Security and privacy services based on biosignals for implantable and wearable devices

Lara María Ortiz Martín

in partial fulfillment of the requirements for the degree of Doctor in Computer Science and Engineering

Universidad Carlos III de Madrid

Advisor:
Dr. Pedro Peris López

September 2019

Para Óliver

Agradecimientos

Me gustaría agradecer esta tesis a aquellas personas que a lo largo de este tiempo me han preguntado, animado y ayudado a que este proyecto llegue a su fin.

En primer lugar, me gustaría agradecer mi tutor Dr. Pedro Peris y a Juan. E. Tapiador por confiar en mí desde el principio, alentarme cuando lo creí imposible y guiarme durante todo el proceso.

Recordar siempre a la gente de la Politécnica porque ahí fue el inicio de esta aventura: Goyi, Ana, Ángeles, Pilar, Isabel y Jesús.

De los Netijanos agradecer su interés y su apoyo a Ángel, Moi, Andrés, Paloma y Gabri. Y en especial quiero agradecer a Teto por darme la flexibilidad en el trabajo para conseguirlo.

También tengo que mencionar a mis chicas del baile, porque aunque ahora ya nos estemos juntas, siempre se interesaron por cómo iba y me animaron a seguir.

No puedo dejar de mencionar a las amigas del cole que por más años que pasen están ahí preocupándose por mi: Rocío, Alba, Lara, María y Raquel.

Mención especial a la gente de Leganés y Albacete por darme tanto cariño y acogerme como una Pepinera o Albaceteña más. También a mis dos jugones favoritos Porre y Gladys por tratarme como una hermana.

De la estancia Sueca tengo que dar las gracias a Gerardo, Hanna, Deborah (y sus pequeños Eithan y Ofir), Raúl, Mauricio, Elena, Miguel, Carlo, Jorge, Lorena, Steven, Michael, Ceci, Carlos, Vincenzo, Rocio y Frida. Formasteis mi nuevo mundo lejos de España y me sentí acogida y feliz.

Sin olvidarme de la familia, tengo que agradecer a mi madre y a Juan, por quererme tanto y apoyarme siempre en mis proyectos y aventuras. También a mis primos, a mi tío Mariano, Cruz y Germán por su cariño e interés en que terminara. Recordar a mis abuelos porque con ellos aprendí muchas cualidades que hoy en día me hacen ser como soy y me ayudan a conseguir objetivos como esta tesis.

Por último, y no menos importantes a mis dos satélites particulares que me quieren con locura y me hacen feliz cada día. Concretamente a mi pequeño Oliver por poner patas arriba mi mundo y llenar cada día de nuevas aventuras. Y a Pica, por todo el esfuerzo y empeño que ha puesto para que yo consiguiera esta tesis.

Published and Submitted Content

During this PhD, most of the research work done have resulted in scientific publications. More concretely, 3 international journals with Impact Factor (IF), one paper in a national conference and one poster in a Tier-1 conference.

Title:	Are the Interpulse Intervals of an ECG Signal a Good Source of Entropy? An In-depth Entropy Analysis Based on NIST 800-90B Recomendation.
PhD candidate role:	First author
Submitted to:	Journal of Future Generation Computer System (2019)
Impact Factor:	4.639 Q1 7/103
Note:	Wholly included in the thesis in Chapter 4

Title:	Feasibility Analysis of Inter-Pulse Intervals Based Solutions for Cryptographic Token Generation by Two Electrocardiogram Sensors.
PhD candidate role:	First author
Published in:	Journal of Future Generation Computer System (2019)
Impact Factor:	4.639 Q1 7/103
URL:	`http://www.sciencedirect.com/science/article/pii/S0167739X18330784`
DOI:	`https://doi.org/10.1016/j.future.2019.02.021`
Note:	Wholly included in the thesis in Chapter 3
Statement:	Since this source is literally fully included in Chapter 3, all material from this source in the thesis is indicated by typographic means and an explicit reference in a footnote in each of the chapters in which this inclusion occurs.

Title:	A summary of: Heartbeats Do Not Make Good Pseudo-Random Number Generators.
PhD candidate role:	First author
Published in:	Actas de las Cuartas Jornadas Nacionales de Investigación en Ciberseguridad (2018).

Title:	Heartbeats Do Not Make Good Pseudo-Random Number Generators: An Analysis of the Randomness of Inter-Pulse Intervals.
PhD candidate role:	First author
Published in:	Entropy (2018)
Impact Factor:	2.305 Q2 22/78
URL:	`https://www.mdpi.com/1099-4300/20/2/94/pdf`
DOI:	`https://doi.org/10.3390/e20020094`
Note:	Wholly included in the thesis in Chapter 2
Statement:	Since this source is literally fully included in Chapter 2, all material from this source in the thesis is indicated by typographic means and an explicit reference in a footnote in each of the chapters in which this inclusion occurs.

Title:	ECG Dj: Heart Beat Synchronization.
PhD candidate role:	First author
Published in:	USENIX (2017).

Other Research Merits

Other contributions done during this PhD which are not included in this dissertation but were published in 2 international journals with IF and one book chapter are:

Title:	Weaknesses of Fingerprint-based Mutual Authentication Protocol.
PhD candidate role:	Second author
Published in:	Security and Communication Networks (2014)
Impact Factor:	0.72 Q3 100/139
URL:	https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1161
DOI:	https://doi.org/10.1002/sec.1161

Title:	Cryptanalysis of the RNTS system.
PhD candidate role:	Second author
Published in:	The Journal of Supercomputing (2013)
Impact Factor:	0.841 Q2 47/102
URL:	https://doi.org/10.1007/s11227-013-0873-3
DOI:	10.1007/s11227-013-0873-3

Title:	Security of EPC Class-1.
PhD candidate role:	Second author
Published in:	Security and Trends in Wireless Identification and Sensing Platform Tags: Advancements in RFID. IGI Global (2013)
URL:	https://www.igi-global.com/chapter/content/68739
DOI:	10.4018/978-1-4666-1990-6.ch002

Abstract

eHealth is a relatively recent term that is frequently used to refer to healthcare services making an extensive use of technology and telecommunications systems. Some examples of eHealth systems can be the electronic health record that allow different healthcare professionals to access to patient data at the same time and from different locations; the ePrescribing in which the entire process of management and control of prescriptions among patients, doctors and pharmacists is digitized, or; the Telemedicine that enables the possibility of monitoring, making diagnosis and treatment remotely to patients. eHealth can be considered as a particular case of the Internet of Things (IoT), where "things" are essentially sensors which are constantly gathering data about the medical condition of a subject. These sensors provide a smarter approach to health services making the decision-making process more accurate and effective due to the fact that patients do not need to be physically on medical centers.

It has been thought that Implantable Medical Device (IMD) such as pacemakers, insulin pumps or cochlear implants were the only devices in charge of measuring biological information. However, there are many other gadgets which can be placed on or around the human body such as smartphones, wristbands or even the smartwatches that can also be used to sense some vital signs of the bearer without interfering in her life. These devices are known as wearables and are basically smart electronic devices with different sensors inside like accelerometer, electrocardiogram, electromyograph, electroencephalogram, electrodermograph, GPS, oximeter, bluetooth proximity, pressure or thermometer that can help to extract biological information from the person wearing them.

When these sensors are placed in the same body and can share information, it is said to be part of a Body Area Network (BAN), also known as Body Sensor Network (BSN) or Medical Body Network (MBAN). The first use of the BAN was in the continuous monitor healthcare domain, especially for patients who need continuous monitoring, e.g., patients suffering from chronic diseases such diabetes, asthma or heart attacks. Nowadays, we can find other applications to improve the performance in sports, for military purposes, or as authentication mechanisms.

When BANs are provided with connectivity, it is said to be a Wireless

Body Area Network (WBAN). These kind of networks usually have a central device (also known as hub, commonly implemented by a smartphone) with Internet connectivity. Due to this connectivity, the gathered information can also be shared not only with other devices in the same network but also be sent to public servers in order to be fully accessible by different people such as medical staff, the user's personal trainer or just for private purposes.

Information gathered by a WBAN usually contains high sensitive due to the nature of the data. Therefore, the security and privacy on these networks have been identified as two of the most challenging tasks by research community. New cryptographic protocols are needed not only to protect the user's identity but also to protect the integrity of the patient's medical data.

Biometric plays an important role because it refers to identification and authentication methods by which, using biological signals, can identify or validate the identity of a person. In the last years, several works have been published on biometric authentication and identification. This kind of authentication systems have great potential because each biological trait must be universal, collectable, unobtrusive, permanent, unique and difficult to circumvent. From a technical point of view, biometrics can be classified into two main groups depending on whether the deployed system uses physiological or behavioral signals. Examples of physiological signals include fingerprint, iris, retina, heart and brain signals. On the contrary, examples of behavioral systems are voice, signature analysis or keystroke dynamics. The main reason why such signals can be easily included in authentication systems is because they exhibit a most if not all of the aforementioned features.

Interest in biometrics has gained momentum in the last years mostly due to the massive use of daily life devices like smartwatches, smartphones and laptops. This interest is not temporary, the global biometric market revenues will reach \$34.6 billion annually in 2020, especially in mobile devices.

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. It can be seen how since the first paper appeared in 2004, proposing that the heart signal might be applied to cryptography, several proposals have been published in the literature. Particularly, the use of heart signal has gained a special attraction in cryptographic application as a random number generator. Such random tokens can be used to generate a private key, as part of an authentication protocol, as an alternative to classical key establishment protocols or can be used on proximity detection protocols among others.

Heart signal contains six different peaks, known by the letters P, Q, R, S, T and U. The fiducial points are used to describe the points of interest which can be extracted from biological signal. Some examples of fiducials points of the Electrocardiogram (ECG) are P-wave, QRS complex, T-wave, R peaks or the RR-time-interval (the time distance between two consecutive R-peaks) also known as Inter-Pulse Interval (IPI) in the literature. Heart signal is a continuous signal that 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. [18] many authors have used quantization algorithms to extract different fiducial points from each Inter-Pulse Interval (IPI) due to its claimed entropy property.

The majority of the proposed works in this area conclude that the last 4-bits of each IPI can be used as a random number because of their high entropy. 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, authentication protocols or peak miss-detection algorithms. As an example, if an authentication protocol requires a 128 bit key to work, it would be necessary to acquire 32 IPIs (i.e., at least 33 consecutive R-peaks). Considering that a regular heart beats at 50-100 Bits per Minute (bpm), the key generation process would take between 20 and 40 seconds. Depending on the system where this protocol is deployed might be feasible.

