Enhancing Heart-Beat-Based Security for mHealth Applications

Article · November 2015

DOI: 10.1109/JBHI.2015.2496151

CITATIONS
18

READS
155

4 authors, including:

Ioannis Sourdis
Chalmers University of Technology
82 PUBLICATIONS **1,095** CITATIONS

SEE PROFILE

Chris I De Zeeuw
Erasmus University Rotterdam
478 PUBLICATIONS **20,321** CITATIONS

SEE PROFILE

Enhancing Heart-Beat-Based Security for mHealth Applications

Robert M. Seepers[1], Christos Strydis[1], Ioannis Sourdis[2], and Chris I. De Zeeuw[1]
[1]Dept. of Neuroscience, Erasmus Medical Center, Rotterdam, The Netherlands
[2]Dept. of Computer Science & Engineering, Chalmers University of Technology, Gothenburg, Sweden
[1]{r.seepers, c.strydis, c.dezeeuw}@erasmusmc.nl [2]sourdis@chalmers.se

***Abstract*—In heart-beat-based security, a security key is derived from the time difference between two consecutive heart beats (the Inter-Pulse-Interval, IPI) which may, subsequently, be used to enable secure communication. While heart-beat-based security holds promise in mobile health (mHealth) applications, there currently exists no work that provides a detailed characterization of the delivered security in a real system. In this paper, we evaluate the strength of IPI-based security keys in the context of entity authentication. We investigate several aspects which should be considered in practice, including subjects with reduced heart-rate variability, different sensor-sampling frequencies, inter-sensor variability (i.e., how accurate each entity may measure heart beats) as well as average and worst-case-authentication time. Contrary to the current state of the art, our evaluation demonstrates that authentication using multiple, less-entropic keys may actually increase the key strength by reducing the effects of inter-sensor variability. Moreover, we find that the maximal key strength of a 60-bit key varies between 29.2 bits and only 5.7 bits, depending on the subject's heart-rate variability. To improve security, we introduce the Inter-*multi*-Pulse Interval (ImPI), a novel method of extracting entropy from the heart by considering the time difference between two non-consecutive heart beats. Given the same authentication time, using the ImPI for key generation increases key strength by up to 3.4x (+19.2 bits) for subjects with limited heart-rate variability, at the cost of an extended key-generation time of 4.8x (+45 sec).**

I. Introduction

Mobile-health (mHealth) is an emerging technology which allows for continuous, remote health care through the use of mobile devices. Body-Area Networks (BANs) may provide continuous patient monitoring through the use of cheap, wearable biosensors [14]. Modern Implantable Medical Devices (IMDs) feature wireless capabilities to allow remote configuration without requiring invasive surgery or data-log broadcasting from a home-monitoring station [9]. Due to the wireless nature of mHealth solutions and the sensitivity of the data transmitted, security has shown to be an important aspect of mHealth. Non-secure communication may allow an adversary to steal private patient data or, worse, alter device parameters or even prevent treatment [8], [14].

The inter-pulse interval (IPI) of heart beats has recently been proposed for securing both wireless IMDs and BANs [17], [18], [20]. In heart-beat-based security (HBBS), each sensor measures a heart-related biosignal, for example, cardiac activity using an electrocardiogram (ECG) or blood flow, and forms a biometric security key based on the time interval between two consecutive heart beats. Previous work has shown that this interval may contain a significant degree of entropy, while it may be measured with some consistency and in different locations of a patient's body. These two characteristics allow IPIs to be used for shared-secret generation between two entities simultaneously sampling the same heart beat, thus forming the basis for security aspects such as key agreement [25], BAN-device pairing [5], [17], [27] or IMD (-emergency) authentication [18], [20].

While HBBS shows potential for mHealth applications, it is not yet clear how much security the IPI may provide in practice. The statistical properties of IPIs are not yet fully understood [19] and most related works have not considered subjects with significantly limited heart-rate variability (HRV) [5], [7], [17], [18], [21]. In addition, the effect of inter-sensor variability (VAR_{is}), i.e., the disparity between heart-beat measurements between two entities, has either been neglected [28] or has not been studied in sufficient detail [19]. A more profound understanding of how these properties affect the security of IPI-based keys could lead to new, more efficient key-generation methods.

In this paper, we evaluate the security performance of heart-beat-based security in the context of entity authentication. Specifically, this paper contributes the following:

- A thorough characterization of the strength of IPI-based keys, investigating several aspects which may occur in practice. Specifically, we consider: *1)* subjects with various degrees of HRV; *2)* different sensor sampling-frequencies; *3)* realistic VAR_{is} based on measurements obtained from ECG and blood-pressure recordings; and *4)* average and worst-case authentication time.
- The first work which considers the use of entropy extraction in HBBS, using a novel method of extraction through the Inter-*multi*-Pulse-Interval (ImPI). The ImPI considers the time difference between two non-consecutive heart beats, resulting in an unprecedented increase in key strength at the cost of an extended key-generation time.

This paper is structured as follows: First, we briefly discuss why HBBS is a suitable biometric for mHealth applications, along with related works, in Section II. In Section III, we describe the existing and improved method of generating keys in HBBS using the IPI and ImPI, respectively. These key-generators are subsequently evaluated in Section IV, after which concluding remarks are given in Section VI.

II. Background and Related Work

In this Section, we first compare HBBS to other biometrics qualitatively, after which we discuss works related to its security performance. HBBS is a form of cardiovascular biometrics, which use the characteristics of a person's cardiac cycle for entity authentication. Cardiovascular biometrics are typically based on an electrocardiogram (ECG), using either a combination of various fiducial features (e.g., "ST-slope" or "ST-interval") or non-fiducial features, for example, the autocorrelation between heart-beat records [6], [16], [23]. Conventionally, a good biometric is one that is easily measured for the general population (universality, measurability, performance), characterizes an individual well (uniqueness), is invariant over time (permanence) and is accepted by the relevant population (acceptability) [13]. HBBS differs from other cardiovascular biometrics in that it uses only a single fiducial feature, that is, the inter-pulse interval (IPI, also denoted as the "RR-interval") between heart-beats. This makes it a suitable candidate for many mHealth applications as [17]:

- Heart beats are measurable throughout the body using many types of cardiovascular recordings, including ECG, blood pressure (BP) and photoplethysmography (PPG). As such, it may be measured through a wide spectrum of sensors and locations (more universally than other cardiovascular biometrics), which is common in, for example, a BAN;
- Heart beats ("R-peaks") are arguably the most distinct feature in any cardiovascular recording, permitting low-cost peak detection and key generation; and
- Cardiac function is one of the most commonly measured values in mHealth. As a result, many systems will already have the required sensors and peak-detectors in place, allowing HBBS to be included at minimal overhead.

The downside of HBBS is that the IPI is a random (time-variant) feature which compares infavourably to other biometrics in terms of permanence [17]. However, its universality and low-cost detection permit all involved entities to generate a fresh, random key for each communication session, increasing security while bypassing several issues related to permanence, such as template outdating [2].

