Future Generation Computer Systems 105 (2020) 346–360

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Are the Interpulse Intervals of an ECG signal a good source of entropy? An in-depth entropy analysis based on NIST 800-90B recommendation ☆

Lara Ortiz-Martin [b], Pablo Picazo-Sanchez [a,*], Pedro Peris-Lopez [b]

[a] Chalmers, University of Gothenburg, Sweden
[b] Universidad Carlos III de Madrid, Spain

ARTICLE INFO

Article history:
Received 12 June 2019
Received in revised form 11 October 2019
Accepted 1 December 2019
Available online 10 December 2019

Keywords:
Entropy
NIST 800-90B
Security
Privacy

ABSTRACT

In recent years many authors have explored the use of biological signals for security issues. In the context of cardiac signals, the use of Inter-Pulse Interval *(IPI)* values as a source of entropy is one of the most widely used solutions in the literature. To date, there is a broad consensus that the four least significant bits of each IPI are highly entropic and can be used, for instance, in the generation of a cryptographic key. In this article, we demonstrate that the choice of the IPI bits used to date may not be the most correct (e.g., the combination of bits 2638 are much better that the common assumed 5678). To come to our conclusions, we have done a rigorous and in-depth study, analyzing cardiac signals from more than 160,000 files from 19 databases of the Physionet public repository and basing our analysis on the NIST 800-90B recommendation.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In the last years, a new way of generating and distributing secret tokens based on the heart signal has gained more and more popularity among security researchers [1,2]. It can be seen how since the first paper appeared in 2004, proposing that the heart signal might be applied to cryptography [3], several proposals have been published in the literature.

In brief, the heart signal—which is a continuous signal—is gathered by some sensors, and it is transformed into a discrete signal. This process is known as *quantization*. While the first algorithm was introduced by Bao et al. [3] and later improved by Xu et al. [4] in 2011, the most common one was proposed by Rostami et al. [5] two years later based on the previous ones. After then, many authors have used such quantization algorithm [6–9] or a slight modification of it [10] to extract a subset of the *Least Significant Bits (LSBs)* from each *Inter-Pulse Interval* (IPI) (i.e., time interval between two R-peaks or heartbeats) due to its claimed entropy property [5].

In a vast majority of the literature, authors rely either directly or indirectly—by referencing other papers, on the fact that the heart signal contains entropy and thus, it might be used in key generation procedures [8,10], authentication protocols [5,11,12] or peak misdetection algorithms [7,13]. There is, however, a standard methodology in all these works based on IPI values: the length of the generated token is given by appending as many bits (typically the four LSBs per Inter-Pulse Interval *(IPI)*) as the protocol needs. On the contrary, some authors do not follow this line but claim that the *Most Significant Bit (MSB)* of the IPIs do not have entropy [14]. In this paper, we will demonstrate that MSBs should also be taken into account to generate tokens with entropy.

When authors check the entropy of the generated tokens, there is a subset of them who specifically claim to use the Shannon entropy [6,8,9,15]. On the contrary, there are others who just say that they test the entropy, providing no more information [10,16] or even there are some authors who directly do not check the entropy but run some random test instead like the *National Institute of Standard and Technology Statistical Test Suite (NIST STS)* [17–19] which, as Rushanan et al. [20] pointed out, is not enough to claim that the *Electrocardiogram (ECG)* can be a good source of entropy.

Nevertheless, to the best of our knowledge, some questions have not been tackled in the literature so far. (1) Are the four LSBs the best ones to create the best token from the entropy point of view? (2) Are there any other possible combinations of bits that achieve more entropy than taken the four LSBs? How good

☆ This work was partially supported by the Swedish Research Council (Vetenskapsrådet) under grant Nr. 2015-04154 (PolUser: Rich User-Controlled Privacy Policies), by the MINECO grant TIN2016-79095-C2-2-R (SMOG-DEV), and by The Leonardo Grants for Researchers and Cultural Creators awarded by the BBVA Foundation.

* Corresponding author.
E-mail addresses: laortizm@inf.uc3m.es (L. Ortiz-Martin), pablop@chalmers.se (P. Picazo-Sanchez), pperis@inf.uc3m.es (P. Peris-Lopez).

https://doi.org/10.1016/j.future.2019.12.002
0167-739X/© 2019 Elsevier B.V. All rights reserved.

they are concerning the four LSBs, and; (3) Is the ECG a source of entropy?

In this article, we answer these questions and demonstrate that only by looking at the Shannon entropy is not enough for the heart signal—and particularly for the IPI values from the ECG signal—to be considered a source of entropy.

1.1. Contributions

In this work, we analyze the entropy of the LSB values extracted from a heart signal according to the *National Institute of Standard and Technology (NIST)* recommendation (i.e., SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation [21]). In theory, the 4 LSBs are the ones with more entropy. However, we followed a methodology based on IPI extraction [5] and concluded that the last bits could not be the best choice one from the entropy point of view.

Traditionally, the way researchers measure the entropy of a given position is by computing as many IPIs as possible and generating a string composed of bits from the same position in the IPI. We argue that this could not be the best strategy. In detail, we show how taking more bits than the 4 LSBs does not decrease the entropy and makes it possible to extract more information from each heartbeat. This property has a direct impact on the performance: the more information we can extract from each heartbeat, the less time the system needs to generate tokens of a given size. Therefore, it makes our proposal faster than those that only use the 4 LSBs.

To facilitate the replication of our results by other researchers, we downloaded and used 19 databases from the Physionet public repository[1] [22]. Our contributions can be summarized as:

- We test 19 databases with information about heart signals from different people. All datasets are taken from the Physionet public repository, which contains heart signals from both healthy volunteers and people with cardiac conditions.
- Contrarily to prior proposals, we demonstrate that the four LSB are not the best bits to be used in cryptographic applications. We generate all the variation without repetition of 8 bits taken from [2, …,5] bits and extract the best combination of bits—combination(s) which achieve(s) the best results in the min-entropy estimators [21]. To the best of our knowledge, this is the first work that aims at extracting the best combination of bits of the IPIs in terms of min-entropy.
- We empirically analyze more than 160,000 files and propose different combinations for extracting tokens by taking 2, 3, 4 and 5 bits which are, in general, much better than taking the 4 LSBs.

The rest of the paper is organized as follows: Section 2 provides some background on biometric authentication using an ECG signal and also a necessary explanation of some random tests. Section 3 describes the evaluation of our implementations and a discussion of the results. A description of the most relevant contributions in this area is summarized in Section 4 while this paper ends with some conclusions and future work in Section 5.

2. Background

2.1. Dataset and IPI extraction

Dataset. We first downloaded 19 databases from the Physionet repository [22] which contain the heart information of several subjects/patients and their ECG signals in one or several channels.

Table 1
For each database the number of patients and the pathology (if any) of the patients involved.

Database	#Patients	Pathology
afdb [23]	23	Atrial fibrillation
afpdb [24]	300	Paroxysmal atrial fibrillation
cebsdb [25]	60	Healthy volunteers
edb [26]	90	Myocardial and hypertension
fantasia [27]	40	Healthy
iafdb [28]	32	Atrial fibrillation or flutter
incartdb [29]	75	Coronary artery disease
ltafdb [30]	84	Paroxysmal
mitdb [31]	48	Arrhythmia
nsrdb [32]	18	No significant arrhythmias
nstdb [33]	15	Mitdb with noise
prcp [34]	10	Healthy
qtdb [35]	105	Holter recordings
sddb [36]	22	Arrhythmia
shareedb [37]	139	Hypertension
slpdb [38]	18	Sleep apnea syndrome
svdb [39]	70	Partial epilepsy
twadb [40]	100	Myocardial problems
vfdb [41]	22	Tachycardia

In these datasets, we can find from healthy people as in cebsdb to patients with myocardial diseases as in edb. Table 1 shows a summary of the main characteristics of the 19 datasets used all throughout this work.

Inter-Pulse Interval *(IPI) extraction.* The time distance between R-peaks (heartbeats) is one of the essential features for cryptography that the ECG has. This time is usually known as *Inter-Pulse Interval* (IPI), and it is particularly interesting because most of the proposed works in this area found out that the four LSBs of each IPI have some entropy [4,5]. There are, on the contrary, some authors who use more bits than the four LSBs [14,42]. Contrarily what it is usually assumed, in this work we empirically demonstrate that taking more than 4 bits might be a good strategy from the entropy point of view and we give the better combination of bits to generate a high entropic sequence based on IPI values (see Section 3) than the usual 4 LSBs.

To process and extract the IPIs from the Physionet repository, we used some scripts provided by them.[2] These scripts were used to obtain the ECG signal from each patient of the 19 databases. After that, we run the well-known Pan–Tomkins's algorithm [43] over the ECG signal to extract the R-peaks. Once we extracted the time intervals, we calculated the difference between each pair of consecutive R-peaks to obtain the so-called IPI values.

Once we computed the IPIs, we run the *quantization* algorithm proposed by Rostami et al. [5], which is a slight variation of the quantization algorithm first proposed by Bao et al. [3] and later improved by Xu et al. [4]. This algorithm fundamentally transforms a continuous signal into a discrete one and applies a Gray code to decrease the errors of the signal.

We took the public source code recently released by Ortiz et al. [44] and made some slight modifications to get all the IPIs. This task was particularly computational demanding due to the amount of IPIs to generate and the number of databases involved in the experiment.

2.2. Entropy & NIST

The concept of entropy was first introduced in 1948 by Shannon in [45]. Roughly speaking, when applied to information theory, entropy measures how probable an event may occur given all possible events, that is, if the frequency of an event is so high,

[1] https://physionet.org/physiobank/database/#ecg.