Most of the proposed solutions in the literature rely on some questionable assumptions. For instance, it is commonly assumed that it possible to generate the same cryptographic token in at least two different devices that are sensing the same signal using the IPI of each cardiac signal without applying any synchronization algorithm; authors typically only measure the entropy of the Least Significant Bit (LSB) to determine whether the generated cryptographic values are random or not; authors usually pick the four LSBs assuming they are the best ones to create the best cryptographic tokens; the datasets used in these works are rather small and, therefore, possibly not significant enough, or; in general it is impossible to reproduce the experiments carried out by other researchers because the source code of such experiments is not usually available.

In this Thesis, we overcome these weaknesses trying to systematically address most of the open research questions. That is why, in all the experiments carried out during this research we used a public database called PhysioNet which is available on Internet and stores a huge heart database named PhysioBank. This repository is constantly updated by medical researchers who share the sensitive information about patients and it also offers an open source software named PhysioToolkit which can be used to read and display these signals. All datasets that we used contain ECG records obtained from a variety of real subjects with different heart-related pathologies as well as healthy people.

The first chapter of this dissertation (Chapter 1) is entirely dedicated to present the research questions, introduce the main concepts used all along this document as well as settle down some medical and cryptographic definitions. Finally, the objectives that this dissertation tackles down are described together with the main motivations for this Thesis.

In Chapter 2 we report the results of a large-scale statistical study to determine if heart signal is a good source of entropy. For this, we analyze 19 public datasets of heart signals from the Physionet repository, spanning electrocardiograms from multiple subjects sampled at different frequencies and lengths. We then apply both ENT and National Institute of Standard and Technology Statistical Test Suite (NIST STS) standard battery of randomness tests to the extracted IPIs. In particular, ENT is a suite composed of the following tests: entropy, Chi-Square, arithmetic mean, Monte Carlo, and serial correlation coefficient statistical tests. As output, ENT reports the overall randomness results after running the aforementioned tests. On the contrary, NIST STS is a suite made of fifteen statistical tests: frequency monobit and block tests, runs, longest run of ones in a block, binary matrix rank, discrete Fourier Transform (spectral) test, overlapping and non-overlapping template matching, Maurer's Universal Statistical tests, linear complexity, serial, approximate entropy, cumulative sums, random excursions and random excursions variant tests. As output, NIST STS reports a p-value which indicates whether the given sequence has passed or not each test.

We implement and reproduce the algorithm previously proposed by Rostami et al. [147] to generate and extract as many keys as possible from the cardiac signal to check the randomness property. This algorithm has the following steps: get the sampling frequency for each signal, which is available in an associated description record; Run Pan-Tomkins's QRS detection algorithm over the ECG signal to extract the R-peaks; get the timestamp of each R-peak and calculate the time difference between each pair of consecutive R-peaks to obtain the sequence of raw IPI values; apply a dynamic quantization algorithm to each IPI to decrease the measurement errors and a Grey code to the resulting quantized IPI values to minimize the error margin of the physiological parameters, and; extract the four LSB from each coded IPI value.

The results we obtain through the analysis, clearly show that a short burst of bits derived from an ECG record may seem random, but large files derived from long ECG records should not be used for security purposes.

In Chapter 3, we carry out an analysis to check whether it is reasonable or not assume that two different sensors can generate the same cryptographic token. We systematically check if two sensors can agree on the same token without sharing any type of information. Similarly to other proposals, we include Error Correcting Code (ECC) algorithms like Bose-Chaudhuri-Hocquenghem (BCH) to the token generation. These algorithms are known as fuzzy extractors and they are usually composed of two main phases: generation and reproduction.

In the generation phase, a biometric signal w is received as input and two parameters are given as output: a secret value R and a public value P. In the reproduction phase, a fresh biometric signal w' is given as input together with the public parameter P, previously generated in the generation phase. If and only if the distance between these two biometric signals—typically the Hamming distance—is less than a given threshold t_r ($Hamming(w, w') < t_r$), then the same output R will be retrieved.

We conclude that a fuzzy extractor (or another error correction technique) is not enough to correct the synchronization errors between the IPI values derived from two ECG signals captured via two sensors placed on different positions. In particular, we demonstrate that a pre-processing of the heart signal must be performed before the fuzzy extractor is applied.

Going one step forward and, in order to generate the same token on different sensors, we propose a synchronization algorithm. To do so, we include a run-time monitor algorithm based on the satisfaction of three important real-time properties: 1. the time between two consecutive peaks of each ECG signal; 2. the relative time between peaks from the different heart signals, and; 3. the total sampling time to return back a valid token. After applying our proposed solution, we run again the experiments with 19 public databases from the PhysioNet repository. The only constraint to pick those databases was that they need at least two measurements of heart signals (ECG_1 and ECG_2). As a conclusion, running the experiments, the same token can be derived on different sensors in most of the tested databases if and only if a pre-processing of the heart signal is performed before extracting the tokens.

In Chapter 4, we analyze the entropy of the tokens extracted from a heart signal according to the NIST STS recommendation (i.e., SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation). When authors check the entropy of the generated tokens, there is a subset of them who specifically claim to use the Shannon entropy. On the contrary, there are others who just say that they test the entropy, providing no more information or even there are some authors who directly do not check the entropy but run some random tests instead like the National Institute of Standard and Technology Statistical Test Suite (NIST STS). However, as Rushanan et al. [149] pointed out, this is not enough to claim that the ECG can be a good source of entropy.

In 2012, the NIST STS published a draft with some recommendations for the entropy sources used for random bit generation. The final document (NIST SP 800-90B) was recently published—early 2018—and can be seen in. 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.

In this chapter, we use the min-entropy estimators proposed by the NIST STS 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. In particular, the estimators of the min-entropy are: The most common value estimate, the collision estimate, the Markov estimate, the compression estimate, the MultiMCW predic-

tion estimate, the lag prediction, the multiMMC prediction estimate, the LZ78Y prediction estimate, the t-Tuple estimate and, the Longest Repeated Substring (LRS) estimate.

We downloaded 19 databases from the Physionet public repository and analyze, in terms of min-entropy, more than 160,000 files. Finally, we propose other combinations for extracting tokens by taking 2, 3, 4 and 5 bits different than the usual four LSBs. Also, we demonstrate that the four LSB are not the best bits to be used in cryptographic applications. We offer 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.

Finally, the last Chapter of this dissertation (Chapter 5) summarizes the main conclusions arisen from this PhD Thesis and introduces some open questions.

Keywords: Randomness, Authentication, Privacy, Implantable Medical Devices, Inter-Pulse Intervals, Biometric.

Abstract (Non-technical)

The proliferation of wearable and implantable medical devices has given rise to an interest in developing security schemes suitable for these devices and the environment in which they operate. One area that has received much attention lately is the use of (human) biological signals as the basis for biometric authentication, identification and the generation of cryptographic keys.

More concretely, in this dissertation we use the Electrocardiogram (ECG) to extract some fiducial points which are later used on crytographic protocols. The fiducial points are used to describe the points of interest which can be extracted from biological signals. Some examples of fiducials points of the ECG are P-wave, QRS complex, T-wave, R peaks or the RR-time-interval. In particular, we focus on the time difference between two consecutive heartbeats (R-peaks). These time intervals are referred to as Inter-Pulse Intervals (IPIs) and have been proven to contain entropy after applying some signal processing algorithms. This process is known as *quantization algorithm*. The entropy that the heart signal has makes the ECG values an ideal candidate to generate tokens to be used on security protocols.

Most of the proposed solutions in the literature rely on some questionable assumptions. For instance, it is commonly assumed that it possible to generate the same cryptographic token in at least two different devices that are sensing the same signal using the IPI of each cardiac signal without applying any synchronization algorithm; authors typically only measure the entropy of the LSB to determine whether the generated cryptographic values are random or not; authors usually pick the four LSBs assuming they are the best ones to create the best cryptographic tokens; the datasets used in these works are rather small and, therefore, possibly not significant enough, or; in general it is impossible to reproduce the experiments carried out by other researchers because the source code of such experiments is not usually available.

In this Thesis, we overcome these weaknesses trying to systematically address most of the open research questions. That is why, in all the experiments carried out during this research we used a public database called PhysioNet which is available on Internet and stores a huge heart database named PhysioBank. This repository is constantly being updated by medical researchers who share the sensitive informa-

tion about patients and it also offers an open source software named PhysioToolkit which can be used to read and display these signals. All datasets we used contain ECG records obtained from a variety of real subjects with different heart-related pathologies as well as healthy people.

The first chapter of this dissertation (Chapter 1) is entirely dedicated to present the research questions, introduce the main concepts used all along this document as well as settle down some medical and cryptographic definitions. Finally, the objectives that this dissertation tackles down are described together with the main motivations for this Thesis.

In Chapter 2 we report the results of a large-scale statistical study to determine if heart signal is a good source of entropy. For this, we analyze 19 public datasets of heart signals from the Physionet repository, spanning electrocardiograms from multiple subjects sampled at different frequencies and lengths. We then apply both ENT and NIST STS standard battery of randomness tests to the extracted IPIs. The results we obtain through the analysis, clearly show that a short burst of bits derived from an ECG record may seem random, but large files derived from long ECG records should not be used for security purposes.

In Chapter 3, we carry out an analysis to check whether it is reasonable or not the assumption that two different sensors can generate the same cryptographic token. We systematically check if two sensors can agree on the same token without sharing any type of information. Similarly to other proposals, we include ECC algorithms like BCH to the token generation. We conclude that a fuzzy extractor (or another error correction technique) is not enough to correct the synchronization errors between the IPI values derived from two ECG signals captured via two sensors placed on different positions.