The key strength of an HBBS system depends on both the randomness of the generated keys and the inter-key disparity allowed for a true-key pair, as will be discussed in Section IV. Accordingly, here we first discuss relevant studies on the IPI-entropy (key-randomness), after which we review a number of related works on the inter-key disparity. The entropy per IPI stems from the heart-rate variability (HRV), a physiological phenomenon caused by the balancing action between the parasympathetic and sympathetic nervous systems [1], [4]. HRV is known to be reduced when either of these nervous systems dominates the other and is affected by, among others, smoking, age, gender, diabetes, brain damage, cardio-vascular disorders (CVDs), mental state and, pherhaps most substantially, exercise [1], [3], [4], [24]. Despite the available knowledge on HRV, only a few works have evaluated the entropy per IPI in the context of security (in bits), considering healthy subjects, hypertensive subjects as well as CVD patients [18], [21], [28], all of which conclude that four highly entropic bits are available per IPI. In addition, a recent, preliminary study has considered the effect of exercise on IPI-entropy, showing that subjects during exercise may lose up to 75% of their entropy compared to subjects at rest [20]. In this work, we build upon the work presented in [20], [21] to provide a more thorough evaluation on the entropy per IPI, considering both subjects with various degrees of HRV and different sensor sampling frequencies.

In an attempt to increase the entropy obtained from IPIs, Bao et al. [5] have proposed using the *m*ulti-Inter-Pulse Interval (*m*IPI) for key generation, where $mIPI_{(i,j)}$ is the accumulation of all IPIs previously considered for key generation, i.e., $mIPI_{(i,j)} = \sum_{i=1}^{j-1} IPI_{(i,i+1)}; j > i$. While our own experiments confirm the *apparent* increase in entropy per $mIPI$, we note that it does *not* enhance security. The *m*IPI attempts to increase randomness using a simple addition and, as famously stated by John von Neumann, "any one who considers arithmetical methods to produce random digits is, of course, in a state of sin" [26]. In this paper we present the Inter-*multi*-Pulse Interval (ImPI) which, contrary to the *m*IPI, does *not* reuse its entropic source and *does* allow for an increase in key strength, albeit at the cost of extended key-generation time. To the best of our knowledge, our work is the first to successfully apply entropy extraction in HBBS.

The inter-key disparity allowed for a true-key pair (Hamming-distance threshold, T_{HD}) is determined by the (expected) VAR$_{is}$ in an mHealth system. While all studies agree that VAR$_{is}$ results in a reduction of security performance, the characteristics of such variability are not fully understood, partially due to the different methodologies followed [19]. Poon et al. [17] and Bao et al. [5] have modeled the VAR$_{is}$ as the difference between an ECG and PPG (Photoplethysmography), showing a significant disparity between generated keys (a 2.06% false-rejection rate has been described for a 128-bit key using $T_{HD} = 48$). Another study has shown a similar disparity (describing a best-case $T_{HD} = 16$ bits for a 60-bit key) by considering VAR$_{is}$ as the difference between ECG and blood-pressure recordings [21]. Other works have either overlooked the VAR$_{is}$ [28] or have modeled it as two different leads of the same ECG [7], [18], both of which cannot be considered realistic for typical mHealth applications. In this work, we use a VAR$_{is}$ model described in [21] which considers multiple biosignals (ECG and blood pressure) measured at different locations of the same body. We consider such a model representative for typical mHealth applications, as it is likely that two different entities, for example, in a BAN, will have access to different biosignals and will be recorded from different locations. We demonstrate how the VAR$_{is}$ affects the security strength considering various parameters, including the bits selected per IPI, the (average) heart rate of a subject, multi-key authentication, sensor-sampling frequency and the average and worst-case authentication time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2015.2496151, IEEE Journal of Biomedical and Health Informatics

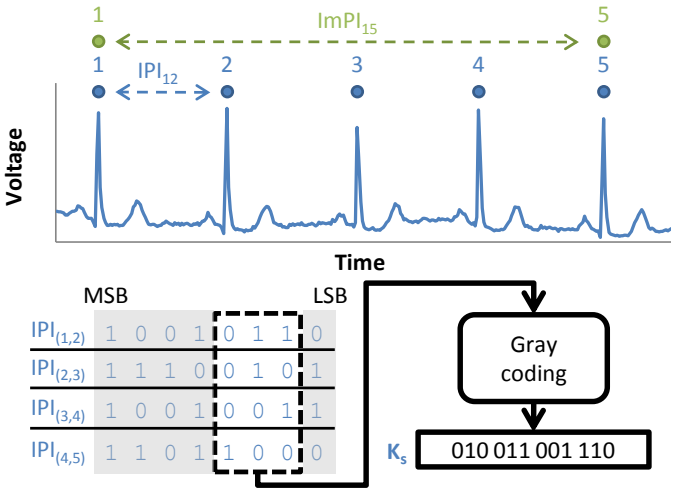


Fig. 1: Key generation using the I(m)PI.

III. Inter-(multi)-Pulse Interval

In this Section we describe the most commonly used method for facilitating entity authentication in HBBS based on the IPI, after which we present our improved method using the ImPI.

Entity authentication in HBBS comprises two steps: Security-key generation by two entities and entity-authentication, if these keys are similar enough. Figure 1 illustrates the method of security-key generation using the IPI [17], [18], [20], [21], [28]. First, each entity detects a number of heart beats from their cardiac biosignals and calculates the time interval (IPI, in this work considered as an 8-bit value) from consecutive beats, i.e., $IPI_{(i,i+1)} = beat_{i+1} - beat_i$. From each IPI, a predefined set of bits m is selected (the key-bit selection, containing n_m bits per IPI) to form a key segment: The most-significant IPI bits are commonly discarded due to their inherent low entropy, while the least-significant IPI bits may be discarded due to a high VAR_{is}[1]. Gray coding is applied to the key segment in order to strengthen it against VAR_{is} (reducing the number of bits affected by a disparity between IPIs), after which n key segments are concatenated to form security key k. Entity authentication is successful if the generated keys are similar enough (not identical, as some disparity may be expected for a given true-key pair due to VAR_{is}). This similarity is commonly assessed by comparing the Hamming distance between the keys to a predefined threshold ($hd(k_1 \oplus k_2) < T_{HD}$, where $hd(x)$ represents the number of non-zero values in x and T_{HD} denotes the Hamming-distance threshold).

It will be shown in our evaluation in Section IV-B that the strength of IPI-based keys is in part limited by the low entropy of the most-significant IPI-bits due to correlations between consecutive heart-beats. We strive to increase the key strength by replacing the IPI with the ImPI in the key-generation process, where we define the ImPI as the time difference considering j consecutive heart beats, i.e., $ImPI_{(j\cdot(i-1)+1,\ j\cdot i+1)} = beat_{j\cdot i+1} - beat_{j\cdot(i-1)+1}$. The ImPI is illustrated for $j = 4$ ($ImPI_{(1,5)}$) in Figure 1. Note that the ImPI is equivalent to the IPI for $j = 1$. By increasing j, we limit the effect of inter-IPI correlations as individual, consecutive heart-beats are ignored for key-generation, resulting in an increase in entropy per ImPI. The true-key-pair disparity, however, depends on the accuracy at which each entity may detect each heart beat in IPI/ImPI generation. As both the IPI and ImPI are calculated based on two heart beats, this disparity remains unaffected. Accordingly, it may be expected (and shown in subsequent Sections) that using the ImPI allows for an increase in key strength, albeit at an increased key-generation time (as more heart beats are required to obtain an ImPI).