[2] https://physionet.org/physiotools/software-index.shtml.

then the information entropy is low [46]. On the contrary, if an event only occurs some times, it is said that it contains more information, and thus, the information entropy is high. More formally, the entropy is defined as the negative logarithm of the probability mass function for the value: $H(X) = -\sum_i^n P_i log P_i$. This measurement of information entropy has been widely used in the literature to verify, in our case, how good or bad a heart signal is from the entropy point of view. In other words, if a heart signal can be used as a source data generator due to its entropy, i.e., if the heart can generate numbers with high entropy, it means that such a signal might be used to generate random numbers. However, by using just the Shannon entropy to claim that a source can be considered random or not is not enough. Let us propose the following sequence of bits "10101010". If someone calculates the Shannon entropy, which is $H(X) = 1$ and does not perform any other entropy tests, she might reach to the wrong conclusion that such a sequence is highly entropic. However, it is quite clear that such a sequence follows a pattern and thus, is far from being a random sequence (see Table 3 to see the complete example).

In 2012, the NIST published a draft with some recommendations for the entropy sources used for random bit generation [47]. The final document (NIST SP 800-90B) was recently published—early 2018—and can be seen in [21]. This document introduces the minimum properties that an entropy source must have to make it suitable for use by cryptographic random bit generators, as well as the *min-entropy* which represents the minimum value after executing a set of tests (estimators) used to validate the quality of the entropy source. Note that the min-entropy value is never higher than the Shannon entropy.

It is important to remark the difference between the NIST min-entropy and the one used in information theory which is a specific case of Rényi's entropy. In the former, uncertainty is measured in terms of a random variable's vulnerability to being guessed in one try by an adversary [48]. This last concept has been recently used by Chizari and Lupu [49] to measure the entropy of the heart signal.

Regarding the size of the dataset, the NIST SP 800-90B recommendation suggests that there is a minimum number of bits that should be used to test the data source. Concretely, authors indicate that "a sequential dataset of at least 1,000,000 consecutive sample values obtained directly from the noise source" is needed. Nevertheless, if this constraint cannot be satisfied, they also contemplate the option of taking small pieces of, at least 1000 samples to create a dataset of 1,000,000 values if all these chunks come from the same data source to be evaluated.

In Table 2, we can find a summary of the size of the databases. In that table, we can split databases up into two main groups: (1) databases that achieve more than 10^6 bits in the generated files, and; (2) databases that do not achieve such threshold. Having that in mind, results regarding databases that do not achieve such a threshold should be taken with a pinch of salt. Despite that, we decided to keep them in the analysis due because many works consider them (e.g., mitdb or qtdb) in their experiments [5,14,50,51].

In our work we are using the min-entropy estimators proposed by the NIST to check if the bit sequences extracted from the heart signal, pass such estimators or not and thus we can consider the heart as entropy data source. The execution of each one of these estimators gives as a result an entropy value which is independent from the others estimators. Finally, the algorithm outputs the minimum value of all the estimators, i.e., min-entropy.

In the following, we describe in more detail the ten proposed estimators to compute the min-entropy by the NIST.

Table 2
For each database, ✗ the denotes whether the size of the generated IPIs is less than 10^6, and; ✓ means that the size is larger than 10^6.

Database	2 bits	3 bits	4 bits	5 bits
afdb	✓	✓	✓	✓
afpdb	✗	✓	✓	✓
cebsdb	✗	✗	✗	✗
edb	✓	✓	✓	✓
fantasia	✓	✓	✓	✓
iafdb	✗	✗	✗	✗
incartdb	✗	✗	✗	✗
ltafdb	✓	✓	✓	✓
mitdb	✗	✗	✗	✗
nsrdb	✓	✓	✓	✓
nstdb	✗	✗	✗	✗
prcp	✗	✗	✗	✗
qtdb	✗	✗	✗	✗
sddb	✓	✓	✓	✓
shareedb	✓	✓	✓	✓
slpdb	✗	✓	✓	✓
svdb	✗	✗	✗	✗
twadb	✗	✗	✗	✗
vfdb	✗	✗	✗	✗

Table 3
Example of min-entropy results using: a 10^6 bits file composed of the sequence "10" (len("10") = 10^6); the first 10^6 bits of π, and; the first 10^6 bits of the output of the `urand` function.

Estimator	len("10") = 10^6	π	`urand`
Most_common	0.99	0.8	1.0
Collision	1.0	0.56	0.93
Markov	0.007	0.72	0.99
Maurer_universal	0.0	0.60	0.84
MultiMCW	0.0	0.81	0.99
Lag	0.0	0.81	0.98
MultiMMC	0.0	0.81	0.99
LZ78Y	0.0	0.81	0.99
t_tuple	0.0	0.70	0.91
LRS	0.0	0.93	0.99

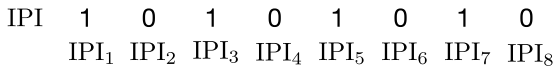


Fig. 1. Position of bits in IPIs.

The Most Common Value Estimate This test first finds the proportion p of the most common value in the dataset, and then constructs a confidence interval for such p.

The Collision Estimate This test is based on [52], and the goal is to estimate the probability of the most-likely output value, based on the collision times—the number of repeated values. This test outputs a low entropy estimate for sources that have a significant bias toward a particular output or value (i.e., the average time to a collision is relatively short) while a higher entropy estimate occurs for a longer average time to collision.

The Markov Estimate This method generates a min-entropy estimate by measuring the dependencies between consecutive values from the dataset. This test is used to test sources with dependencies in the dataset.

The Compression Estimate This test computes the entropy rate of a dataset based on how much the dataset can be compressed. This test is based on the Maurer Universal Statistic [53], and it is computed by generating a dictionary of values, and then computing the average number of samples required to produce an output, based on that dictionary.

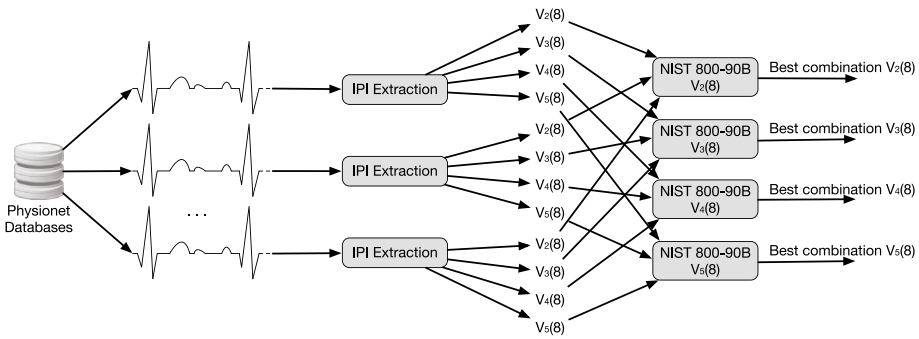


Fig. 2. Scheme of the methodology for the experiments where, for each database we extract the IPIs and generate variations of k items, k = 2...5, from 8 elements and run the NIST 800-90B suite to obtain the combination(s) with highest entropy.

The MultiMCW Prediction Estimate This test is composed of multiple *Most Common in Window (MCW)* sub-predictors, each of which aims to guess the next output, based on the last n outputs. This is done by each sub-predictor, which extracts the most often value in that window of n outputs. This test was designed for cases where the most common value changes over time but remains relatively stationary over reasonable lengths of the dataset.

The Lag Prediction Estimate Similar to the MCW, this test has several sub-predictors, each of which predicts the next output based on a so-called *lag*. This method keeps a counter of the number of times that each sub-predictor was correct and uses the best sub-predictor to predict the next value.

The MultiMMC Prediction Estimate The MultiMMC predictor is composed of multiple *Markov Model with Counting (MMC)* sub-predictors. Instead of keeping the probability of a transition like in a Markov model, the predictors of this test record the observed frequencies for transitions from one output to a subsequent output and makes a prediction based on the most frequently observed transition from the current output.

The LZ78Y Prediction Estimate The LZ78Y predictor is loosely based on LZ78 encoding with the Bernstein's Yabba scheme [54] for adding strings to the dictionary. The predictor keeps a dictionary of strings that have been added to the dictionary so far and continues adding new strings to the dictionary until the dictionary has reached its maximum capacity.

The t-Tuple Estimate This method checks the frequency of t-Tuples, i.e., pairs, triples, etc., that appears in the dataset. It produces an estimate of the entropy per sample based on the frequency of those t-tuples.

The *Longest Repeated Substring (LRS)* Estimate This test estimates the collision entropy (sampling without replacement) of the dataset based on the number of repeated tuples within the input dataset.

We carried out one experiment to help readability and understanding of both, the min-entropy estimators, as well as the results presented throughout this article. We generated: (1) a file of 10^6 bits length repeatedly composed of the string "10"; (2) a file made of the first 10^6 of π, and; (3) a file created of 10^6 bits after running the `urand` function. The results can be seen in Table 3. As a final output, the min-entropy represents the minimum value of all the above estimators, i.e., 0.0, 0.56 and 0.84 respectively, which confirms that only the `urand` seems a good source of entropy.

3. Entropy evaluation of IPIs

In this section, we describe the experiments we carried out to analyze in-depth the entropy quality of IPI values derived from an ECG signal. That is, we show whether IPIs are a good source of randomness. We explain below, in general terms, the methodology used for the analysis of the nineteen datasets.

For all the experiments shown all along this section, we used the same procedure. We first applied the quantization algorithm[3] to extract the IPIs. After that, we generated the variations without repetition of 2, 3, 4 and 5 bits, i.e., we produced $V_k(n) = \frac{n!}{(n-k)!}$ files where n is the length of the IPIs—8 bits, and k is the number of bits. We, therefore, generate 56, 336, 1680 and 6720 files respectively per database.