We demonstrate that a pre-processing of the heart signal must be performed before the fuzzy extractor is applied. Going one step forward and, in order to generate the same token on different sensors, we propose a synchronization algorithm. To do so, we include a runtime monitor algorithm. After applying our proposed solution, we run again the experiments with 19 public databases from the PhysioNet repository. The only constraint to pick those databases was that they need at least two measurements of heart signals (ECG_1 and ECG_2). As a conclusion, running the experiments, the same token can be de-

rived on different sensors in most of the tested databases if and only if a pre-processing of the heart signal is performed before extracting the tokens.
In Chapter 4, we analyze the entropy of the tokens extracted from a heart signal according to the NIST STS recommendation (i.e., SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation). We downloaded 19 databases from the Physionet public repository and analyze, in terms of min-entropy, more than 160,000 files. Finally, we propose other combinations for extracting tokens by taking 2, 3, 4 and 5 bits different than the usual four LSBs. Also, we demonstrate that the four LSB are not the best bits to be used in cryptographic applications. We offer 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.
Finally, the last Chapter of this dissertation (Chapter 5) summarizes the main conclusions arisen from this PhD Thesis and introduces some open questions.

Keywords: Randomness, Authentication, Privacy, Implantable Medical Devices, Inter-Pulse Intervals, Biometric.

Resumen (No técnico)

En los últimos años, el uso de dispositivos *wearables* ha aumentando considerablemente en la población. Entre estos dispositivos podemos destacar los relojes inteligentes o las pulseras que registran la actividad física como los más usados. La característica principal de estos dispositivos es que tienen sensores capaces de medir señales vitales de la persona que los usa, como puede ser el oxigeno en sangre o el ritmo cardiaco. Antes de que aparecieran los wearables, los dispositivos médicos implantables como el marcapasos eran los únicos dispositivos capaces de medir estas señales vitales.

Por otra parte, desde que los teléfonos móviles tienen acceso a Internet, los llamados wearables pueden estar conectados con un teléfono móvil, el cual almacena la información de las señales vitales recogida por los sensores. Estos datos pueden, a su vez, ser mandados a servicios externos como a plataformas de seguimiento deportivo o servicios de telemedicina.

Debido a que la información extraída es altamente sensible, el interés de la comunidad científica para desarrollar esquemas de seguridad adecuados para este tipo de dispositivos y los entornos donde estos operan ha aumentado. En concreto, esta Tesis se centra principalmente en el uso de la señal cardiaca para extraer claves aleatorias que puedan ser usadas posteriormente en protocolos de criptografía con el fin de garantizar tanto la seguridad y privacidad del usuario como la integridad de los datos registrados por los sensores.

La señal cardíaca se suele representar a través del Electrocardiograma (ECG), donde se puede ver la actividad eléctrica del corazón en función del tiempo. El ECG es una señal continua que contiene seis picos, conocidos por las letras P, Q, R, S, T y U. Se ha demostrado en investicaciones previas que la diferencia temporal—intervalo—entre dos picos R consecutivos de la señal cardica, conocido como Inter-Pulse Intervals (IPIs), contiene cierta entropía, haciendo que este intervalo sea un posible candidato para generar números aleatorios.

Sin embargo, la mayoría de las soluciones propuestas en este área asumen como ciertas algunas hipótesis que no se han sido investigadas a fondo por la comunidad científica. Por ejemplo, los mecanismos que se utilizan para comprobar la aleatoriedad de las claves generadas a través de la señal cardíaca solamente tienen en cuenta la entropía; se asume que es posible generar la misma clave desde dos dispositivos

diferentes que están midiendo la misma señal sin aplicar ningún algoritmo de sincronización, o; se considera que los cuatro bits menos significativos son la mejor opción para generar las claves aleatorias.

En esta Tesis, se ha investigado la veracidad de dichas hipótesis y si son aplicables a esquemas reales de seguridad. Además, como los trabajos realizados por otros investigadores previamente usaban bases de datos pequeñas y el código fuente de sus experimentos no esta disponible. En los experimentos llevados a cabo durante esta investigación se han utilizado bases de datos públicas y heterogéneas (tanto en tamaño como en la variedad de las patologias de los pacientes) que contienen registros de señales ECG obtenidas del repositorio PhysioNet, la cual almacena una enorme base de datos de señales fisiológicas. Además, se han hecho público el código fuente necesario para reproducir los experimentos realizados.

El primer capítulo está completamente dedicado a introducir y explicar los conceptos utilizados a lo largo de este documento así como definiciones médicas, criptográficas o la naturaleza de los repositorios públicos de señales cardíacas. También, se describen los objetivos junto con las principales motivaciones para esta Tesis.

El segundo capítulo presenta los resultados de analizar en detalle si la señal cardíaca es una buena fuente de entropía y se puede usar para generar claves aleatorias. Para ello, descargamos 19 bases de datos públicas de señales cardíacas del repositorio Physionet. Generamos las claves a partir de los IPIs del ECG y posteriormente aplicamos los tests estadísticos ENT y NIST STS para determinar la aleatoriedad de los IPIs extraídos. Los resultados obtenidos muestran claramente que cuando la señal es corta——en el tiempo, los claves peden parecer aleatorias. Por el contrario, cuando se obtienen muchas claves pertenecientes a la misma señal cardíaca, entonces las claves no son aleatorias y por tanto no deben usarse con fines de seguridad.

En el tercer capítulo, se comprueba si es razonable o no asumir que dos sensores diferentes midiendo la misma señal cardíaca pueden generar la misma clave criptográfica. Para ello, usamos 19 bases de datos del repositorio Physionet con, al menos dos mediciones de señales cardíacas (ECG_1 y ECG_2). Inicialmente comprobamos que las claves generadas de ambas señales eran muy diferentes. Con el objetivo de generar la misma clave, propusimos un mecanismo de sincronización de la señal previo a la extracción de claves. Tras analizar de nuevo las claves obtenidas con el mecanismo de sincronización propuesto,

fuimos capaces de generar la misma clave desde diferentes sensores en la mayoría de las bases de datos.
En el capítulo cuatro, analizamos la entropía de las claves Inter-Pulse Intervals (IPIs) extraídas de la señal cardíaca de acuerdo con la recomendación NIST STS (SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation). Para ello descargamos 19 bases de datos de señales cardiacas del repositorio público Physionet y analizamos más de 160.000 claves en términos de min-entropía. Tras el análisis, demostramos que los cuatro bits menos significativos no son los mejores bits para ser utilizados en aplicaciones criptográficas y ofrecemos otras combinaciones alternativas para dos (ej. 87), tres (ej. 638), cuatro (ej. 2638) y cinco (ej. 23758) bits que, en general, son mucho mejores que utilizar los cuatro bits menos significativos desde el punto de vista de la entropía.
Finalmente, en el último capítulo de la Tesis se resumen las principales conclusiones y se introducen algunas líneas futuras.

Keywords: Aleatoriedad, Autenticación, Privacidad, Dispositivos Médicos Implantables, Inter-Pulse Intervals, Biometría.

Contents

List of Figures

List of Tables

1

Introduction

Biometric is defined as the unique physiological or behavioural features of human body and thus, they can be used to identify persons. According to some authors, biometrics should have a set of properties in order to be practically usable: acceptable, circumventive, collectible, convenient, costless, permanent, performable, precise, simple, storable, unique and universal [40, 82]. These properties are explained in more detail as follows:

Acceptability There must be an agreement between persons and technology in such a way that people have to allow their personal biometric traits to be captured and evaluated.

Circumvention Which shows how easily the system can be imitated using fraudulent methods.

Collectability The biometric trait must be gathered quantitatively, i.e., each biometric feature should be gathered in order to authenticate it.

Convenience The process of both measuring and storing the features should be done in real time, i.e., should not be a time-consuming task.

Cost The prize of storing biometric traits should be as cheaper as possible.

Permanence It is crucial that the chosen trait should be sufficiently invariant over a period of time, otherwise the original biometric feature musst be periodically renewed.

Performance This property is not only related to the speed but also to the accuracy, and robustness of technology used.

Precision Every feature should be different enough from every other feature.

Simplicity The recording and transmission of the feature should be easy enough to avoid errors.

Storability The biometric trait should be storable.

Universality This property ensures that everybody using a biometric system should possess at least one particular trait that is common for all the people.

Uniqueness Once the biometric feature is common for all the people, the uniqueness property guarantees that the trait should be sufficiently different so that users can be distinguished between them.

Biometrics is split into two main groups according to the nature of the features: behavioral and physical.

Behavioral Biometrics Behavioral biometrics is an area of study in charge of measuring patterns in human activities which make people uniquely identifiable.

Some examples of behavioral biometric features are:

Gait Analysis The way people walk together with image or video analysis or by using the accelerometer of the smartphone [173] can be used to identify people. This is known as gait recognition [203]. More information about the phases and cycles of the human walk can be seen in [89].

Keystroke Dynamics The goal of this biometric system is to identify persons by monitoring the keyboard inputs and thus generating patterns about the way users type [118]. Some applications can be found in [23, 90] and some state-of-the-art surveys in [87, 171].

Mouse Dynamics Mouse distance between two points on the screen, the way the mouse is moved, the drag and drop of the elements or the time that the mouse is idle can be used to identify and authenticate people [51, 165]. A review performed in [83] summarizes the main contributions performed in this area about mouse dynamics authentication.

Signature Analysis Signature is the most extended biometric mechanism for people authentication [197, 47] due to devices like touchpads or digital pens [109].

Voice ID This trait is based on multiple voice characteristics such as vocal tracts, mouth, nasal cavities or lips to authenticate people. However, despite this it is not a biometric trait which can be used in large and scalable systems [82], it is a common technique to be used together with other systems like fingerprint [30].

.

Physical Biometrics Physical biometrics are related to the static traits of a human body that are not subject to change during the time over aging. Some examples of physical biometric features are:

Face Face trait is another wide extended and well known biometric characteristic used nowadays in authentication systems. Eyes, eyebrows, nose, lips and chin, are commonly used by measuring their spatial relationships to authenticate people [82].

Fingerprint This trait refers to a flowing pattern on the fingertip of an individual consisting of ridges and valleys [82]. It can be acquired by using different technologies as optical sensors or total internal reflection sensors. Fingerprint recognition systems have been incorporated into a number of forensic, civilian and commercial applications [111].

Hand and finger geometry Hand geometry recognition systems are based on a number of measurements taken from the hand, including its shape, size of palm, as well as lengths and widths of the fingers [82]. Hand shape biometrics is attractive due to the fact that it can be captured in a relatively user convenient, shape information requires only low resolution images [48].

Heart Before Bao et al. [18], Poon et al. [144] and Bao et al. [17] proposed in 2004, 2006 and 2008 respectively, different protocols to secure BANs where the authors claimed that the ECG signal can be used for biometric purposes. Before these works, there was a believe that the ECG signal could not be used as biometric trait [36]. More detail are given in the following Section.

