devices (IMDs). Heartbeats are used to establish trust between an external reader and the IMD, and key-exchange between the two entities is subsequently facilitated through fuzzy commitment. The protocol has been evaluated in terms of how fast and reliable keys may be exchanged as well as IMD overheads, considering a number of events that can happen in a realistic system, such as inter-sensor variability and heartbeat misdetection. We have proposed a simple, proof-of-concept classification algorithm, which is used to eliminate the adverse effects of heartbeat misdetection during trust establishment. Our protocol holds promise for facilitating IMD-emergency communication and exchanges an 80-bit key reliably in roughly one minute, at a minimal overhead of 88 μJ for the IMD.

8. REFERENCES

[1] S.-D. Bao et al. A novel key distribution of body area networks for telemedicine. In *IEEE BioCas*, pages 1–17, 2004.

[2] S.-D. Bao et al. Using the timing information of heartbeats as an entity identifier to secure body sensor network. In *T-ITB, pp. 772-779*, volume 12. IEEE, 2008.

[3] A. Bogdanov et al. Present: An ultra-lightweight block cipher. In *CHES*, pages 450âĂŞ–466. Springer, 2007.

[4] F. M. Bui and D. Hatzinakos. Biometric methods for secure communications in body sensor networks: resource-efficient key management and signal-level data scrambling. *EURASIP Journal on Advances in Signal Processing*, 2008:109, 2008.

[5] S.-Y. Chang et al. Body area network security: robust key establishment using human body channel. In *USENIX HealthSec*, 2012.

[6] A. Ghaffari et al. A new mathematical based qrs detector using continuous wavelet transform. *CEE*, 34(2):81–91, 2008.

[7] S. K. Gupta, T. Mukherjee, and K. Venkatasubramanian. Criticality aware access control model for pervasive applications. In *PerCom*. IEEE, 2006.

[8] D. Halperin et al. Security and Privacy for Implantable Medical Devices. *PC*, pages 30–39, 2008.

[9] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.

[10] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM CCS*, pages 28–36, 1999.

[11] J. P. Madeiro et al. An innovative approach of qrs segmentation based on first-derivative, hilbert and wavelet transforms. *MED ENG PHYS*, 34(9):1236–1246, 2012.

[12] F. Miao et al. A modified fuzzy vault scheme for biometrics-based body sensor networks security. In *IEEE GLOBECOM*, pages 1–5, 2010.

[13] NIST. Fips 202: Sha-3 standard: Permutation-based hash and extendable-output functions. *Federal Information Processing Standards Publication*, 2015.

[14] C. C. Poon et al. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Commun. Mag.*, pages 73–81, 2006.

[15] M. Rostami et al. Heart-to-heart (h2h): authentication for implanted medical devices. In *ACM CCS*, pages 1099–1112, 2013.

[16] M. Rushanan et al. Sok: Security and privacy in implantable medical devices and body area networks. *Proceedings of the IEEE S&P*, pages 529–539, 2014.

[17] R. M. Seepers et al. Peak misdetection in heart-beat-based security characterization and tolerance. *IEEE EMBC*, 2014.

[18] R. M. Seepers et al. Enhancing heart-beat-based security for mhealth applications. *IEEE J-BHI*, 2015.

[19] R. M. Seepers et al. On using a von neumann extractor in heart-beat-based security. In *IEEE Trustcom*, pages 491–498, 2015.

[20] C. Strydis et al. A system architecture, processor and communication protocol for secure implants. *TACO*, 2013.

[21] N. Verma et al. A micro-power eeg acquisition soc with integrated seizure detection processor for continuous patient monitoring. In *Symp. VLSI Circuits*, pages 62–63. IEEE, 2009.

[22] G.-H. Zhang et al. Analysis of using interpulse intervals to generate 128-bit biometric random binary sequences for securing wireless body sensor networks. *T-ITB*, 16(1):176–182, 2012.

[23] G. Zheng, G. Fang, R. Shankaran, and M. A. Orgun. Encryption for implantable medical devices using modified one-time pads. *Access, IEEE*, 3:825–836, 2015.