[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 (heart beats) to the rest of the body due to, for example, motion and pressure differences.

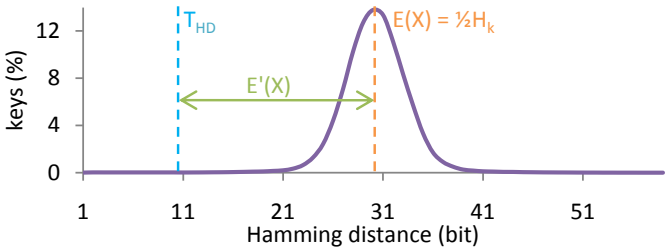


Fig. 2: Key strength KS_{eff} as a function of H_k and T_{HD}.

IV. Evaluation

In this Section, we evaluate the performance of the IPI and ImPI-based key-generators, considering the key-entropy, true-key-pair disparity and authentication time. First, we introduce our experimental setup in Section IV-A, after which our evaluation follows in Section IV-B.

A. Experimental Setup

To evaluate the performance of IPI- and ImPI-based key-generators, we first introduce the effective key strength KS_{eff} as a figure of merit. Using KS_{eff}, we may quantify the security performance as a function of the key-entropy H_k and the required Hamming-distance threshold T_{HD} for a given true-key pair. Afterwards, we present the datasets considered in our evaluation.

1) Key Strength: The strength of a key is determined by the effort required by an attacker to guess it. To quantify the key strength in bits, we define the effective key strength KS_{eff} as the number of *entropic bits* which should be known to an attacker in order to successfully authenticate to the IMD with probability $P_{auth} = 0.5$ [21]. That is, an attacker would have to mount on average $2^{KS_{eff}}$ attacks. To exemplify, in Figure 2 we plot a distribution of Hamming distances between an authentication key and various, randomly selected attacker keys. This distribution $\boldsymbol{X}$ (x being the number of mismatched bits in an n-bit key) is expectedly binomial with an average number of mismatches $E(X) = p_0 \cdot H_k = p_1 \cdot H_k = \frac{H_k}{2}$, where p_0 and p_1 denote the probability of a bit being zero or one (for entropic bits, $p_0 = p_1 = \frac{1}{2}$) and H_k denotes the number of entropic bits in the key (ideally, $H_k = \text{n}$). Since, on average, half the number of entropic bits are mismatched

by simply guessing, for successful authentication an attacker would need to try up to:

$$KS = 2 \cdot E(X) - 1 = H_k - 1 \quad bits,$$

the "-1" term accounting for $P_{auth} = 0.5$.

As it is unlikely that keys will be a perfect match due to VAR$_{is}$, we allow entities to authenticate if their keys differ no more than T_{HD} bits, where T_{HD} denotes the Hamming-distance threshold. As a result, the average number of mismatched bits will be effectively reduced by the amount of "don't care" T_{HD} bits; essentially changing $E(X)$ to $E'(X) = \frac{H_k}{2} - T_{HD}$ (see Figure 2). In this more general case, KS_{eff} is calculated as follows:

$$\begin{aligned} KS_{eff} &= 2 \cdot E'(X) - 1 \\ &= H_k - 2 \cdot T_{HD} - 1 \quad bits. \end{aligned} \tag{1}$$

Note that KS_{eff} may now assume negative values, signifying that an attacker would require less than one attack on average to guess the key ($2^{KS_{eff}} < 1$). Obviously, a negative KS_{eff} will never exist in practice as an attacker would always require at least one attack, i.e., KS_{eff} would be greater or equal to zero. Nevertheless, considering KS_{eff} as a potentially negative value will allow us to investigate exactly how far the generated keys are from providing any form of security ($KS_{eff} > 0$). To determine KS_{eff} we, thus, have to evaluate the key-entropy H_k and required Hamming-distance threshold T_{HD}, the acquisition of which is described next.

a) Entropy: The upper limit H_k of the effective key strength is determined by the randomness of the key-bit selection m (the bit-postions selected per IPI) for key generation. We assess this randomness for different m by using arithmetic-mean, autocorrelation and compression tests over the generated keys (extending the tests used in [21]):

- The arithmetic-mean test evaluates the average probability of a particular key-bit being one or zero, i.e., (P(x_i = 0), P(x_i = 1)) and, thus, represents the randomness when a bit is sampled from a key. This test reveals a bias in the key bits if $P(x_i = 0) \neq P(x_i = 1)$;
- The autocorrelation test determines the probability of a key-bit being identical to its l^{th} neighboring bit, i.e., $P(x_i = x_{i-l})$, where we choose l = 1, 2, 3, ..., 20 to determine if there are any intra-key correlations. A high value for $P(x_i = x_{i-l})$ indicates repetitive patterns in consecutive IPIs, yielding a reduction in entropy (and security) as the bits in $IPI_{(i,i+1)}$ have predictive value over those in $IPI_{(i+l,i+1+l)}$.
- The compression test splits the generated keys into c-sized symbols S and evaluates the frequency of each symbol occuring, i.e., $P(s) = \frac{\sum S=s}{\sum S}$, where $s = 1, 2, 3, ..., 2^c$, S is the value of c consecutive bits and we choose $c = 1, 2, 3, ..., 8$. A high value for $P(s)$ indicates that certain symbols (bit-patterns) s occur more frequently throughout the distribution, indicating correlations between consecutive IPIs and reducing entropy for reasons stated above.

Based on the probabilities calculated using our tests, we may compute the Shannon entropy for the arithmetic mean (H_{am}), autocorrelation (H_{ac}) and compression (H_c) tests as [22]:

$$H = \sum_i p_i \ log_2 \ p_i \tag{2}$$

where p_i is the probability of a particular event, for example, the probability of a given symbol s in the compression test. As a conservative estimation, we define the minimum entropy $H_{min} = min(H_{am}, H_{ac}, H_c)$.