Iris The complex iris texture carries very distinctive information useful for personal recognition [82, 193]. It contains around 266 visible patterns, which forms the basis of several recognition algorithms. Even on the same person, left and right irises are different but they are unique to an individual and is stable with age [53].

Retina A retina-based biometric involves analyzing the layer of blood vessels situated at the back of the eye. It is claimed to be the most secure biometric since it is not easy to change or replicate the retinal vasculature [82]. Retinal scanning can be quite accurate but does require the user to look into a receptacle and focus on a given point. For these reasons, retinal scanning is not warmly accepted by all users, even though the technology itself can work well [105].

Nowadays, the primary application of biometrics is to control access to secure locations [105]. Biometric systems are deployed in office buildings [127], casinos [148, 181], health environments [207, 98], immigration service [182], border control airports [1, 2], seaports [1], border-crossing [1, 41] or security of in bank systems [188, 72] among many other examples.

1.1 Heart Signals in Biometrics

This dissertation is focused on heart signals obtained via ECG, Photoplethysmographic (PPG) or Blood Pressure (BP) and, more concretely on electrical heart signals extracted from the ECG. Traditionally, these signals have been used for medical diagnostics, but recently the use of the heart signals has been applied into security and privacy protocols [7, 8, 17, 34, 136, 147, 158, 160, 162, 169, 185, 200, 205].

The main use of heart biometric signal is generate random numbers [196] as a Pseudorandom Number Generators (PRNG) and these numbers usually are part of key generation protocols in authentication procedures [147, 160, 208].

A crucial part of many cryptographic system is the generation of random numbers in order to provide high-quality of randomness and to guarantee the safety and reliability of the security system. In other words, the quality of the random number generator directly influences how difficult it is to attack the system.

Therefore, in order to check if a generated numbers can be considered random, entropy is used to describe the amount of randomness available. However, there are some public suites like ENT, NIST STS, DieHard tool or TestU01 software that are can be used to evaluating the randomness property [20].

Recent works have demonstrated that ECG signals can be also used as a source of entropy for security purposes [7, 101, 160, 206]. In particular, this is done by calculating the IPI which is the time-interval between two consecutive R-peaks of the ECG.

Concretely, these authors have claimed that the LSBs of the IPI contain a high degree of entropy [7, 8, 34, 136, 147, 158, 160, 162, 169, 185, 200, 205]. Therefore, this property of ECG is used in IPI-based authentication, identification, and key generation protocols (e.g., [8, 158, 162, 185]). Also, the majority of the proposed works in this area, e.g., [8, 17, 158, 162, 185, 205], conclude that the last 4-bits of each

IPI have more entropy.
Nowadays, it is possible to find a myriad of devices equipped with dedicated sensors to measure the heart signal, apart from the common medical electrodes that record the ECG signal i.e., a IMD such as pacemakers, there are some other wearable devices with PPG sensors which record the blood pressure to get heart beats, i.e., wristbands or even smartwatches.
Those sensors are include in the ambit of IoT and eHealth. Which constantly gathering information about the medical condition of a subject. Also, always every sensor are part of a BAN. Information gathered by these networks contain highly sensitive data and provide security and privacy is one of the most challenging tasks by the research community [68, 134, 178].

1.2 ECG Signal Processing

In Figure 2.2 can be seen the typical shape of an ECG signal. In it, the most representative characteristics of the signal can be seen. Peaks of the waves like P, Q, R, S, T and U as well as the segments and the intervals—which is the time when the waveform starts and ends. It is worth mentioning that the QRS complex is the most characteristic waveform of the ECG signal [113] and it starts when the Q wave begins and ends when the S waveform finishes.
When using the ECG signal for Biometrics or cryptographic purposes we essentially found two main ways of proceeding, 1) authors who calculate directly the IPIs of the ECG, or; 2) authors who extract some fiducial points by using a delineation algorithm. In the following we will explain in more detail each procedures to extract information that is used afterwards for cryptography.

1.2.1 IPIs and Quantization Algorithms

In 1999, Juels and Wattenberg introduced the term *fuzzy commitment* [84]. In this cryptographic scheme for biometrics, authors proposed to extract a key from a biometrical signal (process known as *fuzzy extractor*). A fuzzy extractor is a mathematical function f which takes a biometrical signal w and produces both a random string R and a public parameter P. What makes fuzzy extractors particularly interesting for biometrics is that, when the input changes slightly, i.e.,

$w' = w + \epsilon$, the random output R remains invariant if and only if the distance between these two biometric signals—typically the Hamming distance—is less than a given threshold t_r, i.e., $Hamming(w, w') < t_r$ [46].

Between 2001 and 2003 two papers [125, 172] analyzed the correlation of the time interval between R-peaks, also known as Inter-Pulse Interval (IPI), of the ECG and the Heart-Rate Variability (HRV). Authors reached to the conclusion that time between R-R peaks and the HRV can be considered to be similar when the patient is resting. Additionally, authors demonstrated that after doing exercise, neither the R-R nor the P-P intervals can be considered similar to the HRV.

Given these results aforementioned described, Bao et al. [18] published one of the first papers, if not the first one, in 2004 proposing the ECG to be used as authentication mechanism for biometrics by using the time interval between consecutive R-peaks. Since then, several authors have used this time interval between consecutive R-peaks to generate random numbers. The process of transforming the ECG, which is a continuous signal, into a discrete one is known as *quantization* or as *dynamic quatization.*

1.2.2 Delineation of ECG

Since most of the clinically information that the ECG has, is in the intervals and the amplitudes of the wave peaks and boundaries, an algorithm must be used to extract such information from the ECG. This procedure is known as delineation [113] and it is still challenging nowadays. The first step to extract the fiducial points is to get an R-peak [92]. After that, the QRS complex and the P and T waves can be delineated. This is what most of the algorithms do and from there, forward and backward seek windows can be defined and finally some other techniques to enhance the needed waves and fiducial points can be applied [112]. However, as Martinez et al. [113] pointed out, the detection of the wave onset and offset directly from the ECG is not trivial due to the signal amplitude is significantly low in comparison to the wave boundaries. In addition to that, the noise level can also be higher than the signal itself.

Both, in medical research as well as in other fields of engineering there can be found several ways to calculate the delineation of the ECG. Mathematical models [108], Wavelet Transforms (WTs) [4, 28, 146], hidden markov models [4, 60] or artificial intelligence algorithms [103, 115, 152] are just a few examples about how to obtain some fiducial points from the ECG. However, in cryptography we could exclusively find authors who use WTs.

1.2.2.1 Wavelet Transforms

The Wavelet Transform (WT) provides a description of the signal. By applying a linear transform, the WT decomposes a signal into different components at different scales, i.e., it decomposes the signal at a different frequencies. More formally, a wavelet is essentially used to refer to a family of basis functions of the Hilbert space $L^2(\mathbb{R}^n)$, generated from a finite set of normalized functions ψ_i where i is chosen from a finite set I, and from two operations: scaling (a), and translation (b). More concretely, the WT of a signal $x(t)$ is:

$$W_a x(b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt,\ a > 0 \tag{1.1}$$

We can distinguish between two WTs: Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). CWT is typically generated by letting both the translation and scale operations vary continuously. On the contrary, DWT uses a pair of filters to successively isolate both low and high pass components of a signal [174]. Hence, due to the non-stationary nature of the ECG, the DWTs are specially suitable for such a signal and the continuous repetition of its patterns/waveforms, e.g., QRS complex or P and T waves, at different frequencies [113, 15].

Martínez et al. [113] provides a detailed explanation about how, by taking as a prototype wavelet (ψ_i) a smoothing function, discretizing either or both parameters a or b and taking a dyadic grid on the time-scale plane such that $a = 2^k$ and $b = 2^k l$, where k controls the dilation or translation and l the position of the wavelet function, then the transform is then called *dyadic wavelet transform*, with basis functions:

$$\psi_{k,l}(t) = 2^{-k/2}\psi(2^{-k}t - l);\ \text{where}\, k, l \in \mathbb{Z}^+ \tag{1.2}$$

Roughly speaking, the main idea behind the DWT is that it analyses the signal at different resolution through the decomposition of the signal into several successive frequency bands. To do so, two set

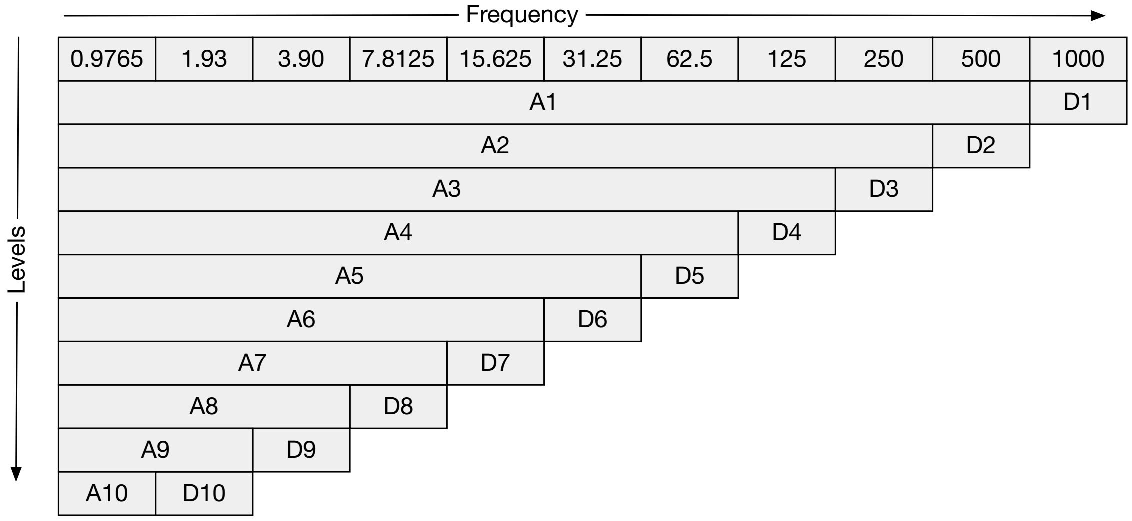


Figure 1.1: DWT decomposition of a signal sampled at 1 kHz.

functions are typically used: $\psi_{k,l}(t) = 2^{-k/2}\psi(2^{-k}t - l)$ and $\phi_{k,l}(t) = 2^{-k/2}\psi(2^{-k}t - l)$, normally linked with the low pass and the high pass filters respectively [43].
Let us explain the DWT concept with an illustrative example when applied to ECG signals [15]. In Figure 1.1 we have included a decomposition using wavelets of a signal $x(t)$ sampled at 1kHz (like iafdb [139] or ptbdb [24] databases). This decomposition is repeated in order to increase the frequency resolution as well as the approximation coefficients decomposed with both high ($\psi_{k,l}$) and low ($\phi_{k,l}$) pass filters. Note that Figure 1.1 can also be represented as a binary tree, known as filter bank [52], where nodes represent sub-spaces with different time-frequencies of the signal.