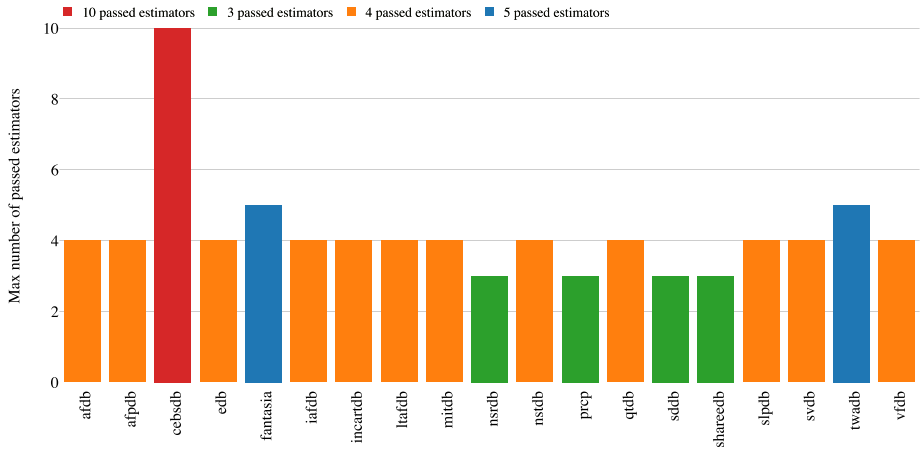
Fig. 1 shows an example of an IPI and the notation we use to refer to how a file is made of. In that Figure as well as for the rest of the paper, the 1st bit (IPI_1) is the Most Significant Bit (MSB) whereas the last one (IPI_8) is the Least Significant Bit (LSB). For example, when we say "bits 268", "a combination of bits 268", or just "268", we refer that a file is generated by a concatenation of bits placed in the 2nd, 6th and 8th positions of the IPIs (i.e., $IPI_2||IPI_6||IPI_8$) belonging to a concrete database, e.g., "000" in the example of Fig. 1. Another example might be the combination of bits 78 (i.e., $IPI_7||IPI_8$), which can also be read as the file made of the last 2 LSB of IPIs, e.g., "10" in the example of Fig. 1.

After running the min-entropy (using ten estimators) in our previous example against three well-known examples (see Table 3), we assume and without loss of generality, that an estimator is successful when the entropy is higher than 0.7. Note that we impose this threshold and depending on the security application, it might be more or less restrictive. In Section 3.5, we show the results we would have got by using a threshold of 0.9 which is the threshold we have observed in many scientific papers for a sequence to be considered entropic or not [6,8,14,50,55]. Fig. 2 summarizes the methodology we followed for the execution of the experiments.

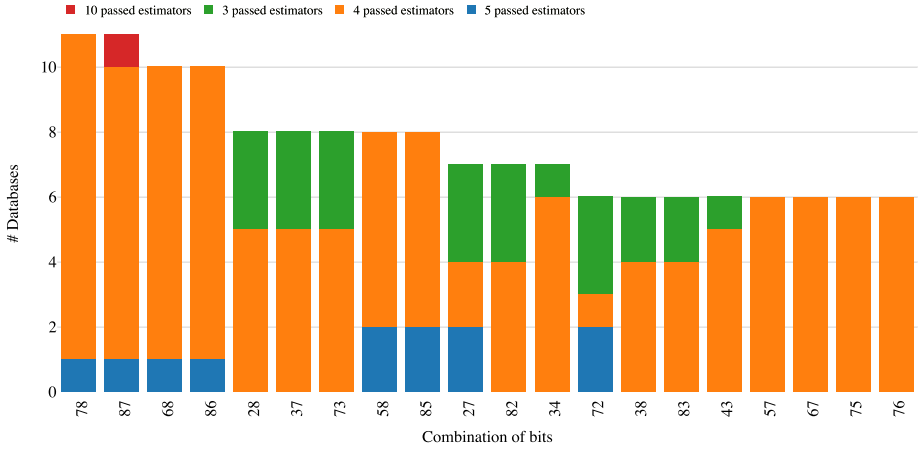
We followed the same methodology for carrying out all the experiments as well as for analyzing the results after running the min-entropy estimators provided by the NIST SP 800-90B recommendation. First, we obtained the maximum number of estimators (e.g., Collision and Markov estimators are higher than the specified threshold) that a combination of bits may achieve, i.e., for each database, we computed all the variations without repetitions and selected the best combination(s) that passes the maximum number of min-entropy estimators.

Second, and using the thresholds computed previously (maximum number of passed estimators), we grouped all the

[3] See [5,44] for more details about the quantization algorithm as well as for the source code.



(a) Number of passed estimators



(b) Best combinations.

Fig. 3. Entropy analysis of files generated by extracting 2 bits from IPIs. Fig. 3a represents the maximum number of passed estimators that achieves at least one combination of bits. Fig. 3b shows the best and most common combination of bits of databases.

databases, and for each combination of bits, we counted the number of databases that achieves these thresholds. All this information is displayed in a figure in which each column represents a histogram. With this experiment, we can: (1) corroborate one of the main differences between the min-entropy estimators and the classical Shannon entropy: e.g., the order in which the entropy source generates the random sequence matters; (2) obtain a detailed list of the best and most common combination of bits to be chosen for different length of the IPIs, and; (3) compare the results we got with the combination that it is usually used in the IPI-based papers.

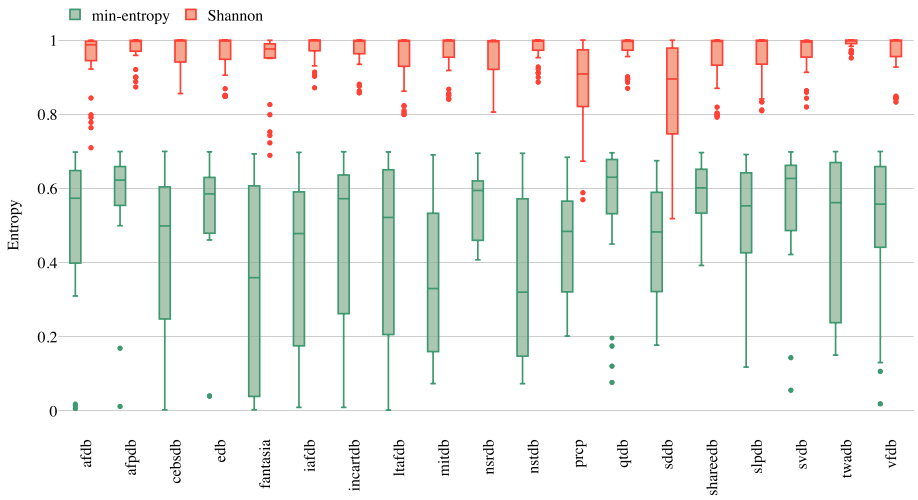
It is remarkable that, given the high demanding operations, we had to implement a few strategies to speed up and to improve the performance of the experiments. In particular we: (i) executed the estimators sequentially (following the same order we introduced them in Section 2.2) until we found an estimator that failed it and we stopped the execution, and; (ii) we introduced a slight modification to the min-entropy python project provided by the NIST in such a way that we only take the first 10^6 characters from the generated files. This patch allowed us to speed up considerably the last two estimators which are the most computationally demanding (i.e., t-Tuple and LRS estimators). According to our estimations, if we would not have done that, executing all the min-entropy estimators to all the 19 to all the variations without repetition for 8 elements taking from 2, 3, 4 and 5 bits would have taken us more than one year of computing these results in a Linux based cluster with 16 CPUs and 40 Gb of RAM.

Besides, we carried out another experiment to see the differences between the Shannon entropy versus the min-entropy tests. The goal of this experiment is to demonstrate that, only by using the Shannon entropy is not enough to claim that a source can be entropic or not. In particular, for this test, we took the minimum value after running all the estimators (as suggested by NIST) as the min-entropy, and we additionally computed the Shannon entropy for all the variations without repetition for each one of the databases. We repeated this experiment for bits length from 2 to 5.

Finally, we generated heatmaps to see the number of failed estimators per database. These results will shed some light on the weaknesses of the bit streams generated. It is essential to remark two main things: 1. the heatmaps do not show the combinations of databases that passed all the estimators, and; 2. estimators are executed sequentially, and that is why in the heatmaps there are estimators with no numbers.

3.1. $V_2(8)$ *variations of two bits without repetition*

We analyzed the results after running the min-entropy estimators for variations without repetition of 2 bits and the results are shown in Figs. 3 and 4. In average, it can be seen how in most of the databases, the maximum number of successful tests



(a) Shannon entropy vs min-entropy tests.

	most_common	collision	markov	maurer_universal	MultiMCW	Lag	MultiMMC	LZ78Y	t_tuple	LRS
afdb	14	19	2	16	5					
afpdb	14	27	0	11	4					
cebsdb	14	0	0	4	12	2	0	0	17	6
edb	14	15	0	23	4					
fantasia	18	4	2	4	22	6				
iafdb	14	15	0	1	20	0	0	0	6	
incartdb	14	15	0	13	14					
ltafdb	14	12	0	16	14					
mitdb	14	4	0	8	30					
nsrdb	14	29	0	13						
nstdb	14	4	0	6	32					
prcp	40	2	8	6						
qtdb	14	14	0	19	9					
sddb	38	4	4	10						
shareedb	14	28	0	14						
slpdb	14	8	0	24	10					
svdb	14	17	0	21	4					
twadb	6	6	0	13	21	10				
vfdb	14	12	0	18	12					

(b) Failed estimators.

Fig. 4. Entropy analysis of files generated by extracting 2 bits from IPIs. Fig. 4a depicts a comparison of the min-entropy and the Shannon entropy. Fig. 4b shows a heatmap of the most failed estimators per database.

(estimator higher than 0.7) is four (see Fig. 3a). In the case of both fantasia and twadb there is at least a combination of bits that passes five estimators of the min-entropy test while in the case of nstdb, prcp, sddb and shareedb there is at least one combination that passes three estimators at the same time. Finally, cebsdb is the only one where at least one combination passes all the estimators at a time. In particular, this combination is the one composed of the two LSBs in the inverse order, i.e., 87. It is noticeable that this combination is represented in Fig. 3b.

In addition to that, from such a plot, we conclude that any of the permutations of the last 2 LSBs are the best one to be chosen if only 2 bits are picked as entropy source (i.e., in 11 out of 19) have such combinations as the best ones) followed by the permutation of the bits 8 and 6 (common in 10 out of 19).

We also tried to find some correlation between the composition of the databases without success. For instance, the set of healthy databases is composed of {cebsdb, fantasia, nsrdb, prcp} and the combination of bits 78 is not considered to be the best option in any of them. On the contrary, the combination of bits 87 is the best one only in the cebsdb database.