H_k is expressed in terms of equivalent entropic bits, that is, the probability of guessing key k is equivalent to guessing a key with H_k truely entropic bits (where a truely entropic bit satisfies $H = 1$, $p_1 = p_0 = \frac{1}{2}$). To compute H_k, we first calculate H^i_{min} for all i IPI-bit positions and subsequently obtain p^i_0 and p^i_1 from eq. 2. For each IPI-bit position, a symbol S may be formed by concatenating n_{eq} bits. Based on p^i_0 and p^i_1, the highest probability of guessing S is $p_S = p^{n_{eq}}_{max}$, where[2] $p_{max} = max(p^i_0, p^i_1)$. By setting $p_S = \frac{1}{2}$, i.e., S is equivalently random as a truely entropic bit (as $p_S = p_{\overline{S}} = \frac{1}{2}$), we may compute the number of bits required to form S as $n_{eq} = log_{p_{max}}(\frac{1}{2})$. Accordingly, each IPI bit shall have equivalent entropy $H^i_{eq} = \frac{1}{n_{eq}}$. After calculating H^i_{eq} for each IPI-bit position, the entropy of the key-bit selection m may be obtained[3] from $H^m_{eq} = \sum_i H^i_{eq}$ for all $i \epsilon m$. Finally, as n key-segments are combined to form key k, we obtain $H_k = H^m_{eq} \cdot n$.

b) Hamming-Distance Threshold: T_{HD} is a function of the desired probability of key-matching and VAR$_{is}$. Lowering T_{HD} allows for an increase in KS_{eff} (as an attacker's key is required to be more similar to the actual key), yet also reduces the chance of successful matching for a true-key pair. To determine T_{HD}, we compare the keys generated by two entities and see at what threshold T_{HD} the keys would lead to authentication *reliably*, where we define reliable authentication of a new key as successful authentication within a predefined, upper time limit with probability $P_{auth} = 1 - 10^{-6}$ [21]. Without loss of generality, in this work we set the key length to 60 bits and the time limit to 60 seconds. We expect that such an authentication criterion will be feasible for some of the most safety-critical applications of IPI-based security, such as providing emergency-authentication credentials [18], [20]. We evaluate a 60-bit key as it is allows us to easily assess the key strength under our authentication constraints, as has been done in prior work [18], [21].

We model VAR$_{is}$ as the time difference between the heartbeats measured by ECG and blood-pressure (BP) recordings obtained from the *Fantasia dataset* [12], that is, VAR$_{is}$ = $beats_{BP} - beats_{ECG}$. We consider this model realistic for typical mHealth applications, such as a BAN, as it incorporates the effects of both different biosignals and measurement locations. As our used datasets provide ECG

[2]Given that $p_0 = 1 - p_1$, a maximum operator is used so as to get the highest probability between p_0 and p_1. This is the only way that, when concatenating multiple bits n, we can get a combined probability $p^n_{0|1} = 0.5$.

[3]In this and previous work, we have not found any *intra*-IPI dependencies (between IPI bits), permitting H^m_{eq} to be calculated as a linear addition of the H^i_{eq} of the selected IPI bits [21]

TABLE I: Average bit-error rate (BER) dataset due to VAR_{is} when applied to the *MIT-Regular* dataset.

Bit #	0	1	2	3	4	5	6	7
BER	0.46	0.29	0.15	0.08	0.04	0.02	0.01	0.00

TABLE II: Dataset specifications.

Dataset	**#Subjects**	**#IPIs**	**Avg. heart rate (BPM)**	**Sensor freq. (Hz)**
MIT-Regular	11	21696	69.3	360
MIT-Ectopic	12	16008	81.7	360
MIT-Episode	20	38424	86.4	360
RE-Rest	58	10668	75.8	200
RE-Exercise	53	11864	101.4	200

recordings only (first entity), we add VAR_{is} to these recordings to emulate BP recordings (second entity) [21]. The validity of this approach is supported by the following simularities between our model and established works which measure the second recording directly:

- The time difference between the recording are normally distributed, as also described in [11], [18];
- The bit-error rates (presented in Table I) are similar to those reported in [11]. Note that the bit-error rate is substantial (0.46) for the least-significant IPI-bits and shows an exponential decrease for more significant IPI-bits; and
- The relation between T_{HD} and the resulting authentication rate [21] is analogous to that reported in [5], [11], [17].

2) Datasets: Table II shows the number of IPIs, average heart rate (in Beats Per Minute (BPM)) and sensor-sampling frequencies of the datasets used in our experiments. As we consider CVD patients as likely users of (cardiac) IMDs, we have used the *MIT-BIH arrhythmia* (*MIT-**) dataset, a commonly used dataset containing recordings of subjects with a wide variety of CVDs [10], [15]. In order to investigate the impact of cardiac arrhythmias on the entropy of IPIs, we have split this dataset into the following subsets: *MIT-Regular:* Subjects which show less than 0.5% of abnormalities from a normal sinus rhythm; *MIT-Ectopic:* Subjects with 0.5% to 10% of their heart beats being ectopic (premature ventricular or atrial contraction); and *MIT-Episode:* Subjects which exhibit episodes of ventricular bigeminy, trigeminy, tachycardia or with more than 10% of their beats being ectopic. In addition, we have used the *Rest-And-Exercise* (*RE-**) dataset from the BioSec ECG-database [2]. This dataset contains two sets of recordings, one from subjects at rest (*RE-Rest*) and one from the same subjects immediately after exercise (*RE-Exercise*). Using the *RE-Exercise* dataset will allow us to investigate the strength of keys generated for subjects during exercise, which is known to drastically reduce HRV (and, thus, entropy per IPI) as described in Section II. Besides, as the recordings in the *RE-** dataset are sampled at 200 Hz, roughly half of the *MIT-** dataset (360 Hz), we may characterize the key strength as a function of sampling frequency by comparing the *RE-Rest* and *MIT-Regular* datasets.

TABLE III: Entropy-test results for the *MIT-Regular* dataset.

Bit	H^i_{am}	H^i_{ac}	H^i_c	H^i_{min}	H^i_{eq}
0	1.00	1.00	1.00	1.00	0.91
1	1.00	1.00	1.00	1.00	0.91
2	1.00	1.00	1.00	1.00	0.92
3	1.00	1.00	1.00	1.00	0.91
4	1.00	0.98	0.97	0.97	0.74
5	0.99	0.89	0.86	0.86	0.48
6	0.99	0.74	0.70	0.70	0.30
7	0.83	0.46	0.42	0.42	0.13

B. Experimental Results

As described in the previous Section, the effective key strength KS_{eff} is used to quantify the performance of IPI- and ImPI-based key-generators. To obtain KS_{eff}, we first provide a detailed description of the key-entropy H_k, followed by an evaluation of the required Hamming-distance threshold T_{HD} for a given true-key pair. KS_{eff} is, then, derived by considering Equation (1).

1) Entropy: In this Section, we evaluate the entropy per I(m)PI, considering the frequency and HRV characteristics of the used datasets. This evaluation is first carried out for the baseline key-generator which is based on the IPI, after which we show how the ImPI improves the key entropy.

a) IPI: Let us first consider the situation where only one bit is selected per IPI for the baseline key-generator. Table III presents the entropy results for the *MIT-Regular* dataset, showing the test results for all i IPI-bits (H^i_{am}, H^i_{ac} and H^i_c) and the resulting min-entropy H^i_{min}. Other datasets have similar results and are discussed later in this Section. In line with related work, we see that the four least-significant bits of each IPI contain a high degree of entropy, scoring the maximum 1.00 for all tests. From IPI-bit position 4 onwards, we find that the entropy results are gradually decreasing: While H^i_{am} appears mostly unaffected, we see a substantial decrease in H^i_{ac} and H^i_c. That is, these most-significant IPI-bits do not show a particular bias, they show significant correlations between consecutive IPIs (the minimum value for H^i_{ac} and H^i_c were obtained using test parameters l = 1 and c = 8, respectively), effectively reducing entropy. Table III also presents the equivalent entropy per IPI-bit H^i_{eq}. Note that even though H^i_{min} is considerably high for several bit positions (1.00), H^i_{eq} is substantially lower with a maximum value of 0.92: Due to the logarithmic scale onto which H^i_{min} is defined, even a small difference between the maximum-attainable entropy ($H^i_{min} = 1$) and the measured H^i_{min} results in a significant reduction in H^i_{eq}. For the most-significant IPI bits, the impact on H^i_{eq} is more dramatic.