We have included a summary of the surveyed publications along with the number of extracted information in bits, the year when they where published together with the extraction algorithm used in the papers can be seen in Table 1.4.

1.3 Data Acquisition

In this section we detail how the surveyed papers test their proposals with respect to the heart signals. In particular, we found two main groups: i) authors that use proprietary data, and; ii) authors who use data from public repositories.
Regarding the first point, in general, it is impossible to test the cor-

rectness of the papers based on proprietary data since authors did not make public those data. We argue that in research all the data should be public in order to allow others to reproduce the same results and start from that point. In other words, let the science improve. If repositories are private, not only researchers have to repeat the same experiments once and again but also different results can be found for the same problem and, as far as we know, there is nothing to compare with, i.e., no one can be sure if the results extracted from the experiments are correct or better than others or not.

On the other hand, we found in the literature three main public repositories that authors have used in their research: the Telemetric and ECG Holter Warehouse Project (THEW) [42], the Biosec [3], and; the Physionet [57] datasets. Due to the amount of researchers who use Physionet is in comparison to the other two mentioned databases (see Table 1.1), we are significantly giving more information about this repository, explaining its composition and how the data should be read for future experiments on this field. Nevertheless, we still give some brief descriptions about the other two datasets like how to retrieve the data as well as the structure.

THEW Thew is a public repository created and maintained by the University of Rochester Medical Center and it is available at `http://thew-project.org/index.htm`. This repository is made of 22 main databases which go from healthy young ECGs to heart information about people with heart or renal diseases (see Table 1.2 for a complete description of the repository). What makes this database particularly interesting is the amount of healthy people that it has. concretely, this repository has two databases named "E-OTH-12-0689-025" and "E-HOL-03-0202-003" with ECGs of 689 and 202 healthy patients respectively. In most of the databases of this repository, files are given in both *.cpp* and *.m* formats to be managed under C++ and Matlab respectively. Additionally, authors have published some other features to read the ECG and the annotation files in Matlab directly from their server[3].

[1] It is not specified in the paper

[2] Authors claim they use 294 patients but qtdb only consists of 105 patients

[3] More info: `http://thew-project.org/THEWFileFormat.htm`

	Database(s)
Altop et al. [8]	50 subjects from the MIMIC II Waveform
Bao et al. [17]	99 subjects from a private repository
Bao et al. [16]	14 healthy subjects from a private repository
Camara et al. [28]	private repository and 202 patients from E-HOL-03-0202-003 (THEW dataset)
Hong et al. [71]	10 subjects from private dataset
Karthikeyan et al. [88]	47 subjects from mitdb and 23 from afdb
Kim et al. [91]	PhysioNet[1]
Koya et al. [93]	ptbdb and mitdb
Moosavi et al. [124]	15 subjects from mitdb
Pirbhulal et al. [143]	25 healthy from a private repository; 20 from mitdb, and; 44 with cardiac diseases from a private repository
Rostami et al. [147]	47 subjects from mitdb; 290 from ptbdb, and; 250 from mghdb
Seepers et al. [160]	48 subjects from mitdb and fantasia database
Seepers et al. [162]	42 subjects from mitdb and fantasia, and; 111 subjects from BioSec dataset
Vasyltsov et al. [184]	private dataset
Xu et al. [195]	Physionet[1]
Yao et al. [196]	294 patients from qtdb[2]
Zaghouani et al. [198]	36 subjects from nsrdb and 48 from mitdb
Zhang et al. [200]	20 subjects from a private dataset and 64 from edb
Zheng et al. [206]	18 from nsrdb; 79 from edb; 47 from mitdb, and; 23 afdb
Zheng et al. [205]	18 subjects from nsrdb and 79 from edb

Table 1.1: Databases taken by authors.

BioSec Biosec is a repository created and maintained by researchers from Toronto University by which users who want to download any of the 7 databases must make a material transfer agreement. This repository is available at `https://www.comm.utoronto.ca/~biometrics/databases.html` and it is essentially composed of databases of healthy people. The specifications of this repository are in contradiction in the official website so we decided not to include them as a table.

Physionet It is a public repository on Internet which stores a huge heart database named PhysioBank. This repository is constantly updated by medical researchers who share the sensitive information about patients—without disclosing any other relevant information that can link the real patients, and it also offers an open source software named PhysioToolkit which can be used to read and display these signals. At the time of writing, Physionet has more than 75 databases classified into two main families: clinical databases (include demo-

Database	**SID**	**Leads**	**Sampling**	**ECGs**	**Patients**	**Size**
Acute Myocardial Infarction	E-HOL-03-0160-001	3	200 Hz	160	93	15.2GB
Coronary Artery Disease	E-HOL-03-0271-002	3	200 Hz	271	271	26.2GB
Healthy	E-HOL-03-0202-003	3	200 Hz	202	202	19.2GB
Thorough QT study #1	E-HOL-03-0102-005	3	200 Hz	102	34	4.5GB
Thorough QT study #2	E-HOL-12-0140-008	12	1,000 Hz	140	70	267GB
Torsades de Pointes (TdPs)	E-OTH-12-0006-009	12	180 Hz	6	6	1.3GB
Sotalol IV and History of TdPs	E-OTH-12-0068-010	12	1,000 Hz	68	34	244MB
AF and cardioversion	E-OTH-12-0073-011	12	1,000 Hz	73	73	1.7GB
Chest Pain (IMMEDIATE LR ECG)	E-HOL-12-1172-012	12	180 Hz	1172	1154	338GB
Genotyped Long QT syndrome	E-HOL-03-0480-013	2 or 3	200 Hz	480	307	43.2GB
Chest Pain (IMMEDIATE HR ECG)	E-HOL-12-0171-014	12	1,000 Hz	171	171	296GB
Exercise testing and perfusion imaging	E-OTH-12-0927-015	12	1,000 Hz	927	927	23GB
ESRD patients during and after hemodialysis	E-HOL-12-0051-016	12	1000 Hz	51	51	187GB
FDA1- quinidine, verapamil, ranolazine, dofetilide	E-OTH-12-5232-020	12	1000 Hz	5232	22	1.7GB
FDA2- quinidine, verapamil, ranolazine, dofetilide	E-HOL-12-0109-021	12	1000 Hz	109	22	231GB
AF Conversion	E-OTH-12-0089-022	12	1000 Hz	26	26	3.23GB
“Strict” LBBB	E-OTH-12-0602-024	12	1 kHz	602	602	157MB
Young Healthy	E-OTH-12-0689-025	12	500kHz	689	689	207MB
Collaborative studies (require the submission of a research proposal to an ad-hoc THEW committee)						
DEFINITE Study (NorthWestern Univ.)	E-HOL-03-0401-017	3	500 Hz	401	236	110GB
Occluded Artery Trial (Stony Brooke Univ.)	E-OTH-03-0802-018	3	500 Hz	802	223	6GB
Quinidine (AZCERT)	E-OTH-12-2365-019	12	500 Hz	2423	24	17MB
IQ-CSRC	E-HOL-12-0118-023	12	1000 Hz	118	20	22.8MB
TOTAL						**15TB**

Table 1.2: Summary of the THEW databases. SID refers to the unique ID in the repository; Leads refers to the number of leads acquiring in the most recordings; Sampling refers to the sampling frequency of the ECG waveforms; ECGs is the number of ECG recordings; Patients is the number of subjects involved, and; Size is the storage space of the database.

graphics, vital sign measurements made at the bedside, laboratory test results, procedures, medications, caregiver notes, images and imaging reports, and mortality) and waveform databases (high resolution continuous recordings of physiological signal). Each family has different categories: biomedical, brain or cardiopulmonary signals. Additionally, the health condition of the patients varies considerably: healthy people, heart diseases, apnea or epilepsy condition among others.

When focusing on the structure of the files, we can have three main different files: 1) header files (*.hea*). This files contain the metadata of the record. 2) signal annotations (*.atr*). This files contain the annotations of the biometrical data, and; 3) biometrical data (*.dat*). This files have all the gathered personal information the patient. Apart from those files, it is quite frequent to have a `RECORDS` file where all the names of the files are written down as well as a file named `ANNOTATORS` where the information of how to read the *.atr* files is explained.

We have included in Table 1.3 all the databases we found all along the literature together with the number of patients that each one of the databases is composed of (column 2), if more than once ECG channel is provided (column 3) and a description that Physionet provides for each one of them (column 4).

When Physionet is used on multiple scientific articles, we found that authors use arbitrary databases. There is no formal framework, rules or set of tests that authors should run in order to test and compare their proposals with others. Hence, it is hard to objectively say which one really solves what problem. Just to cite a particular example, Zheng et al. [205] compare their proposal with Zhang et al.'s work [200] in terms of the NIST STS performance. However, this comparison is far from being objective since authors use both different samples and different random tests. Despite of using the same database—edb, they do not use the same number of patients apart from the fact that Zhang et al. use 20 patients from a private repository whereas Zheng et al. use 18 subjects from nsrdb.