We created a boxplot in Fig. 4a to show a comparison of using the Shannon entropy versus the min-entropy. From such a plot, it is quite clear to see the difference between these tests. Therefore, and under the NIST SP 800-90B recommendation, there are no databases which might be considered suitable as a good source of randomness from the entropy point of view.

Finally, Fig. 4b depicts the estimators where databases fail with most frequency. It is interesting to see that there is only one record in the cebsdb database that passes all the estimators at a time (note that the sum of all the numbers in the cebsdb row gives 55 and the $V_2(8) = 56$). Particular attention should be put in both prcp and sddb, where a majority of the records fail in the first estimator (most_common) which indicates that the sequences of bits are clearly not balanced, i.e., there are more 1's than 0's or the other way around. In general, most of the databases stop their execution after running the MultiMCW estimator which means that, despite the number of symbols (1's and 0's) vary over the sequence, that difference is in fact not that much and thus it is relatively easy to predict.

3.2. $V_3(8)$ variations of three bits without repetition

We generated the $V_3(8)$ and obtained 336 files per database. After that, we executed the min-entropy estimators to each one of the files to obtain the maximum number of passed estimators. The results can be seen in Fig. 5a. In particular, we can assort

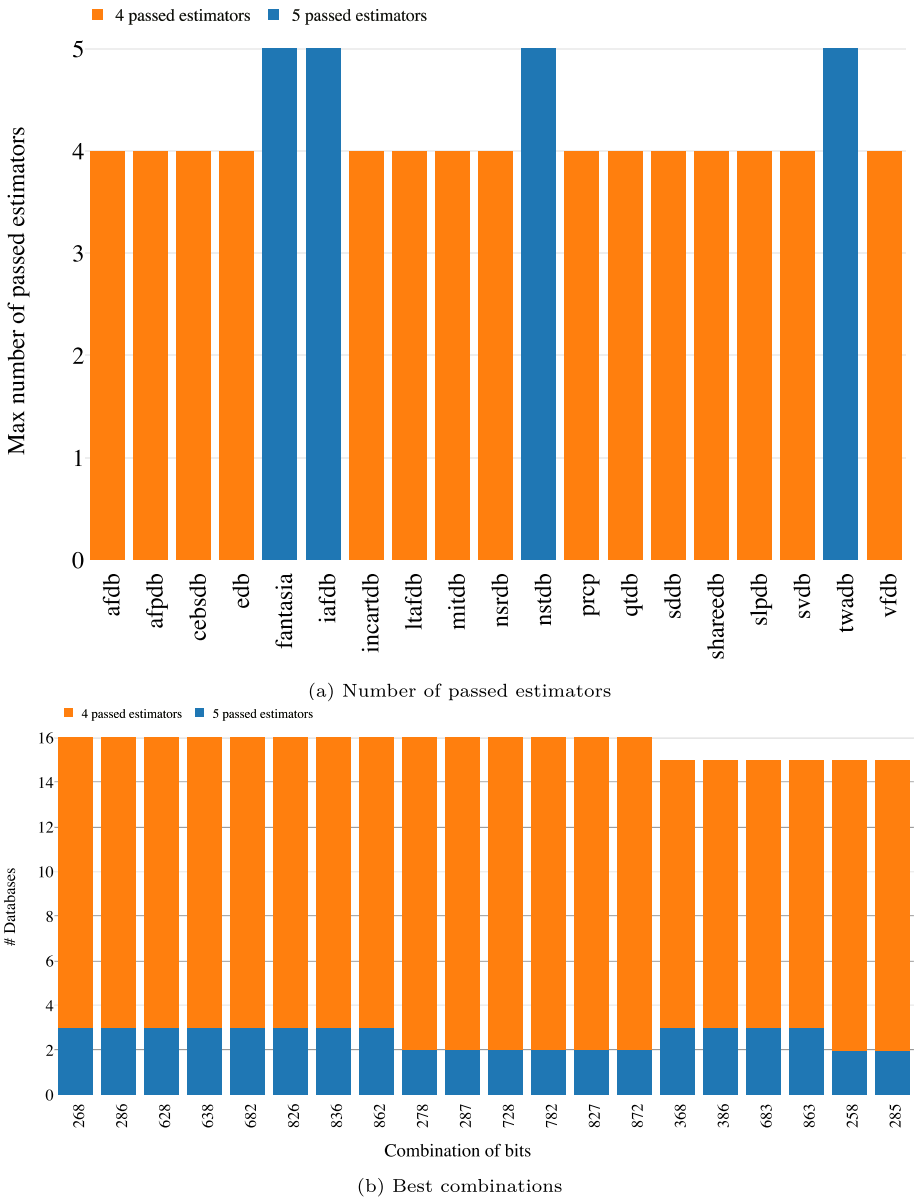


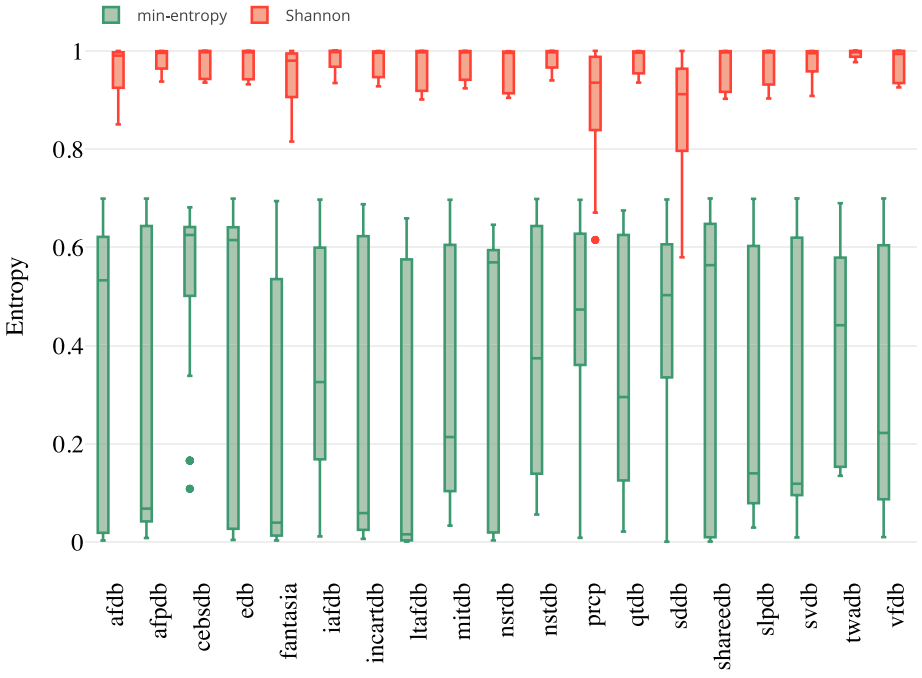
Fig. 5. Entropy analysis of files generated by extracting 3 bits from IPIs. Fig. 5a represents the maximum number of passed estimators that achieves at least one combination of bits. Fig. 5b shows the best and most common combination of bits of databases.

the results for this experiment into two groups of databases: (1) a first group with databases that passed 5 estimators (fantasia, iafdb, nstdb and twadb), and; (2) a group with databases that passed 4 estimators (afdb, afpdb, cebsdb, edb, incartdb, ltafdb, mitdb, nsrdb, prcp, qtdb, sddb, shareedb, slpdb, svdb and vfdb). Contrarily to the first experiment, now databases seem to converge between four and five passed estimators at the most. However, we cannot extract any conclusion about the nature of the databases, i.e., healthy databases and people with diseases are mixed indistinguishably.

Fig. 5b shows, a clear tendency: the 2nd MSB appears in all the combination of bits which achieve the best results: {268, 286, 628, 682, 826, 862, 278, 287, 728, 782, 827, 872}, but: {638, 836 }. Additionally, if we take into account the size constraint suggested by NIST recommendation (see Table 2), we can claim that the best combinations of bits for $V_3(8)$ are given by the permutation of the positions 2, 6 and 8, i.e., $P_3\{2, 6, 8\} = \{268, 286, 628, 682, 826, 862\}$, the permutation of the positions 2, 7 and 8, i.e., $P_3\{2, 7, 8\}$ and the combinations 638 and 836. These results are clearly in contradiction to what many researchers claimed about which are the best bits to choose, i.e., the composition of the LSBs, which in this case would have been the combination of bits 678.

We conducted one more experiment to check how many databases have the combination of bits 678 as the best one. We obtained that afdb, afpdb, cebsdb, edb, fantasia, incartdb, ltafdb, mitdb, qtdb, shareedb, slpdb, svdb, twadb, vfdb databases have it (14 out of 19). However, the set mentioned before: $\{P_3\{2, 6, 8\}\} \bigcup \{P_3\{2, 7, 8\}\} \bigcup \{638, 836\}$, apart from the aforementioned databases, they also have in common nstdb and nsrdb databases (16 out of 19).

It can be seen in Fig. 6a a comparison between values obtained from running the Shannon entropy and the min-entropy estimators for all the generated $V_3(8)$ variations and grouped per databases. In this plot, it can be observed how, just by calculating the Shannon entropy, cannot be claimed that the heart signal



(a) Shannon entropy vs min-entropy tests.

	most_common	collision	markov	maurer_universal	MultiMCW	Lag	MultiMMC	LZ78Y	t_tuple	LRS
afdb	114	7	0	90	125					
afpdb	72	0	0	29	235					
cebsdb	126	0	0	0	210					
edb	120	0	0	54	162					
fantasia	96	30	0	12	30	168				
iafdb	78	0	0	1	256	1				
incartdb	90	0	0	24	222					
ltafdb	126	0	0	24	186					
mitdb	108	0	0	0	228					
nsrdb	126	0	0	120	90					
nstdb	72	0	0	11	229	24				
prcp	222	0	40	2	72					
qtdb	90	0	0	0	246					
sddb	240	0	11	49	36					
shareedb	126	0	0	96	114					
slpdb	126	0	0	25	185					
svdb	96	0	0	35	205					
twadb	0	0	0	0	215	121				
vfdb	90	0	0	27	219					

(b) Failed estimators.