To understand the effects of HRV and sensor-sampling frequency on the entropy per IPI, we depict H^i_{eq} for the various datasets in Figure 3, including confidence intervals (with a confidence coefficient of 0.01). Note that H^i_{eq} is monotonously decreasing for all datasets as a function of i, i.e., the inclusion of more significant IPI-bits in they key-bit selection m will inevitably result in a reduction of H^m_{eq}. From Figure 3, we make three interesting observations: *1)* All of the *MIT-** datasets maintain a relatively high H^i_{eq} ($\geq$ 0.90) for their four least-significant IPI-bits. While it appears that ectopic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2015.2496151, IEEE Journal of Biomedical and Health Informatics

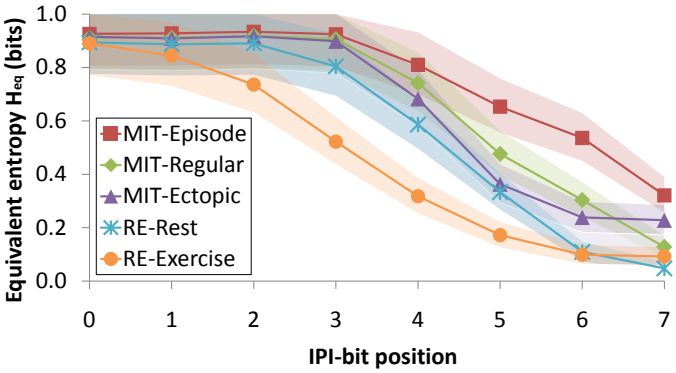


Fig. 3: Entropy per IPI bit H_{eq}^{i} for the considered datasets.

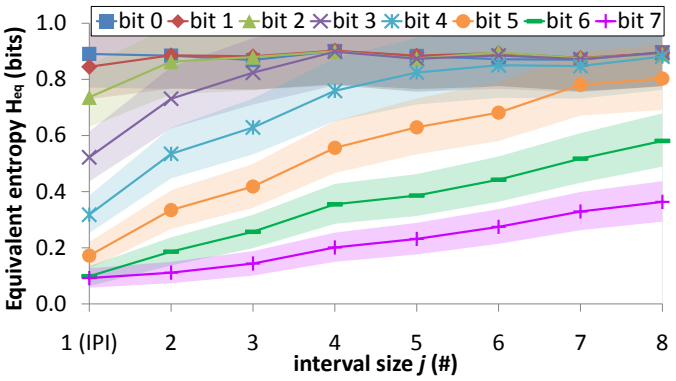


Fig. 4: Entropy per ImPI bit H_{eq}^{i} as a function of interval size j, here depicted for the *RE-Exercise* dataset.

beats result in a slightly lower H_{eq}^{i} compared to a regular heart rate (comparing *MIT-Ectopic* to *MIT-Regular*), we find that the entropy of patients during episodes of arrhythmia is significantly higher (*MIT-Episode*); *2)* Comparing the *MIT-Regular* to the *RE-Rest* datasets shows the effect of a lower sensor-sampling rate. The *RE-Rest* follows the same trend as the *MIT-Regular* dataset, albeit shifted to the left by one bit position, i.e., lowering the sampling frequency reduces the entropy which may be obtained; and *3)* The *RE-Exercise* dataset shows a rapid decrease in entropy from IPI-bit position 1 onwards compared to other datasets, i.e., subjects with limited HRV show a significant reduction in entropy per IPI.

b) ImPI: Let us now consider H_{eq}^{i} for ImPIs as a function of interval size j, as depicted in Figure 4 for the *RE-Exercise* dataset. Recall from Section III that the ImPI is equivalent to the IPI for $j = 1$. Other datasets follow similar trends and will be discussed later in this Section. First, looking at i = 0 (the least-significant ImPI-bit), we observe that H_{eq}^{0} remains at its maximum value of 0.89 bit. As this bit position already contains a strong degree of entropy, increasing the interval size j per ImPI does not increase H_{eq}^{0}. For subsequent bit positions, however, we find that increasing j *does* increase their entropy. Bit position 2, for example, has an entropy H_{eq}^{2} of 0.74 for $j = 1$; 0.86 for $j = 2$; and reaches the "ceiling" of 0.89 bit for $j = 3$. For more significant ImPI-bits, the increase in H_{eq}^{i} is more limited. Regardless of bit position, though, all trends in Figure 4 appear to be monotonously increasing, i.e., increasing j results in an increase in entropy per ImPI.

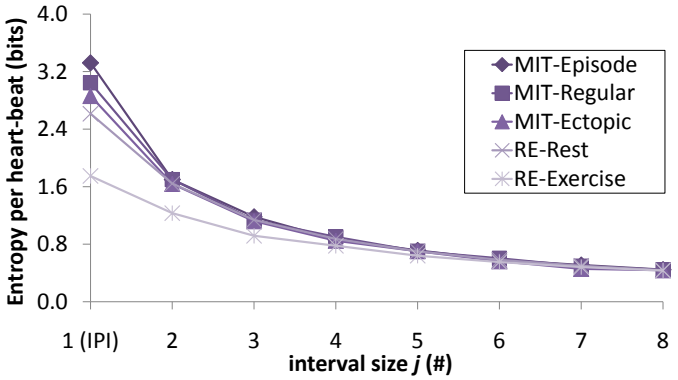


Fig. 5: Entropy per heart beat H_{eq}^{m}/j using ImPI-bit positions 2-5.

The downside of increasing j is that j times more heart beats are required to obtain one ImPI, i.e., less ImPIs may be generated in a given amount of time compared to IPIs. To provide a direct comparison in terms of extraction rate, that is, the entropy extracted per heart beat, we normalize the obtained entropy per I(m)PI by the heart-beat intervals considered (H_{eq}/j). A representative example is provided in Figure 5 for the various datasets, where the key-bit selection m is bits 2-5 of each I(m)PI. For $j = 1$ (IPIs), we find a difference in H_{eq}^{m}/j between the various datasets due to the differences in the entropy H_{eq}^{m} per IPI, as previously shown in Figure 3. By subsequently increasing j, we find that H_{eq}^{m}/j is reduced for all datasets, in particular for the *MIT-** datasets which have a high H_{eq}^{m}/j for $j = 1$. Datasets with high initial entropy (j = 1) cannot benefit from j >1, resulting in progressively lower entropy for increasing j's. Datasets with limited entropy per IPI (*RE-Exercise*, *RE-Rest*), on the other hand, allow for an increase in H_{eq}^{m} when j is increased, resulting in a less dramatic reduction in H_{eq}^{m}/j. Due to this saturation of entropy, we find that H_{eq}^{m} (and, thus, the entropy per ImPI) becomes asymptotically the same as j is increased for all datasets.