1.3.1 Amount of Information per IPI

We can classify surveyed papers according to the number of bits of the IPIs that authors take. We found in the literature a wide disparity in this matter. There are authors who: i) do not mention the number of taken bits [196, 198]; ii) take 2 bits [71]; iii) vary the number of taken

Database	**Files**	**Leads**	**Sampling**	**Heart condition**
aami-ec13 [76]	10	1	720 Hz	Tachycardia
afdb [55]	23	≥2	0.1 Hz ∼ 40 Hz	Atrial fibrillation
afpdb [119]	300	≥2	128 Hz	Paroxysmal atrial fibrillation
ahadb [77]	2	≥2	250 Hz	Healthy and ventricular ectopy
apnea-ecg [135]	77	1	100 Hz	Tachycardia
cebsdb [54]	60	≥2	5,000 Hz	Healthy
cdb [121]	53	x	250 Hz	Holter recordings
cudb [128]	9	x	250 Hz	Ventricular problems
edb [170]	90	≥2	250 Hz	Myocardial and hypertension
fantasia [80]	40	1	250 Hz	Healthy
iafdb [139]	32	≥2	1,000 Hz	Atrial fibrillation or flutter
incartdb [140]	75	≥2	257 Hz	Coronary artery disease
ltafdb [138]	84	≥2	128 Hz	Paroxysmal
mimic2wdb [151]	25328	≥2	125 Hz	Intensive Care Unit (ICU)
mitdb [120]	48	≥2	360 Hz	Arrhythmia
mghdb [192]	202	3	360 Hz	Unstable patients in ICU
nsrdb [141]	18	≥2	128 Hz	No significant arrhythmias
nstdb [122]	15	≥2	360 Hz	Mitdb with noise
prcp [122]	10	≥2	250 Hz	Healthy
ptbdb [24]	545	14	1,000 Hz	Myocardial and Healthy controls
qtdb [96]	105	≥2	250 Hz	Holter recordings
sddb [61]	22	≥2	250 Hz	Arrhythmia
shareedb [116]	139	≥2	128 Hz	Hypertension
slpdb [74]	18	≥2	250 Hz	Sleep apnea syndrome
stdb [6]	28	2	360 Hz	Stress tests
svdb [63]	70	≥2	128 Hz	Partial epilepsy
szdb [5]	7	1	200 Hz	Partial epilepsy
twadb [123]	100	≥2	500 Hz	Myocardial problems
vfdb [62]	22	≥2	250 Hz	Tachycardia

Table 1.3: Summary of the databases.

bits [16, 200]; iv) take 4 bits [8, 91, 93, 147, 160, 195]; v) take between 5 and 8 bits [184, 206, 17, 162]; vi) take 16 bits [143, 205], or; vii) take more than 16 bits [28, 88].

As it can be seen in Table 1.4, the majority of the surveyed papers extract 4 bits of information per IPI, however, this tendency is changing since 2016 in favour of extracting more information nowadays. Only two [91, 93] of the last published papers in this field still extracted 4 bits whereas others [124, 162, 143, 28, 88] have started to use more information from the heart signal.

1.4 Motivation and Objectives

2/4/19 15:42

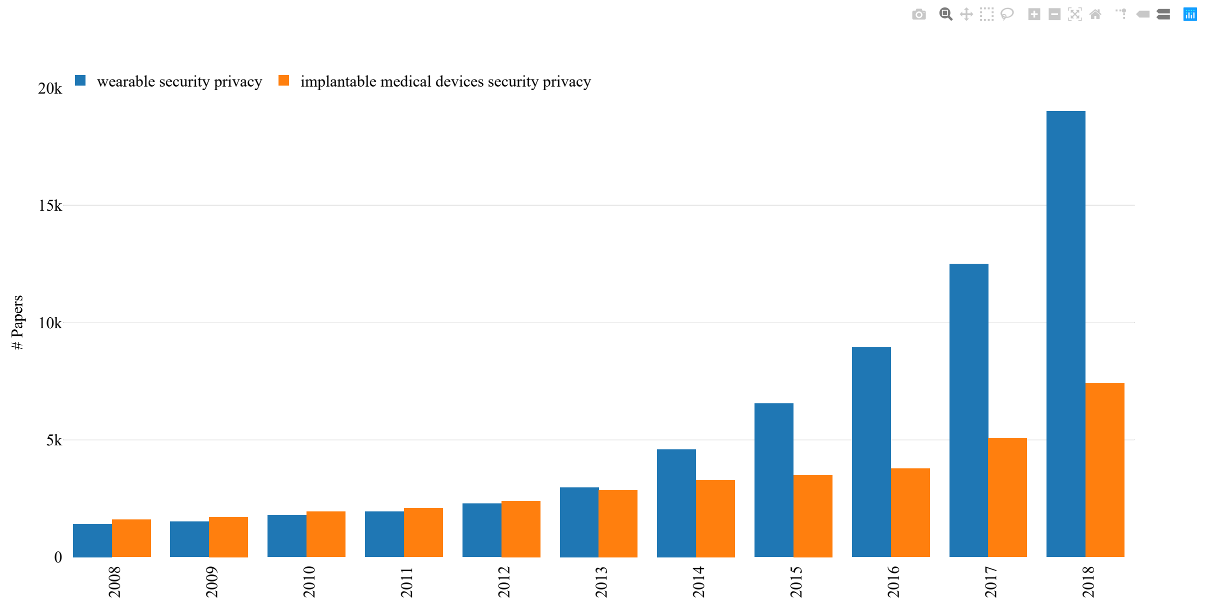


Figure 1.2: Evolution of the number of papers published in Google Scholar [75] about security and privacy in implantable medical devices or wearables.

file:///Users/pablop/Documents/devel/histogram/intro-bar.html

Página 1 de 1

Security and privacy issues have been described as two of the most challenging problems of IMDs and, more generally, wearables [134, 68, 178]. As can be seen in Figure 1.2 researchers have increased their efforts in studying security and privacy in wearable and implantable medical devices.

In this dissertation three main topics are covered in depth. First, we systematically evaluate the heart signal, and more concretely, the ECG as a source of random numbers to be used on cryptography protocols, concluding that the four LSBs of each IPI should no be considered random (see Chapter 2). Second, we evaluate if the common assumption

	Year	Information per IPIs
Camara et al. [28]	2018	23 bytes
Karthikeyan et al. [88]	2018	4 chunks of 16 bits
Pirbhulal et al. [143]	2018	16 bits
Kim et al. [91]	2018	4 bits
Koya et al. [93]	2018	4 bits
Moosavi et al. [124]	2017	8 bits
Seepers et al. [162]	2017	8 bits
Zheng et al. [205]	2016	16 bits
Vasyltsov et al. [184]	2016	5 bits
Altop et al. [8]	2016	4 bits
Zheng et al. [206]	2015	7 bits
Seepers et al. [160]	2015	4 bits
Zaghouani et al. [198]	2015	No info is provided
Bao et al. [16]	2013	3 to 4 bits
Rostami et al. [147]	2013	4 bits
Zhang et al. [200]	2012	2 to 11 bits
Xu et al. [195]	2011	4 bits
Hong et al. [71]	2011	2 bits
Yao et al. [196]	2011	No info is provided
Bao et al. [17]	2008	8 bits

Table 1.4: Summary of the surveyed papers and the number of bits extracted from IPIs.

about if two sensors placed on different locations around the human body can derive the same cryptographic key is realistic or not. We conclude that a synchronization of the signal is needed before extracting the random token (see Chapter 3). Finally, we evaluate the heart signal as a source of entropy and more concretely, if the four LSBs are the best bits to generate token with high entropy, concluding that there are many other bits which can be taken to generate better tokens (see Chapter 4). In particular, this Thesis is focused on the following main objectives:

O1. Analyze if heart signals can be used to generate random numbers.

O2. Evaluate if two different sensors can derive the same token from the heart signal.

O3. Propose a solution to generate the same token in two different sensors.

O4. Appraise if heart signal can be considered a good source of entropy.

1.5 Main Contributions & Organization

During this PhD several contributions in the field of security and privacy services based on biosignals for implantable and wearable devices have been published. As a result of the achievement of the aforementioned objectives two main contributions have been accomplished:

C1. The first contribution of this PhD can be read in Chapter 2. In it, we report the results of a large-scale statistical study to determine whether such an assumption is (or not) upheld. For this, we analyze 19 public datasets of heart signals from the Physionet repository, spanning electrocardiograms from 1,353 subjects sampled at different frequencies and with lengths that vary between a few minutes and several hours. We believe this is the largest dataset on this topic analyzed in the literature. We then apply a standard battery of randomness tests to the extracted IPIs. Under the algorithms described in this chapter and after analyzing these 19 public ECG datasets, our results raise doubts about the use of IPI values as a good source of randomness for cryptographic purposes. This has repercussions both in the security of some of the protocols proposed up to now, and also in the design of future IPI-based schemes.

C2. In Chapter 3 we address the problem of how two devices that are sensing the same heart signal can generate the same cryptographic token by extracting them from the IPIs of each cardiac signal. Our analysis is based on the use of a run-time monitor, which is extracted from a formal model and verified against predefined properties, combined with a fuzzy extractor to improve the final result. We first show that it is impossible, in general, to correct the differences between the IPIs derived from two captured ECGs signals when using only error correction techniques, thus being impossible to corroborate previous claims on the feasibility of this approach. Then, we provide a large-scale evaluation of the proposed method (run-time monitor and fuzzy extractor) over 19 public databases from the Physionet repository containing heart signals. The results clearly show the practicality of our proposal achieving a 91% of synchronization probability for healthy individuals. Additionally, we also conduct an experiment to check how long the sensors should record the heart signal in order to generate tokens of 32, 64 and 128 bits. Contrarily to what it is usually assumed (6, 12, and 24 seconds for individuals with a heart rate of 80 beats-per-minute), the sensors have to wait 13, 28 and 56.5 seconds on median, respectively, to derive the same token from both sensors.

C3. In Chapter 4 we answer three questions: 1) Is the heart signals a good source of entropy? 2) Are the four LSBs the best ones to create the best token from the entropy point of view? 3) Are there any other possible combinations of bits that achieve more entropy than taken the four LSBs? In our analysis we do a rigorous and in-depth study, analyzing cardiac signals from more than 160,000 files from 19 databases of the Physionet public repository following the NIST 800-90B recommendation. 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). We offer 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 o view.

Finally, this dissertation summarizes the main conclusions arisen from this PhD Thesis and introduces some open questions in Chapter 5.

2

Heartbeats Do Not Make Good Pseudo-Random Number Generators: An Analysis of the Randomness of Inter-Pulse Intervals

2.1 Introduction

eHealth is a relatively novel term that is commonly used to refer to healthcare services delivered through—or making an extensive use of—technology and telecommunications systems. eHealth can be seen as a special subset of the IoT, where "things" are essentially sensors which are constantly gathering information about the medical condition of a subject. Additionally, when these sensors are placed in, on, or around the human body to monitor anywhere and anytime vital signs of the bearer, it is said to be part of a BAN[1]. BAN devices can communicate with a central device (also known as hub, which is commonly implemented by a smartphone) with Internet connectivity, and in a near future all these devices will be able to interact directly between each other.