Fig. 6. Entropy analysis of files generated by extracting 3 bits from IPIs. Fig. 6a depicts a comparison of the min-entropy and the Shannon entropy. Fig. 6b shows a heatmap of the most failed estimators per database.

can be considered a good source of entropy. The results are far from being acceptable, and this leads us to compute one final plot regarding which estimators are the worst ones, i.e., a statistical analysis of which estimators that the bit streams generated failed the most are.

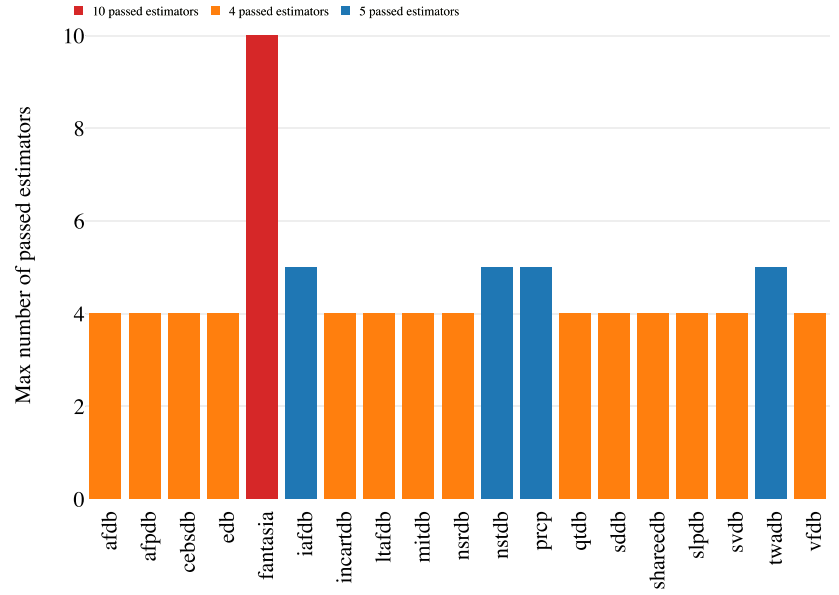
In Fig. 6b we can see that in general, most of the databases fail in the MultiMCW. Besides, the files of fantasia database usually fail in the Lag estimator—which is an extended and improved version of the MultiMCW estimator. It is also worth mentioning that prcp and sddb databases fail in the first estimator, i.e., Most_common. This estimator predicts the next output based on previous knowledge. Thus, we can claim that both prcp and sddb are not good choices when using 3 bits. Finally, it is interesting to see that none of the databases managed to execute the last four estimators because they failed in previous ones.

3.3. $V_4(8)$ *variations of four bits without repetition*

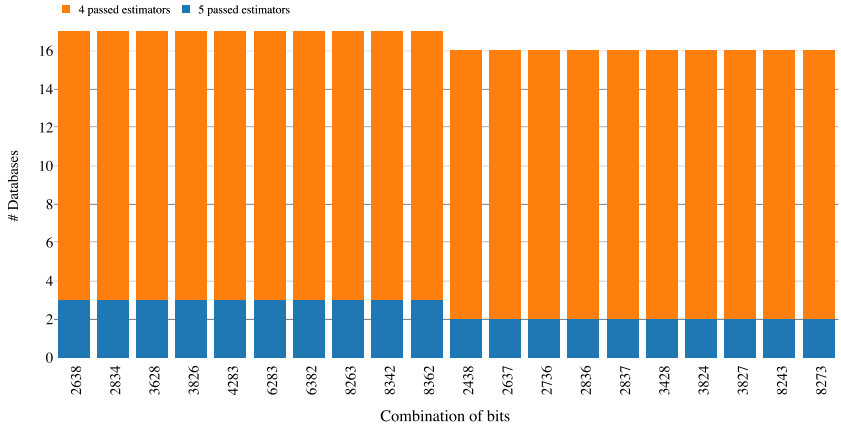
For this experiment, we generated the 1680 possible variations without repetition corresponding to how many different ways four items from eight elements can be chosen. We passed the min-entropy estimators to all the 1680 * 19 files, and the results can be seen in Figs. 7 and 8.

It is interesting to see that the results are quite similar to the obtained in the previous experiment (i.e., $V_3(8)$). In detail, the number of estimators that are successfully passed in each database is exactly the same with some exceptions. Prpc database passes one more estimator and, in fantasia database, at least, a combination of bits passes all the estimators at a time. In relation to Fig. 7b, the set of combinations of bits that passes more tests are the following ones: {2638, 2834, 3628, 3826, 4283, 6283, 6382, 8263, 8342, 8362}.

In this case, it is remarkable that none of the databases of the group that passes five estimators (i.e., {iafdb, nstdb, prcp, twadb}), achieves the minimum requirement in terms of size that the NIST recommendation establishes (see Table 2). Despite that, and given the fact that we found some works in the literature which directly or indirectly use some of these databases for security purposes [56,57], we decided to keep them in the interpretation of the results.



(a) Number of passed estimators



(b) Best combinations.

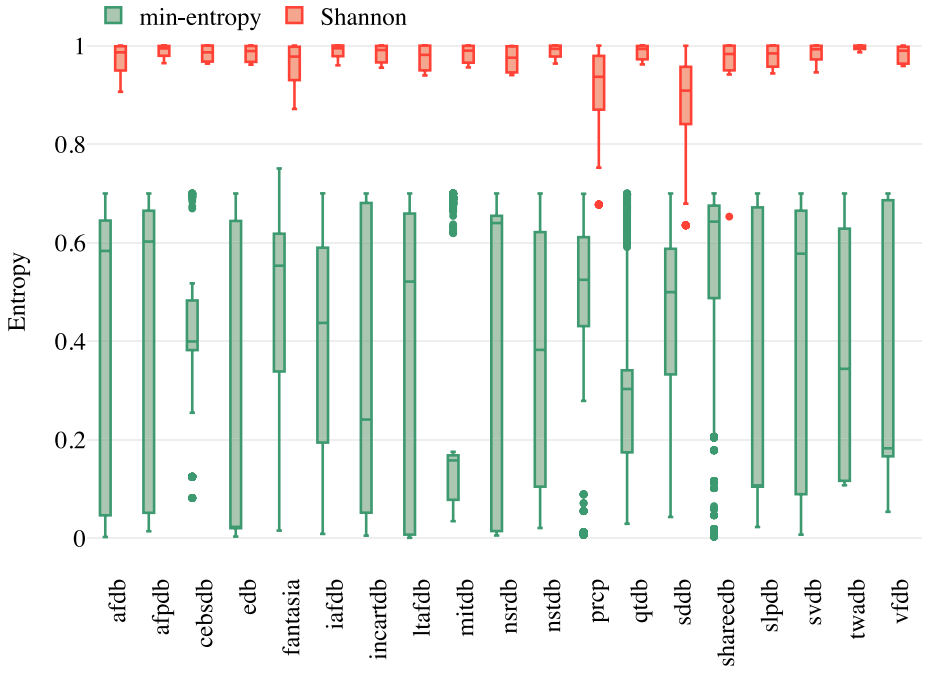
Fig. 7. Entropy analysis of files generated by extracting 4 bits from IPIs. Fig. 7a represents the maximum number of passed estimators that achieves at least one combination of bits. Fig. 7b shows the best and most common combination of bits of databases.

Fig. 7b depicts the best and most common combination of bits for $V_4(8)$. One thing that drew our attention regarding such a plot is that the 3rd bit appears in all the top 20 of the most common combinations whereas the 5th MSB (or the 4th LSB) is not in any of them. Remember that for four bits, the combination that has usually been taken in the literature is 5678 [5].

We conducted the same verification as in the previous experiments to certify where the combination of bits used by the majority of the IPI-based papers is. We got that such a combination is considered to be the best one in cebsdb, edb, mitdb, qtdb, shareedb, slpdb, svdb, twadb, vfdb, i.e., (9 out of 19). To compare the improvement we achieved by using any of the set of the best combination we generated, i.e., {2638, 2834, 3628, 3826, 4283, 6283, 6382, 8263, 8342, 8362}, we computed the databases that have any of the elements of such a set which are: afdb, afpdb, cebsdb, edb, iafdb, incartdb, ltafdb, mitdb, nsrdb, nstdb, qtdb, sddb, shareedb, slpdb, svdb, twadb, vfdb, i.e., the same databases that the combination of the last 4 LSBs (i.e., 5678) plus 8 more databases (17 out of 19in total). With this, we can conclude that the combination that is usually taken in the literature is, by far, not the best one that can be chosen from the min-entropy point of view according to the NIST recommendation.

In Fig. 8a, we can see a comparison between the Shannon entropy and the min-entropy tests when applied to all the $V_4(8)$ variations. Similar to the rest of the performed experiments, we cannot say that there is a significant improvement concerning the $V_3(8)$ experiment. We can see how some databases improve their results like afpdb, fantasia, ltafdb, shareedb or svdb, but the general improvement is not breakthrough. Contrary to what can be claimed using the Shannon entropy, from the min-entropy values we can conclude that the 4th bits of the IPI values are not entropic (i.e., they are not a good source of randomness).

Finally, Fig. 8b shows the distribution of the most failed tests for tokens generated taking four bits of the IPI. Once again, it is interesting to see how the fantasia database obtains the best results of all the experiments. Even though 109 combinations fail the t_tuple estimator, it is the only database that achieves that estimator (remember that estimators are executed sequentially and we stop when one of the tests fails). About the rest of the databases, it can be seen how now the databases fail the Collision instead of the Compression (Maurer Universal Statistic tests) estimators. Recall that the Collision estimator mainly detects when the source is biased towards a particular value, whereas the Compression estimator tries to compress the values and generates the average number of samples needed to produce such an



(a) Shannon entropy vs min-entropy tests.

	most_common	collision	markov	maurer_universal	MultiMCW	Lag	MultiMMC	LZ78Y	t_tuple	LRS
afdb	552	495	0	6	627					
afpdb	0	910	0	0	770					
cebsdb	0	33	0	0	1647					
edb	24	611	0	0	1045					
fantasia	504	116	6	5	128	755	34	1	109	
iafdb	72	293	0	0	1069	246				
incartdb	240	487	0	0	953					
ltafdb	700	143	0	0	837					
mitdb	288	56	0	0	1336					
nsrdb	840	369	0	0	471					
nstdb	0	266	0	0	996	418				
prcp	1032	346	106	2	170	24				
qtdb	48	345	0	0	1287					
sddb	1296	76	11	16	281					
shareedb	542	642	0	0	496					
slpdb	600	175	0	0	905					
svdb	240	628	0	0	812					
twadb	0	59	0	0	653	968				
vfdb	408	308	0	0	964					

(b) Failed estimators.