2) Hamming-Distance Threshold: We next evaluate the required Hamming-distance threshold T_{HD} for a given true-key pair. First, we consider T_{HD} for an IPI-based key-generator as a function of the key-bit selection m, multi-key authentication and heart rate. Afterwards, we describe the effects on T_{HD} when using the ImPI.

a) IPI: Figure 6 depicts T_{HD} as a function of the key-bit selection m (recall that m is formed by selecting n_m bits per IPI (bpi)) and a-multi-key authentication (described later) for the *MIT-Regular* dataset. When m includes the least-significant IPI bit (starting from bit 0), we find a high value for T_{HD}. This value generally drops when selecting more significant bits for m: As these more significant IPI-bits are less sensitive to VAR_{is}, they contribute relatively little to the disparity between two keys.

Using n_m bpi's implies that the number of IPIs needed to form a 60-bit key is reduced (to $\frac{60}{n_m}$), allowing for multiple authentication attempts to be made within our 60-second authentication-time constraint. We refer to this as multi-key authentication. As we require an entity to authenticate reliably

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2015.2496151, IEEE Journal of Biomedical and Health Informatics

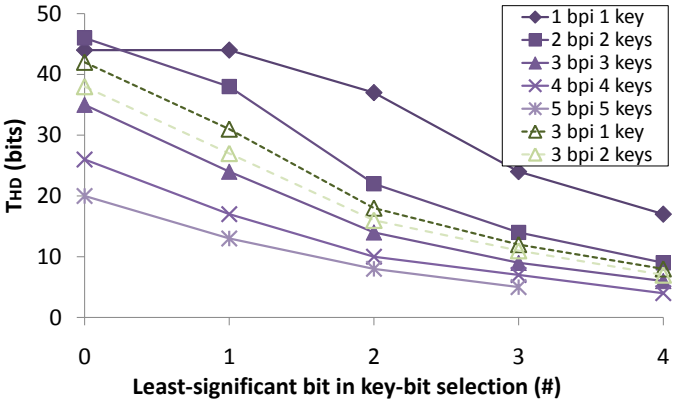


Fig. 6: T_{HD} for the *MIT-Regular* dataset as a function of the bits selected per IPI and multi-key authentication. n_m consecutive bits are selected per IPI (bpi), starting from the IPI-bit position on the x-axis.

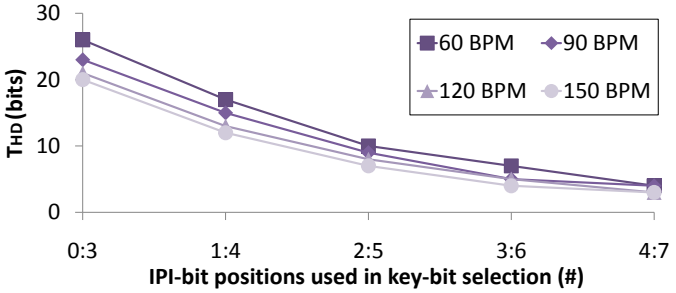


Fig. 7: T_{HD} as a function of heart rate. Four consecutive bits are selected per IPI.

with probability $P_{auth} = 1-10^{-6}$ within 60 seconds, having a attempts results in $P_{auth-key} = 1-\sqrt[a]{10^{-6}}$ for each individual key. In turn, this lowers T_{HD}: To illustrate, Figure 6 also depicts T_{HD} when selecting 3 bpi, where T_{HD} is based on $a = 1$, 2 or 3 authentication attempts (keys). Note that T_{HD} is decreased with increasing values of a.

So far, we have discussed T_{HD} for the *MIT-Regular* dataset, of which the average heart rate is 68.3 BPM. As a higher heart rate implies faster key generation, it may be possible to further decrease T_{HD} as a function of the heart rate by increasing the number of authentication attempts. In practice, an entity could calculate the total time t required to obtain enough IPIs for key generation, derive the possible number of authentication attempts within our authentication-time constraint as $a = \frac{60}{t}$ and base T_{HD} on a-multi-key authentication. To exemplify, Figure 7 depicts T_{HD} for various heart rates, where keys are generated using four bits per IPI. Note that a higher heart rate results in a reduction in T_{HD}. This reduction in T_{HD} is most noticeable when least-significant IPI-bit positions are included in the key-bit selection.

Multi-key authentication does not only benefit T_{HD}: As each key authenticates with a probability of $1-\sqrt[a]{10^{-6}}$, we may improve the average authentication time significantly. For example, using $n_m = 2$, 3 or 4 bpi results in an authentication probability of 99.997%, 99.978% or 99.944% per key, while requiring $\frac{1}{n_m}$ of the key-generation time when n_m = 1 bpi. Table IV presents the average time required to generate a key for our used datasets, based on the number of bits selected.

TABLE IV: Average 60-bit-key-generation time in seconds.

Dataset	Heart rate (BPM)	Bits per IPI (#) 1	2	3	4	5	6
MIT-Regular	68.3	52.7	26.3	17.6	13.2	10.5	8.8
MIT-Ectopic	81.7	44.1	22.0	14.7	11.0	8.8	7.3
MIT-Episode	86.4	41.7	20.8	13.9	10.4	8.3	6.9
RE-Rest	75.7	47.5	23.8	15.8	11.9	9.5	7.9
RE-Exercise	101.4	35.5	17.8	11.8	8.9	7.1	5.9

Obviously, both a higher heart rate and the use of more bits per IPI lead to faster key generation and authentication time.

b) ImPI: Let us now discuss T_{HD} for an ImPI-based key-generator. As with the IPI, each ImPI is calculated as the difference between two heart beats, where the detection of each heart beat is subject to VAR$_{is}$. Our experiments have confirmed that the disparity between two keys is independent from the number of considered heart beats per ImPI j, i.e., T_{HD} is not directly affected by the used heart beats. However, T_{HD} is indirectly affected, as increasing j increases the average key-generation time by a factor j. This reduces the number of keys which may be generated in the 60-second authentication window, leading to an increase in T_{HD} given the discussion on multi-key authentication above. Moreover, certain key-bit selections may no longer be feasible: For example, when $j = 5$ and $n_m = 3bpi$, subjects from the *MIT-Regular* would require an average key-generation time of $5 \cdot 17.6 = 88$ s (see Table IV), exceeding our authentication time constraint.