This chapter is based on this [130] publication

[1] Also known as a BSN

Information gathered by a BAN, which may contain highly sensitive data privacy-wise, is usually shared with other devices in the network and can also be sent to public servers in order to be accessible by different people such as medical staff, the user's personal trainer or just for private purposes. It has been thought that IMD such as pacemakers, insulin pumps or cochlear implants were the only devices in charge of measuring biological information. However, there are many other gadgets such as smartphones, wristbands or even the smartwatches that can be used to sense some vital signs of the bearer without interfering in her life.

Secure this network and the gathered sensitive data has been identified as one of the most challenging tasks by the research community [134, 68, 178] before deploying it in a real scenario. As an example, imagine that if someone who is equipped with sensors whose information is shared via wireless, it could be easy for an attacker to sniff the communication channel in order to listen to the transmitted packages and get some knowledge about the bearer. Therefore, new cryptographic protocols are needed not only to protect the user's identity but also to protect the integrity of the patient's medical data [37, 106].

Biometrics refer to identification and authentication methods that, using biological signals, can identify or validate the identity of a person. In the last years, several works have been focused on biometric authentication and identification [147, 160, 208]. This kind of authentication systems have a great potential because each biological trait must be universal, collectable, unobtrusive, permanent, unique and difficult to circumvent [145]. Biometric signals can be classified into physiological and behavioural signals [49]. Examples of physiological signals include face recognition, fingerprint, iris, ECG, Electromyogram (EMG) or Galvanic Skin Response (GSR). Behavioural traits have also been proposed, such as the voice, signature, keystroke dynamics or lip movements, among others.

Biometrics have also been used to generate personal cryptographic keys [196] by using biological signals as a PRNG. Therefore, in order to check if a given sequence of numbers can be considered random, there are some well known tests like Shannon's entropy, Monte Carlo test or frequency test among others. However, instead of using a subset of tests, there are some public suites like ENT[2] test—a software

[2] ENT can be downloaded at http://www.fourmilab.ch/random/

published by the NIST STS, DieHard [3] tool or TestU01[4] software that are more likely when evaluating the randomness property [20]. It is important to remark that ENT test was initially though for general purposes whereas the other suites are focused on guaranteeing some security properties.

2.1.1 Overview of Our Results

In the last years, entropy analysis has been shown as an effective mechanism to assist doctors in medical problems [64]. For instance, the analysis of brain images can help to the detection of some brain diseases [190, 201]. Another good example is the detection of cardiac problems through the analysis of ECG records [94, 107, 166]. In addition, and outside of the medical context, recent works have demonstrated that ECG signals can be also used as a source of entropy for security purposes [7, 101, 160, 206]. In particular, this is done by calculating the IPI which is the time-interval between two consecutive R-peaks of the ECG. If an arbitrary peak R occurs at the time $t_R(i)$, then IPI can be computed as the time difference between $t_{R(i)}$ and $t_{R(i-1)}$: $IPI_{(i)} = t_{R(i)} - t_{R(i-1)}$, as can be seen in Figure 2.2. We refer the reader to Section 2.2.2 for more details about the components of an ECG signal, and to Section 2.4.2 for the IPI extraction algorithm. Nowadays, apart from the common medical electrodes that record the ECG signal, there exist a myriad of devices equipped with dedicated sensors to measure the heart signal. For instance, measuring the heart rate can determine the efficiency of a workout or even the calories that someone has burnt. In order to do so, the exercise machines used in gyms normally have some metallic areas located on the support bars which interpret small electrical signals passing through the skin. There are, however, some other wearable devices with PPG sensors which record the blood pressure to get heart beats, i.e., a device illuminates the skin with a light source like a LED to detect the changes in the light absorption. Nowadays PPG monitors are usually found in most of the wristbands and smartwatches. Some other mechanisms like chest bands are commonly used by athletes when they are training or even in competitions to check their heart rates.

Many authors have claimed that the LSBs of the IPI contain a high

[3] Diehard can be downloaded at http://stat.fsu.edu/pub/diehard/

[4] TestU01 can be downloaded at http://simul.iro.umontreal.ca/testu01/tu01.html

degree of entropy [7, 8, 34, 136, 147, 158, 160, 162, 169, 185, 200, 205]. In addition, most of these authors use some public databases to prove this entropy property and thus, with this method, the resulting bits can be considered as random numbers and can be part of key generation protocols in authentication procedures.

Recent IPI-based authentication, identification, and key generation protocols (e.g., [8, 158, 162, 185]) suffer from two main weaknesses. First, they only use measures of entropy to determine whether the generated cryptographic material (keys and other intermediate values such as nonces) are random or not. Second, the datasets used in these works are rather small and, therefore, possibly not significant enough. Additionally, such datasets contain ECG signals obtained both from healthy subjects and others that suffer some heart-related pathology, and it is unclear whether this feature has some influence on the overall quality (i.e., randomness) of the derived bits. Some of these observations have been already raised in [20], in which authors pointed out the need to perform a more sound assessment of the quality of the generated keys using larger datasets and additional randomness tests. Nevertheless, the code which authors run these experiments is not available.

In this Chapter, we overcome these weaknesses by performing an analysis of the randomness of 19 different public databases containing heart signals. Our contributions can be summarized as:

- We have downloaded 19 public databases with information about heart signals from different people. All datasets are taken from the Physionet network[5] [57], which contain heart signals from both healthy volunteers and people with cardiac conditions. We then extracted the last four bits of the IPI of each person per database, thus creating a bit stream whose quality can be tested. In doing so, we attempt to address the gap detected in [20].
- We analyze all files independently to check if the ECG can be considered to be a good random number generator. To do so, two random number suites (ENT—general purpose—and NIST STS—security) have been run over all previously generated files. To the best of our knowledge, this is the first work that discusses how the ECG signal should be used in cryptographic protocols as a source of random numbers. Our scripts are made public[6] to

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

[6] https://github.com/aylara/Random_ECG

facilitate the replication of our results by other researchers.

- Contrarily to prior proposals, we demonstrate that the ECG signal contains some degree of randomness but its use in cryptographic applications is questionable. Some databases obtained reasonable results on either ENT or NIST STS. However none of the tested databases obtained good results on both at the same time except the mitdb database.

The rest of the Chapter is organized as follows: Section 2.2 provides some background on biometric authentication according to ECG and the basic description of some random tests. Section 2.4 describes the evaluation of our implementations and a discussion of the results. This Chapter ends with some conclusions in Section 2.5.

2.2 Background

In this section we provide some background on related work: biometric authentication, IPI-based authentication and key derivation protocols, and randomness tests.

2.2.1 Biometric Authentication

Biometric protocols provide security services such as the authentication and identification of a given person among a large set of people. Figure 2.1 illustrates the standard pipeline of a biometric system, from the signal acquisition and preprocessing to the final decision-making process to identify/authenticate the subject. At the core of the system there is a pattern-matching process between a freshly acquired template built from the subject's signal and a previously stored template. The matching process is usually done by defining an acceptance threshold and calculating the Hamming distance between the both templates to decide whether the subject is or is not authenticated. The signal is usually acquired by sensors that can be located in, on, or around the human body. Examples of well-known biometric signals include the iris [193], the fingerprint and face [50, 110], the voice [82] and the ECG [167].

Biometric approaches have been combined with traditional cryptographic primitives in several ways, including the replacement of matching algorithms by secure versions [81, 179], using biometric templates in Secure Multiparty Computation (SMC), homomorphic encryption

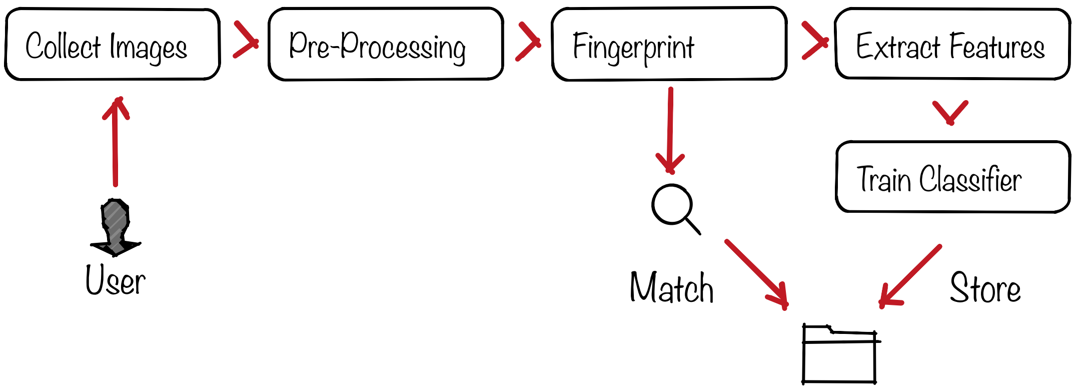


Figure 2.1: Architecture of a generic biometric recognition system.

schemes [39, 50, 180], or with elliptic curves [33, 59]. Apart from cryptographic proposals, the use of biometric signals to generate cryptographic keys has been widely studied in the literature (see, e.g., [8, 34, 129, 136, 147, 158, 162, 169, 185, 200, 205]). In most of these works, authors obtain a biological signal from different sensors or devices, such as the Electroencephalogram (EEG), the PPG, the ECG or accelerometers and check whether the signals can be considered random or not. To do so, the common practice is to extract some feature(s) from the signal and then run several randomness tests to validate the hypothesis.

Particularly, the use of IPIs has gained a special attraction in cryptographic application as a random number generator. For instance, in [73, 117, 186] to generate a private key, in [147] to be part of an authentication protocol, in [99, 195, 199] as an alternative to classical key establishment protocols or in [85] as part of a proximity detection protocol. It is worth noting the transcendence that IPIs have in all aforementioned scenarios and why it is crucial the random number generation.