Fig. 8. Entropy analysis of files generated by extracting 4 bits from IPIs. Fig. 8a depicts a comparison of the min-entropy and the Shannon entropy. Fig. 8b shows a heatmap of the most failed estimators per database.

output. Roughly speaking, it can be observed how by increasing the number of bits per file, they can be more compressed, but they are biased by either having more 0's or 1's values.

3.4. $V_5(8)$ *variations of five bits without repetition*

The last variation that we computed is $V_5(8)$. In general, it can be seen in Fig. 9a how the maximum number of estimators that databases got are almost the same than in $V_4(8)$. Nevertheless, in this case, fantasia database passes only eight out of ten estimators at the most (10 out of 10 in $V_4(8)$). With all this, we can conclude that taking 5 bits does not directly affect the final min-entropy value.

This result is an improvement in terms of performance due to the usual procedure to generate random tokens, i.e., appending some bits to create bit streams of a given size. For instance, to create a token of 128 bits by appending the four LSBs, it would be needed—at least—32 IPIs. On the contrary, if five bits were used, then—at least—27 IPIs would be required. Note that, if for example, a healthy subject beats one time per second (60 bits per minute), we will be saving 5 s to generate the same key.

Regarding the best combination of bits (see Fig. 9b), it can be seen how they are an extension of the previous combinations in $V_4(8)$. For example, in $V_4(8)$, the best combination is 2638 whereas for five bits, the same combination plus the 2nd MSB forms one of the best options (i.e., 26 387). Another similar case can be seen for the combination of 36 287 bits.

Also, note that the top 20 of best and most frequent combinations are in common in 19 out of 19. These results, once again corroborate the previous conclusion: it is better to use 5 bits instead of 4. This fact means that for instance, the combination of bits 26 387 is the best one possible in any of the tested databases. Additionally and for completeness, we looked for the usually assumed combination of bits 45 678 being best one in only 11 databases (afpdb, cebsdb, edb, incartdb, ltafdb, mitdb, qtdb, shareedb, svdb, twadb, vfdb) which is far from any of the best combinations.

Fig. 10a shows both Shannon and the min-entropy values. The results follow the same pattern as in the previous experiments in the sense that there is a considerable distance between them. However, we can now see how the min-entropy values of the

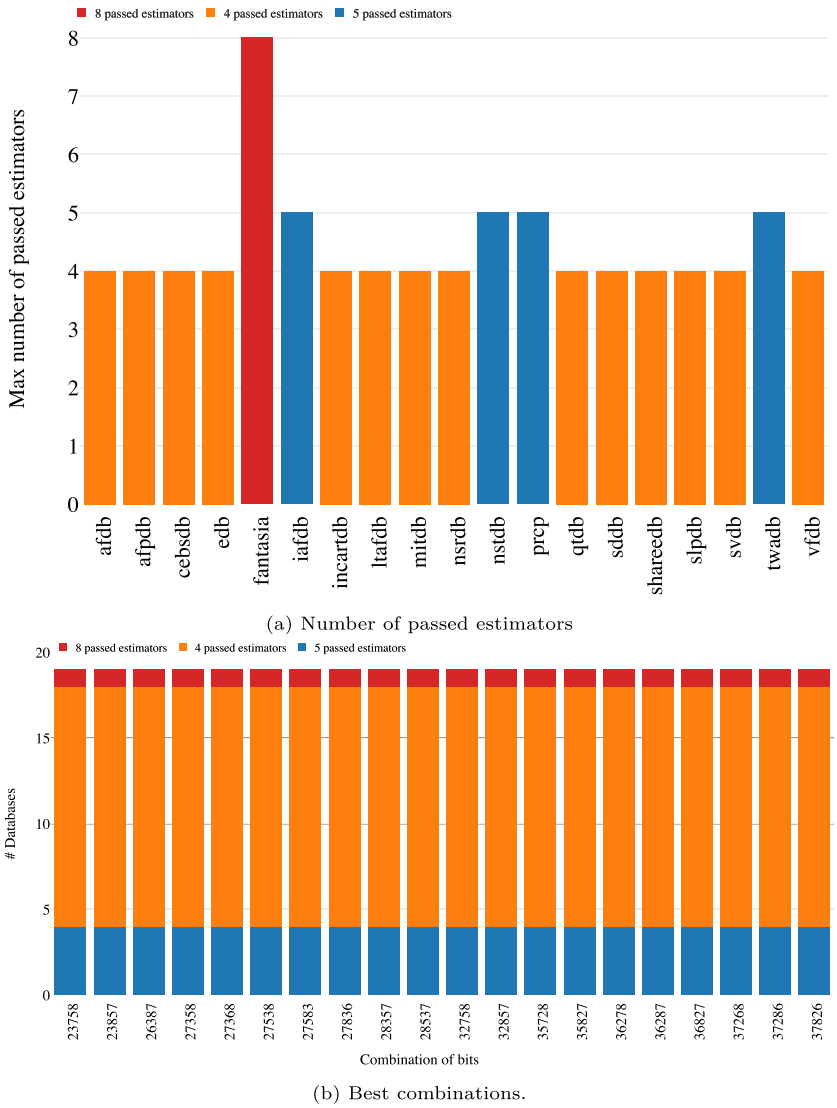


Fig. 9. Entropy analysis of files generated by extracting 5 bits from IPIs. Fig. 9a represents the maximum number of passed estimators that achieves at least one combination of bits. Fig. 9b shows the best and most common combination of bits of databases.

databases are less spread than in any of the previous experiments. Once again, and following the same line as in $V_3(8)$ and $V_4(8)$, we can see how fantasia database achieves the best results in average (note that the median is close to 0.6). In any case, in general, from the perspective of min-entropy, the results are not good enough to consider the ECG signal as a good source of entropy.

Finally, in Fig. 10b, we can see that most of the databases fail in the MultiMCW estimator (similar to $V_3(8)$ and $V_4(8)$ experiments). Fantasia database achieves the best results of the tested databases concluding then that fantasia database is the best one.

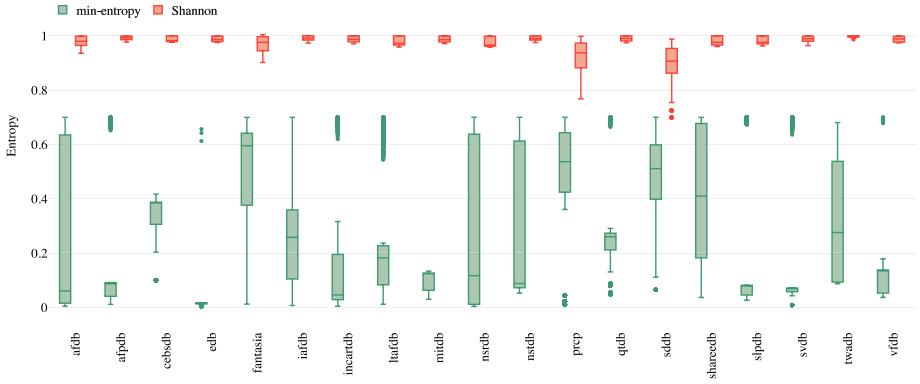
3.5. Limitations and discussion

As mentioned at the beginning of this Section, we set up a threshold of 0.7 to claim when a sequence of bits passes or not each one of the estimators of the min-entropy. We are aware that this threshold might be subjective. Unfortunately, even with this relaxed threshold, the results are not as good as it is usually claimed in the literature so far. In order to be more realistic, we can increase up that threshold to 0.9, which is the number we have observed in many scientific papers [6,8,14,50,55] as well as the result we obtained after running the `urand` function (see Table 3) and we summarize the results in Fig. 11. These plots show that results are worse—as it was expected—than setting the threshold to 0.7 in terms of the maximum number of passed estimators—note that ten is the ideal value.

Finally, in Table 4, we summarize the conclusions we got from all the experiments executed. It is also interesting to mention that we limited the representation of the results to the top 20. This has direct repercussions in $V_5(8)$, both Fig. 9b and Table 4 where only the first 20 combinations are shown. However, more combinations are not there and achieve the same results.

4. Related work

Bao et al. [3], Poon et al. [58] and Bao et al. [17] proposed in 2004, 2006 and 2008 respectively, different protocols to secure *Body Area Networks (BANs)*. In these proposals, the authors claimed that the ECG signal, and in particular the *Inter-Pulse Intervals* (IPIs) value have entropy, and therefore, can be used for security purposes. In the following, we try to summarize and classify the most relevant contributions as well as the methodology authors used (if any) to check that the chosen bits have entropy. More concretely, we only focus on those works which systematically pick the n Least Significant Bits (LSBs) of the IPIs. We leave out of this summary those works where no info is given