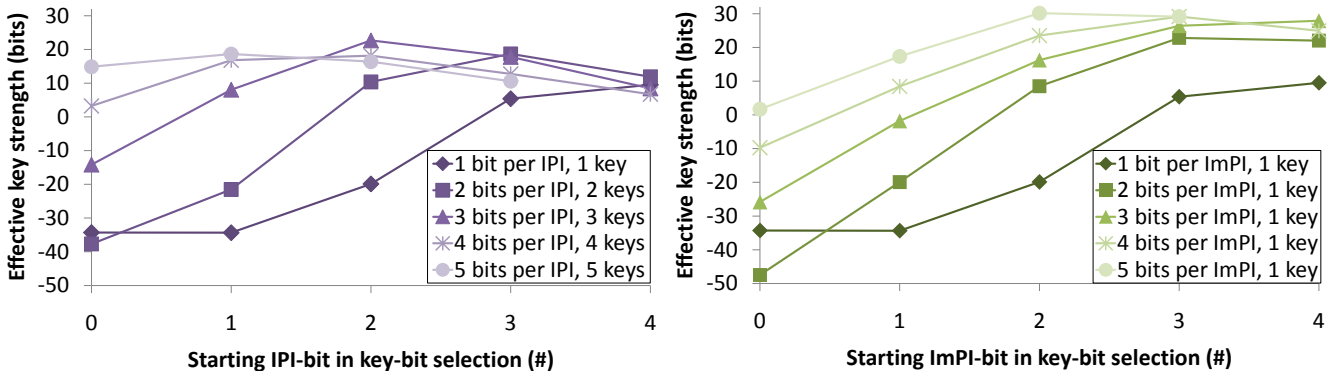
3) Key Strength: Based on the H_k and T_{HD}, we may now calculate the effective key strength KS_{eff}. Here, we calculate H_k based on the accumulation of the entropy of individual I(m)PI-bits included in the key-bit selection (as discussed in Section IV-B1) and base T_{HD} on both the bits selected per I(m)PI, the average heart rate per dataset and multi-key authentication. For all cases, KS_{eff} is evaluated for a 60-bit key and reliable authentication with probability $P_{auth} = 1 - 10^{-6}$ within 60 seconds[4], as described in Section IV-A1. First, we discuss KS_{eff} for IPI-based keys, after which we conclude with the results for ImPI-based keys.

a) IPI: Figure 8a depicts the key strength for the *MIT-Regular* dataset, varying the IPI bits in the key-bit selection[5]. Other datasets yield similar results and are discussed at the end of this Section. First, let us consider KS_{eff} when a single key is generated using 1 bpi (in 52.7 seconds, see Table IV). For bit position 0, we find a negative KS_{eff} of -34.2 bits: While bit 0 contains the most entropy ($H_k = 53.6$ bits), it is also the most strongly affected by VAR$_{is}$ ($T_{HD} = 44$ bits), resulting in a negative KS_{eff}. As the entropy for the 4 least-significant IPI bits is roughly the same for the *MIT-Regular* dataset (see Table III), while VAR$_{is}$ decreases, we find an increase in KS_{eff} up to bit position 4, at which point $KS_{eff} = 9.5$ bits. From bit 4 onwards, KS_{eff} once again

[4]As discussed in Section IV-B2, certain key-bit selections result in a key-generation time significantly smaller than 60 seconds. In these cases, multiple authentication attempts may be made within our authentication constraint of 60 seconds, which effectively decreases T_{HD} and, thus, increases KS_{eff}.

[5]Recall from Section IV-A that a negative key strength ($KS_{eff} < 0$) indicates that an attacker is more likely to authenticate on their first attempt than not, i.e., the generated keys provide practically no security.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2015.2496151, IEEE Journal of Biomedical and Health Informatics



(a) IPI (b) ImPI

Fig. 8: Effective key strength KS_{eff} for the *MIT-Regular* dataset using I(m)PI-based key generation. n_m consecutive bits are selected per I(m)PI, starting from the I(m)PI-bit position on the x-axis. (a) IPI-based key generation, where KS_{eff} is based on n_m authentication attempts; (b) ImPI-based key generation where KS_{eff} is based on a single authentication attempt by setting $j = n_m$, i.e., only one key is generated.

drops: While T_{HD} does decrease for more significant IPI bits, the even steeper decrease in entropy results in negative KS_{eff} scores.

By using multiple bpi's, it becomes possible to generate multiple keys in the same time of generating a single key using 1 bpi – and, thus, perform multiple authentication attempts. This results in an increase in KS_{eff} as may be observed from Figure 8a, which may be attributed to the effect of multi-key authentication on T_{HD}. From all key-bit selections, the maximum KS_{eff} (22.7 bits) is obtained by generating a key using IPI bits 2-4: In this case, T_{HD} is based on 3 authentication attempts, where a single key is generated in 17.6 seconds (see Table IV). That is, reliable authentication using 3 keys is provided in $3 \cdot 17.6 = 52.7$ seconds.

Following the same methodology for all datasets, Table V summarizes the best key-bit selections and resulting key strengths for each dataset. For the *MIT-** datasets, we find a $KS_{eff} \geq 19.1$ bits. The *MIT-Episode* dataset yields a more substantial KS_{eff} = 29.2 bits compared to its counterparts, attributed to the high H_{eq}^{i} which may be found in its most-significant bits. Note that all *MIT-** datasets exclude IPI-bits 0 and 1 from their key-bit selection. Conversely, it may be stated that these sensors are oversampling (by a factor 4) and that a sensor with a 1/4th the sampling rate (90 Hz) would be more than sufficient. For the *RE-Exercise* dataset, we find a maximum $KS_{eff} = 5.7$ bits obtained using IPI bits 1-3, significantly smaller than for the *RE-Rest* dataset ($KS_{eff} =$ 16.4 bits). The reduced entropy per IPI of these former subjects prohibits the generation of strong security keys. Finally, the average key-generation time for each individual key is equal to or less than 17.4 seconds for all datasets, allowing over 99.9% of the authentication attempts to complete within this time as discussed in the previous Section.

b) ImPI: In the previous Sections it was shown that by increasing the interval size j, the entropy per ImPI is increased (i.e., increasing H_k) while less keys may be generated in the same time, increasing T_{HD}. To understand the key strength as a function of j, let us first set the number of selected bits per I(m)PI $n_m = j$. In doing so, only one ImPI key is generated

TABLE V: Best key strength per dataset using the IPI.

Dataset	bits selected	Best KS_{eff} (bit)	Single key-gen time (s)
MIT-Regular	2-4	22.7	17.6
MIT-Ectopic	2-4	19.1	14.7
MIT-Episode	2-6	29.2	8.3
RE-Rest	1-5	16.4	11.9
RE-Exercise	1-3	5.7	11.8

and we exclude the effect of multi-key authentication on T_{HD}. Figure 8b depicts a representative example of this evaluation for the *MIT-Regular* dataset. Similar to IPI-based keys (Figure 8a), we find that the ImPI-key strength is limited when the key-bit selection includes the least-significant ImPI bits and is increased when including more significant bits.

We may now determine the most efficient solution – using multiple IPI keys or a single ImPI key – by comparing the results for IPI and ImPI-based keys in Figures 8a and b. When the key-bit selection includes I(m)PI bit positions 0 or 1, we find that the KS_{eff} of an ImPI-based key is lower than that of an IPI-based key. As discussed in Section IV-B1a, the entropy of these bit positions is high even if j = 1 and is barely increased as a function of j, i.e., H_k does not change significantly. T_{HD}, on the other hand, is increased substantially by lowering the number of generated ImPI-keys, resulting in an overall reduction in KS_{eff}. When the key-bit selection is shifted to more significant bits, we find that an ImPI-based key yields a *stronger* KS_{eff}: While T_{HD} is increased by reducing the number of generated keys, the substantial increases in entropy due to the used bit positions yields a higher KS_{eff}. The strongest ImPI key ($KS_{eff} = 30.2$ bits) is obtained using ImPI bits 2-7 and provides reliable authentication within 52.7 seconds. Under the same authentication constraints, the strongest IPI key (discussed before) has a more limited key strength of $KS_{eff} = 22.7$ bits. ImPI-based keys may, thus, achieve a higher key strength than IPI-based keys.

Finally, by varying both j and the selection of bits (j does not necessarily equal n_m), we derive the best possible KS_{eff} for each dataset, as presented in Table VI. In line with our previous conclusions, we find that the datasets which

TABLE VI: Best key strength per dataset using the ImPI, compared to the strongest IPI-based keys.

Dataset	interval size (j)	bits selected	Best KS_{eff} (bit)	Single key-gen time (s)
MIT-Regular	6	2-7	30.2 (+33%)	52.7 (3.0x)
MIT-Ectopic	6	2-6	26.6 (+39%)	52.9 (3.6x)
MIT-Episode	4	3-6	29.8 (+2%)	55.5 (6.7x)
RE-Rest	5	2-6	31.3 (+92%)	47.5 (4.0x)
RE-Exercise	8	2-6	24.9 (+3.4x)	56.8 (4.8x)

already contain a high degree of entropy do not benefit much from using the ImPI, most notably the *MIT-Episode* dataset. For datasets with lower entropy, however, we find substantial increases in the key strength, up to KS_{eff} = 24.9 bits (+3.4x compared to the optimal IPI-bit selection) for the *RE-Exercise* dataset. That is, when the entropy per IPI is limited, the ImPI provides stronger security than IPI-based keys. It is interesting to observe that when using the ImPI, all datasets shift their key-bit selection to the more significant bits per ImPI, taking advantage of the increase in H_{eq} and minimal increase in T_{HD}. Finally, note that while these keys are generated within our authentication-time constraint of 60 seconds, we do find a substantial increase in key generation time between 3-6.7x.

V. Acknowledgement

This work has been supported by the EU-funded projects DeSyRe (Grant agreement no: 287611) and SHARCS (Grant agreement no: 644571) and would not have been completed without the feedback of dr. Foteini Agrafioti.

VI. Conclusion

This paper has presented a thorough evaluation of the security performance of a heart-beat-based-security system which uses IPI as a source of entropy, considering the effects of (limited) HRV, sensor-sampling frequencies, VAR_{is} and multi-key authentication. In addition, we have introduced a novel key-generator based on the Inter-*multi*-Pulse Interval (ImPI), which considers the time interval between two non-consecutive heart beats. It was shown that while successful authentication may occur within 17.4 seconds for an IPI-based key-generator, the effective key strength may be as low as 5.7 bits for subjects with limited HRV. This key strength was successfully increased by up to 3.4x (+19.2 bits) through using the ImPI-based key generation, at the cost of an increase in key-generation time of 4.8x (from 11.8 sec to 59.8 sec). That is, using the ImPI in key generation results in stronger keys than using the IPI, given the same authentication time. In order to maximize the security of heart-beat-based systems, future security protocols should consider the possibility of dynamically adjusting the key-generation settings, as revealed by this work.

References

[1] U. R. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri. Heart rate variability: a review. *Medical and biological engineering and computing*, 44(12):1031–1051, 2006.

[2] F. Agrafioti, F. M. Bui, and D. Hatzinakos. Medical biometrics in mobile health monitoring. *Security and Communication Networks*, 4(5):525–539, 2011.

[3] I. Antelmi et al. Influence of age, gender, body mass index, and functional capacity on heart rate variability in a cohort of subjects without heart disease. *AJC*, 93(3):381–385, 2004.

[4] B. M. Appelhans and L. J. Luecken. Heart rate variability as an index of regulated emotional responding. *Review of general psychology*, 10(3):229, 2006.

[5] 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.

[6] L. Biel, O. Pettersson, L. Philipson, and P. Wide. Ecg analysis: a new approach in human identification. *Instrumentation and Measurement, IEEE Transactions on*, 50(3):808–812, 2001.

[7] 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. In *Proceedings of the USENIX conference on Health Security and Privacy*, pages 5–5, 2012.

[8] T. Denning et al. Absence makes the heart grow fonder: new directions for implantable medical device security. In *HotSec*, 2008.

[9] T. Denning et al. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *SIGCHI*, pages 917–926, 2010.

[10] A. L. Goldberger et al. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[11] T. Hong et al. An improved scheme of ipi-based entity identifier generation for securing body sensor networks. In *IEEE EMBC*, pages 1519–1522. IEEE, 2011.

[12] N. Iyengar et al. Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics. *AJP-Regu*, 271(4):R1078–R1084, 1996.

[13] A. K. Jain, R. Bolle, and S. Pankanti. *Biometrics: personal identification in networked society*. Springer Science & Business Media, 1999.

[14] M. Li et al. Data security and privacy in wireless body area networks. *Wireless Communications, IEEE*, 17(1):51–58, 2010.

[15] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Eng Med Biol*, 20(3):45–50, 2001.

[16] K. N. Plataniotis, D. Hatzinakos, and J. K. Lee. Ecg biometric recognition without fiducial detection. In *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pages 1–6. IEEE, 2006.

[17] 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.

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

[19] 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.

[20] R. M. Seepers et al. Adaptive entity-identifier generation for imd emergency access. In *ACM CS2*, pages 41–44, 2014.

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

[22] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[23] T.-W. Shen, W. Tompkins, and Y. Hu. One-lead ecg for identity verification. In *IEEE EMBS/BMES Annual Cofnerence and Fall Meeting*, volume 1, pages 62–63. IEEE, 2002.

[24] M. P. Tulppo, T. Makikallio, T. Takala, T. Seppanen, and H. V. Huikuri. Quantitative beat-to-beat analysis of heart rate dynamics during exercise. *American Journal of Physiology-Heart and Circulatory Physiology*, 271(1):H244–H252, 1996.

[25] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. Pska: usable and secure key agreement scheme for body area networks. *ITB, IEEE Trans. on*, 14(1):60–68, 2010.

[26] J. von Neumann. Various techniques used in connection with random digits. *Monte Carlo Method, National Bureau of Standards Applied Math*, pages 36–38, 1951.

[27] F. Xu et al. Imdguard: Securing implantable medical devices with the external wearable guardian. In *INFOCOM*, pages 1862–1870. IEEE, 2011.

[28] 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.