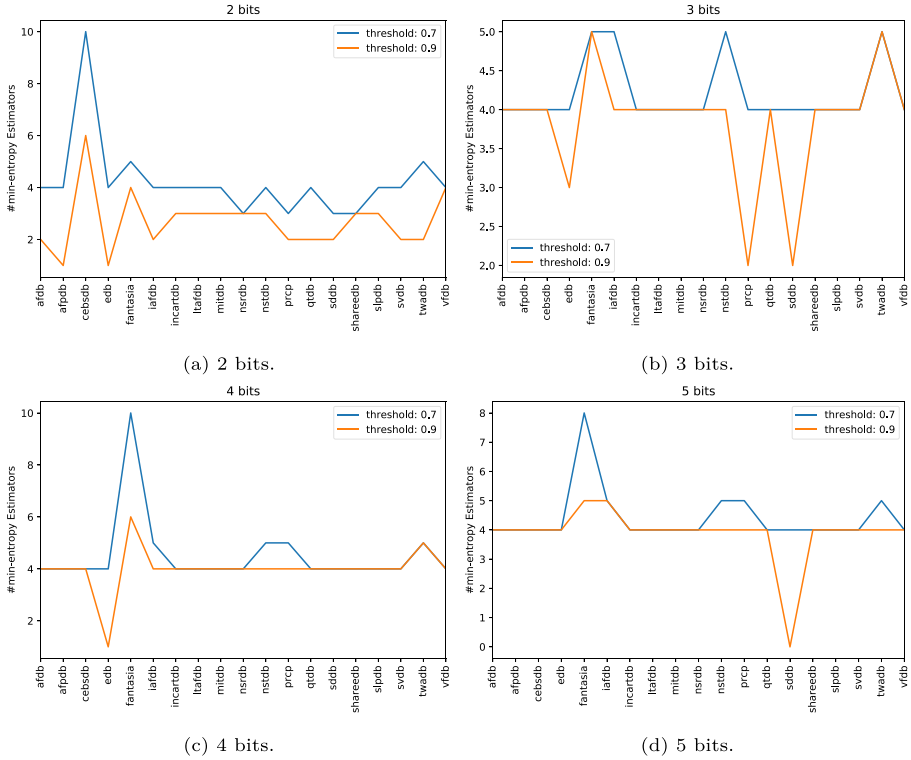


(a) Shannon entropy vs min-entropy tests

	most common	collision	markov	maurer	universal	MultiMCW	Lag	MultiMMC	LZ78Y	t tuple	LRS
afdb	1560	366	0	26		4768					
afpdb	0	222	0	0		6498					
cebsdb	0	0	0	0		6720					
edb	0	3	0	0		6717					
fantasia	1800	203	0	2		298	3583	233	0	601	
iafdb	0	229	0	0		3770	2721				
incartdb	0	129	0	24		6567					
ltafdb	240	702	35	29		5714					
mitdb	0	0	0	0		6720					
nsrdb	1489	344	14	74		4799					
nstdb	0	269	0	0		3286	3165				
prcp	4320	39	202	0		1623	536				
qtdb	0	134	0	0		6586					
sddb	5520	3	119	0		1078					
shareedb	1920	286	4	0		4510					
slpdb	1560	31	0	0		5129					
svdb	0	354	0	0		6366					
twadb	0	0	0	0		1925	4795				
vfdb	0	13	0	0		6707					

(b) Failed estimators

Fig. 10. Entropy analysis of files generated by extracting 5 bits from IPIs. Fig. 10a depicts a comparison of the min-entropy and the Shannon entropy. Fig. 10b shows a heatmap of the most failed estimators per database.



(a) 2 bits.

(b) 3 bits.

(c) 4 bits.

(d) 5 bits.

Fig. 11. Min-entropy comparison with thresholds equal to 0.7 and 0.9.

Table 4
Summary of the experiments (first column) carried out together with the best combination of bits (second column) and the number of common databases that have these combinations (last column).

Variations	Best combinations	Databases
$V_2(8)$	$\{P_2\{7, 8\}\}$	11 out of 19
$V_3(8)$	$\{P_3\{2, 6, 8\}\} \bigcup \{P_3\{2, 7, 8\}\} \bigcup \{638, 836\}$	16 out of 19
$V_4(8)$	{2638,2834,3628,3826,4283,6283,6382,8263,8342,8362}	17 out of 19
$V_5(8)$	{23758,23857,26387,27358,27368,27538,27583,27836} $\bigcup$ {28357,28537,32758,32857,35728,35827,36278,36287} $\bigcup$ {36827,37268,37286,37826}	19 out of 19

about how long the sequence is [51,59] or those which do not use the quantization algorithm proposed by Rostami et al. [5] using wavelets instead [10,60].

In 2010, Venkatasubramanian and Gupta [11] proposed an IPI-based protocol to secure communications between sensors in a *Body Sensor Network (BSN)*. This protocol was based on Poon et al.'s work [58] and authors did not corroborate the entropy of IPI values. A year later, in 2011 Xu et al. [4] proposed IMDGuard, a security scheme for *Implantable Medical Devices (IMDs)* where the key establishment is based on IPIs. In this paper, authors were the first who introduced the quantization algorithm—a pre-processing signal algorithm—and carried out an in-depth analysis of IPI randomness by running a subset of NIST STS tests [61] and stated that the four LSBs are random.

Based on previous works [4,11,58], in 2013, Rostami et al. [5] carried out an experiment against ptbdb, mitdb and mghdb databases and, after extracting the IPIs and running the quantization algorithm they corroborated that the last 4 LSBs of the IPI are totally uncorrelated just by calculating the Shannon entropy. In the same year and based on the same papers, Hu et al. proposed OPFKA [62], a secure key establishment protocol. However, authors: (1) did not corroborate previous results, and; (2) did not mention which databases from Physionet repository they used for their experiments.

Seepers et al. [8] proposed in 2015 a key generation procedure based on IPIs. Regarding the entropy evaluation, authors analyzed the entropy of the 4 LSBs obtained from the IPIs from 42 subjects from mitdb and fantasia, and 111 subjects from BioSec[4] database. They claimed that the Shannon entropy was gradually decreasing when taking more significant bits. Another key generation protocol was proposed a year later, in 2016, Altop et al. [6] based on IPIs. Authors used to test their proposal 50 subjects from the MIMIC II Waveform [63], and they calculated the Shannon entropy to check how entropic the heart signals are. We want to highlight that none of the described proposals checks different combinations of bits as we proposed in this work.

Recently, Kim et al. [7] studied the peak misdetection issue and proposed a recovery key exchange protocol. In their proposal, they used the Physionet repository, but they did not specify which databases they used and tested their solution by using a subset of NIST STS suite and all the tests provided by AIS.31 [64].

Koya et al. [16] proposed a hybrid mutual authentication and key agreement scheme for *Wireless Body Area Networks (WBANs)*. Authors appended the four LSBs to create a 128-bits token and tested their proposal against both ptbdb and mitdb databases. Similarly to previous works, authors just calculated the standard Shannon entropy to check and claim that the generated tokens are random.

5. Conclusions

In this article, we scrutinized the IPI values from an ECG signal as an entropy source generator. In detail, we analyzed and empirically demonstrated that taking the Least Significant Bits (LSBs) of the IPI values, as has been done so far in most contributions [6–10], is not the best approach for randomness generation. Instead, we generated variations without repetition of eight elements—corresponding to the position of the bits in an IPI—taken from two, three, four and five bits respectively and generated thousands of files that we then analyzed by using the min-entropy estimators proposed by the NIST SP 800-90B. Note that the use of the min-entropy is a more conservative approach, and this value will never surpass the Shannon entropy. From this analysis, we offered other alternative combinations for two (e.g., 87), three (e.g., 638), four (e.g., 2638) and five (e.g., 23758) bits which are, in general, much better than taking the four LSBs from the entropy point of view.

As future work, we plan to analyze the randomness quality of the files generated with the best combinations of IPI bits obtained from our in-depth and rigorous analysis. As suggested by NIST SP 800-90B, a conditioning component (e.g., to reduce bias and/or increase the entropy rate) may be necessary to improve the randomness quality of the final output.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Bai, J. Lin, G. Li, H. Wang, P. Ran, Z. Li, D. Li, Y. Pang, W. Wu, G. Jeon, A lightweight method of data encryption in bans using electrocardiogram signal, Future Gener. Comput. Syst. 92 (2019) 800–811.
[2] L. Gonzalez-Manzano, J.M. de Fuentes, P. Peris-Lopez, C. Camara, Encryption by heart (ebh) – using ecg for time-invariant symmetric key generation, Future Gener. Comput. Syst. 77 (2017) 136–148.
[3] S.-D. Bao, L.-F. Shen, Y.-T. Zhang, A novel key distribution of body area networks for telemedicine, in: IEEE International Workshop on Biomedical Circuits and Systems, 2004, pp. 1–17.
[4] F. Xu, Z. Qin, C.C. Tan, B. Wang, Q. Li, IMDGuard: Securing implantable medical devices with the external wearable guardian, in: INFOCOM, 2011, pp. 1862–1870.
[5] M. Rostami, A. Juels, F. Koushanfar, Heart-to-heart (H2H): authentication for implanted medical devices, in: CCS, 2013, pp. 1099–1112.
[6] D.K. Altop, A. Levi, V. Tuzcu, Deriving cryptographic keys from physiological signals, Pervasive Mob. Comput. (2016).
[7] J. Kim, K. Cho, Y.-K. Kim, K.-S. Lim, S.U. Shin, Study on peak misdetection recovery of key exchange protocol using heartbeat, J. Supercomput. (2018).
[8] R.M. Seepers, C. Strydis, I. Sourdis, C.D. Zeeuw, Enhancing heart-beat-based security for mhealth applications, IEEE J. Biomed. Health Inf. PP (99) (2015) 1.
[9] R.M. Seepers, C. Strydis, I. Sourdis, C.I.D. Zeeuw, On using a von Neumann extractor in heart-beat-based security, in: Trustcom/BigDataSE/ISPA, 2015 IEEE, Vol. 1, 2015, pp. 491–498.
[10] M.V. Karthikeyan, J.M.L. Manickam, ECG-signal based secret key generation (ESKG) scheme for WBAN and hardware implementation, Wirel. Pers. Commun. (2018).
[11] K.K. Venkatasubramanian, S.K.S. Gupta, Physiological value-based efficient usable security solutions for body sensor networks, ACM Trans. Sens. Netw. 6 (4) (2010) 31:1–31:36.
[12] T. Belkhouja, X. Du, A. Mohamed, A.K. Al-Ali, M. Guizani, Biometric-based authentication scheme for implantable medical devices during emergency situations, Future Gener. Comput. Syst. 98 (2019) 109–119.

[4] https://www.comm.utoronto.ca/~biometrics/databases.html.

[13] R.M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, C.I. De Zeeuw, Peak misdetection in heart-beat-based security: Characterization and tolerance, in: EMBC, 2014, pp. 5401–5405.
[14] S. Pirbhulal, H. Zhang, W. Wu, S.C. Mukhopadhyay, Y. Zhang, Heartbeats based biometric random binary sequences generation to secure wireless body sensor networks, IEEE Trans. Biomed. Eng. 65 (12) (2018) 2751–2759.
[15] S. Pirbhulal, O.W. Samuel, W. Wu, A.K. Sangaiah, G. Li, A joint resource-aware and medical data security framework for wearable healthcare systems, Future Gener. Comput. Syst. 95 (2019) 382–391.
[16] A.M. Koya, D.P. P., Anonymous hybrid mutual authentication and key agreement scheme for wireless body area network, Comput. Netw. 140 (2018) 138–151.
[17] S.-D. Bao, C.C. Poon, Y.-T. Zhang, L.-F. Shen, Using the timing information of heartbeats as an entity identifier to secure body sensor network, Trans. Inf. Tech. Biomed. 12 (6) (2008) 772–779.
[18] T. Hong, S.D. Bao, Y.T. Zhang, Y. Li, P. Yang, An improved scheme of IPI-based entity identifier generation for securing body sensor networks, in: EMBS, 2011, pp. 1519–1522.
[19] Y. Zhang, Y. Sun, P. Phillips, G. Liu, X. Zhou, S. Wang, A multilayer perceptron based smart pathological brain detection system by fractional fourier entropy, J. Med. Syst. 40 (7) (2016) 173.
[20] M. Rushanan, A.D. Rubin, D.F. Kune, C.M. Swanson, SoK: Security and privacy in implantable medical devices and body area networks, in: Security & Privacy, 2014, pp. 524–539.
[21] M.S. Turan, E. Barker, J. Kelsey, K.A. McKay, M.L. Baish, M. Boyle, Recommendation for the Entropy Sources Used for Random Bit Generation, NIST Special Publication, 2018.
[22] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220.
[23] M. GB, M. RG, A new method for detecting atrial fibrillation using R-R intervals, Comput. Cardiol. (1983).
[24] G. Moody, A. Goldberger, S. McClennen, S. Swiryn, Predicting the onset of paroxysmal atrial fibrillation, in: Computers in Cardiology, 2001, pp. 113–116.
[25] M.A. García-González, A. Argelagós-Palau, M. Fernández-Chimeno, J. Ramos-Castro, A comparison of heartbeat detectors for the seismocardiogram, in: Computing in Cardiology, 2013, pp. 461–464.
[26] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. Moody, C. Zeelenberg, C. Marchesi, The european ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography, Eur. Heart J. 13 (9) (1992) 1164–1172.
[27] N. Iyengar, C. Peng, R. Morin, A.L. Goldberger, L.A. Lipsitz, Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics, Amer. J. Physiol.-Regul. Integr. Comp. Physiol. 271 (4) (1996) R1078–R1084.
[28] Physionet, intracardiac atrial fibrillation database, 2018, URL https://physionet.org/physiobank/database/iafdb/.
[29] Physionet, st.-petersburg institute of cardiological technics 12-lead arrhythmia database, 2018, URL https://physionet.org/physiobank/database/incartdb/.
[30] S. Petrutiu, A.V. Sahakian, S. Swiryn, Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans, Europace 9 (7) (2007) 466–470.
[31] G.B. Moody, R.G. Mark, The impact of the MIT-bih arrhythmia database, Eng. Med. Biol. Mag. IEEE 20 (3) (2001) 45–50.
[32] Physionet, the MIT-BIH normal sinus rhythm database, 2018, URL https://physionet.org/physiobank/database/nsrdb/.
[33] G.B. Moody, W. Muldrow, R.G. Mark, A noise stress test for arrhythmia detectors, Comput. Cardiol. 11 (3) (1984) 381–384.
[34] T. Heldt, M.B. Oefinger, M. Hoshiyama, R.G. Mark, Circulatory response to passive and active changes in posture, in: Computers in Cardiology, 2003, pp. 263–266.
[35] P. Laguna, R.G. Mark, A. Goldberg, G.B. Moody, Database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG, in: Computers in Cardiology, Vol. 1997, 1997, pp. 673–676.
[36] S.D. Greenwald, The Development and Analysis of a Ventricular Fibrillation Detector (Ph.D. thesis), Massachusetts Institute of Technology, 1986.
[37] P. Melillo, R. Izzo, A. Orrico, P. Scala, M. Attanasio, M. Mirra, N. De Luca, L. Pecchia, Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis, PLoS One 10 (3) (2015) e0118504.
[38] Y. Ichimaru, G.B. Moody, Development of the polysomnographic database on CD-ROM, Psychiatry Clin. Neurosci. 53 (2) (1999) 175–177.
[39] S.D. Greenwald, R.S. Patil, R.G. Mark, Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information, in: Computers in Cardiology, 1990, pp. 461–464.
[40] G.B. Moody, The physionet/computers in cardiology challenge 2008: T-wave alternans, in: 2008 Computers in Cardiology, IEEE, 2008.
[41] S.D. Greenwald, The development and analysis of a ventricular fibrillation detector (Ph.D. thesis), Massachusetts Institute of Technology, 1986.
[42] S. Peter, B. Pratap Reddy, F. Momtaz, T. Givargis, Design of secure ECG-based biometric authentication in body area sensor networks, Sensors 16 (4) (2016).
[43] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, IEEE Trans. Biomed. Eng. BME-32 (3) (1985) 230–236.
[44] L. Ortiz-Martin, P. Picazo-Sanchez, P. Peris-Lopez, J. Tapiador, G. Schneider, Feasibility analysis of inter-pulse intervals based solutions for cryptographic token generation by two electrocardiogram sensors, Future Gener. Comput. Syst. (ISSN: 0167-739X) 96 (2019) 283–296.
[45] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[46] B.B. Gupta, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, 2018.
[47] E. Barker, J. Kelsey, Recommendation for the entropy sources used for random bit generation, Draft NIST Special Publication, 2012, pp. 800–900.
[48] A. Rényi, On measures of entropy and information, in: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, University of California Press, Berkeley, Calif., 1961, pp. 547–561.
[49] H. Chizari, E. Lupu, Extracting randomness from the trend of IPI for cryptographic operators in implantable medical devices, 2018, CoRR abs/1806.10984.
[50] S.R. Moosavi, E. Nigussie, S. Virtanen, J. Isoaho, Cryptographic key generation using ECG signal, in: CCNC, 2017, pp. 1024–1031.
[51] L. Yao, B. Liu, G. Wu, K. Yao, J. Wang, A biometric key establishment protocol for body area networks, Int. J. Distrib. Sens. Netw. 2011 (2011).
[52] P. Hagerty, T. Draper, Entropy bounds and statistical tests, in: NIST Random Bit Generation Workshop, 2012, pp. 1319–1327.
[53] U.M. Maurer, A universal statistical test for random bit generators, J. Cryptol. 5 (2) (1992) 89–105.
[54] D. Salomon, Data Compression: The Complete Reference, Springer Science & Business Media, 2004.
[55] G. Zheng, G. Fang, R. Shankaran, M.A. Orgun, Encryption for implantable medical devices using modified one-time pads, IEEE Access 3 (2015) 825–836.
[56] R. Lazzeretti, J. Guajardo, M. Barni, Privacy preserving ECG quality evaluation, in: MM & Sec, 2012, pp. 165–174.
[57] B. Wu, G. Yang, L. Yang, Y. Yin, Robust ECG biometrics using two-stage model, in: ICPR, 2018, pp. 1062–1067.
[58] C.C.Y. Poon, Y.-T. Zhang, S.-D. Bao, A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health, IEEE Commun. Mag. 44 (4) (2006) 73–81.
[59] E.K. Zaghouani, A. Jemai, A. Benzina, R. Attia, ELPA: A new key agreement scheme based on linear prediction of ECG features for WBAN, in: EUSIPCO, 2015, pp. 81–85.
[60] C. Camara, P. Peris-Lopez, H. Martí n, M. Aldalaien, ECG-RNG: A random number generator based on ECG signals and suitable for securing wireless sensor networks, Sensors 18 (9) (2018).
[61] L.E. Bassham III, A.L. Rukhin, J. Soto, J.R. Nechvatal, M.E. Smid, E.B. Barker, S.D. Leigh, M. Levenson, M. Vangel, D.L. Banks, N.A. Heckert, J.F. Dray, S. Vo, SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Tech. Rep, National Institute of Standards & Technology, 2010.
[62] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, D. Chen, OPFKA: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks, in: INFOCOM, 2013, pp. 2274–2282.
[63] M. Saeed, M. Villarroel, A.T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T.H. Kyaw, B. Moody, R.G. Mark, Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database, Crit. Care Med. 39 (5) (2011) 952.
[64] H. Park, J.-S. Kang, Y. Yeom, Probabilistic analysis of AIS. 31 statistical tests for TRNGs and their applications to security evaluations, J. Korea Inst. Inf. Secur. Cryptol. 26 (1) (2016) 49–67.

Lara Ortiz-Martin is a PhD. student in Computer Science at Universidad Carlos III de Madrid. She received a MSc. degree in Computer Science from the same university. Her current research interests include systems security, applied cryptography and biometrics. She is also working on web technologies in the industry.

Pablo Picazo-Sanchez received the PhD. degree in Computer Science from the Carlos III University of Madrid, in 2016. Currently he works at Chalmers University, Sweden, in a postdoc position in the Formal Methods division. His current research interests include systems security, applied cryptography and web security.

Pedro Peris-Lopez holds an Associate Professor position at Universidad Carlos III de Madrid. He has an M.Sc. in Telecommunications Engineering (2004) and PhD in Computer Science (2008) from University Carlos III of Madrid. His research interests are in the fields of cryptography, computer forensics, signal processing, and artificial intelligence. Nowadays, his research is mainly focused on Implantable Medical Devices (IMD) and Biomedical applications. He has published many articles (56) in International Journals with impact factor and papers (45) in International Conferences of recognized prestige (peer-reviewed; 2-4 reviewers). His works have a high impact: the whole of his works have more than 4000 cites, and his h-index is 29 (12/2019 – Google Scholar). For additional information see: www.lightweightcryptography.com