2.2.2 IPI-Based Security Protocols

Figure 2.2 shows a typical ECG trace. The signal contains six different peaks, known by the letters P, Q, R, S, T and U. Heartbeats are commonly measured as the time distance between two consecutive R-peaks. This is known as *Inter-Pulse Interval (IPI)*, and several works published over the last decade have noted that the sequence

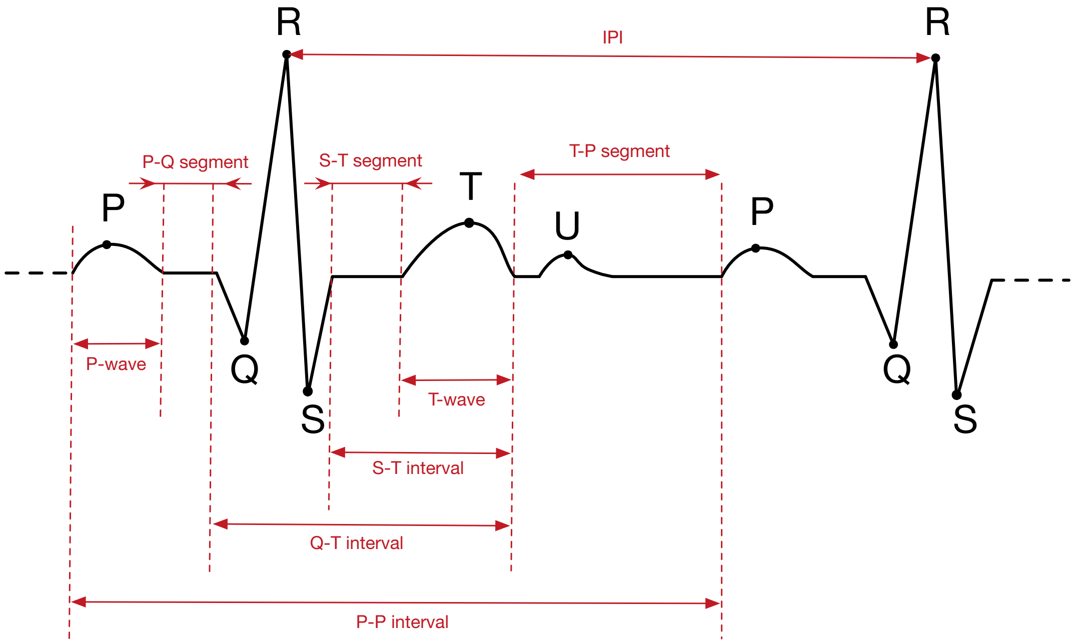


Figure 2.2: A typical electrocardiogram (ECG) signal and its main features: peaks (P, Q, R, S, T, U), waves, segments and intervals.

of IPIs contains some entropy. To obtain such random bits, each IPI should be first quantized, i.e., represented in binary code using some coding scheme. Most works omit the details about the particular coding scheme used, which is quite unfortunate since this is a critical component for the entropy (or lack thereof) of the resulting binary sequence. One notable example is the work of Rostami et al. [147], which will be described in more detail in Section 2.4.2 since it is the coding scheme used in this Chapter.

The majority of the proposed works in this area, e.g., [8, 17, 158, 162, 185, 205], conclude that the last 4-bits of each IPI can be used as a random number because of their high entropy. Thus, if an authentication protocol requires a 128 bit key to work, it would be necessary to acquire 32 IPIs (i.e., at least 33 consecutive R-peaks). Considering that a regular heart beats at 50-100 bpm, the key generation process would take between 20 and 40 seconds. To prove that the extracted bits have a certain level of randomness, most works use either the common Shannon or Rényi entropies [185]—which are not enough to claim the randomness property of a sequence of beats. Additionally, in [17, 35, 147, 195, 200, 205], authors remark the same claims about the randomness of the IPIs by running the NIST STS battery of randomness tests, whereas in [160] authors rely on the ENT suite. Table 2.1 summarizes the datasets that the existing works in this area have used. Additionally, in the last column, the number of executed tests can be seen where, for instance NIST STS(5/15) means that authors have run 5 tests out of 15 that the NIST STS suite has. Note that [35] is the only work where authors run all tests that NIST STS is composed of. We were not able to find the main reasons of running a subset of tests in the rest of the works that use NIST STS.

Work	Dataset	Randomness Test
[8]	50 subjects from the MIMIC II Waveform	Shannon's Entropy
[17]	99 subjects from a private dataset	NIST STS (5/15)
[35]	50 subjects from a private dataset	NIST STS (15/15)
[71]	Not specified	NIST STS (6/15)
[147]	47 subjects from mitdb; 290 from ptdb; 250 from mghdb	NIST STS (8/15)
[158]	mitdb (no info is given)	Shannon's Entropy
[162]	mitdb (no info is given)	Shannon's Entropy
[160]	mitdb (no info is given)	ENT
[185]	mitdb (no info is given)	Rényi's Entropy
[195]	PhysioNet[7]	NIST STS (9/15)
[200]	84 subjects from a private dataset and European ST-T	NIST STS (5/15)
[205]	18 subjects from MIT-BIH and 79 from the European ST-T	NIST STS (10/15)

Table 2.1: Datasets and number of run tests used by related work.

2.2.3 Randomness Tests

One key aspect of all IPI-based protocols is the assumption that some bits (four, typically) of each IPI are highly entropic. This condition is necessary, but not sufficient to guarantee the security of the protocol. In other words, high entropy does not necessarily imply randomness. Therefore, more sophisticated tests should be also applied to ensure that the values are indistinguishable from a random sequence.

In this Chapter, we have used ENT [189] and NIST STS [20] suites to evaluate how good the generated random numbers are. In particular, ENT is a suite composed of the following tests: entropy, Chi-Square, arithmetic mean, Monte Carlo, and serial correlation coefficient statistical tests. Finally, ENT reports the overall randomness results after running the aforementioned tests. On the contrary, NIST

[7] It is not specified in the paper

STS is a suite made of fifteen statistical tests: frequency monobit and block tests, runs, longest run of ones in a block, binary matrix rank, discrete Fourier Transform (spectral) test, overlapping and non-overlapping template matching, Maurer's Universal Statistical tests, linear complexity, serial, approximate entropy, cumulative sums, random excursions and random excursions variant tests. Finally, NIST STS reports a p-value which indicates whether the given sequence has passed or not each test.

For completeness, we refer the reader to the Section 2.3 where we provide a brief description of each one of the tests that form part of both NEST and NIST STS suites.

2.3 Random Tests

2.3.1 ENT

ENT [189] is a battery of tests used to evaluate pseudorandom number generators. The program reports the overall randomness results after running five different statistical tests: entropy, Chi-Square, arithmetic mean, Monte Carlo, and serial correlation coefficient. A brief description of each test is given next.

Entropy Test. This test measures the amount of information of the sequence, expressed as a number of bits per character. The higher the result, the more random the sequence is considered.

Chi-square Test. The Chi-Square test is one of the most commonly used tests and is extremely sensitive to errors in pseudorandom sequence generators. The test computes the chi-square distribution for the input stream of bits and provides the result both as an absolute number and a percentage indicating how frequently a truly random sequence would exceed the calculated value.

Arithmetic Mean. The value of this test indicates the result of adding up all bytes in the sequence and dividing it by the sequence length (in bytes). The closer the result is to 128, the more random the results.

Monte Carlo Value for π. This test estimates the value of π through a standard Monte Carlo method using the input sequence, which is considered random when the computed value is close to the true value of π. The test outputs both the estimated value of π and the error.

Serial Correlation Coefficient. This test attempts to capture correlations in the sequence by checking how much each byte in the stream depends upon the previous one. The closer the result is to 0, the more random the sequence.

2.3.2 NIST STS

The NIST STS test suite [20] is a set of fifteen statistical tests to evaluate random and pseudo-random number generators used in cryptographic applications. These tests are often used as a first step in spotting low-quality generators, but they are by no means a substitute for cryptanalysis. In other words, successfully passing all tests does not guarantee that the generator is strong enough.

All tests take as input a sequence of (binary) numbers and return a p-value that is then used to assess whether the sequence passed or not each test. In the following we briefly describe each test in turn.

Frequency (Monobit) Test. This is one of the simplest test which checks if the input sequence has a balanced number of ones and zeroes (i.e., if the distribution of bits is uniform).

Frequency Test within a Block. This test is an extension of the frequency monobit test, which can be considered as a particular case with the block size M equal to 1. For values $M > 1$, this test checks if the frequency of ones in an M-bit block is approximately $M/2$.

The Runs Test. This test measures whether the number of runs of ones and zeroes of various lengths are as would be expected for a truly random sequence [20]. A run is a consecutive sequence of bits with the same value. The test returns a p-value per block length.

Longest-Run-of-Ones in a Block. This test checks the length of the longest run of ones in a previously defined block with length M and compares it with the expected value for a truly random sequence.

The Binary Matrix Rank Test. This test generates $m \times n$ binary matrices over $GF(2)$ using the values of the input sequence (each row of a matrix is a substring of the sequence) and checks whether the ranks are linearly dependent.

Discrete Fourier Transform (Spectral) Test. This test calculates the Discrete Fourier Transform of each subsequence of bits and computes the peaks, which might reveal patterns in the original sequence. The test uses a threshold $t = \sqrt{log(1/0.05)n}$, where n is the length of the sequence. If the number of peaks is at most 5%, the sequence can be considered as random.

Non-overlapping Template Matching Test. In this test, the random sequence is split into M substrings of length l. The test seeks for the number of occurrences of a given template. If the pattern is found, the algorithm resets the substring M to the bit after the found pattern, otherwise M is reset to the next bit.

The Overlapping Template Matching Test. This test is identical to the non-overlapping template matching test but using overlapping substrings (i.e., using a sliding window that advances 1 bit at a time).

Maurer's "Universal Statistical" Test. The purpose of this test is to detect if the sequence can be significantly compressed without loss of information. One of the main drawbacks of this test is that it requires a substantially long sequence for the result to be relevant.

The Linear Complexity Test. This test computes the linear complexity of the input sequence. If the value is too short, the sequence is not considered random enough.

The Serial Test. The focus of this test is to calculate the frequency of all possible overlapping M-bit patterns in the whole sequence. That is, each M-block should have the same probability of appearing than any other M-bit pattern.

The Approximate Entropy Test. This test is focused on the frequency of all possible overlapping m-bit pattern in a sequence. In short, this test compare the frequency of two adjacent lengths (m and $m + 1$) to the expected result for a random sequence.

The Cumulative Sums Test. In this test, zeroes are converted to negative ones and ones remain the same. This test is based on the maximum distance from zero of a random walk defined by the cumulative sum of the sequence. For a random sequence, the cumulative sum should be close to zero.

The Random Excursions Test. This test calculates the number of cycles having exactly K visits in a cumulative sum random walk, which is derived from partial sums.

The Random Excursions Variant Test. This test is similar to the random excursion test but, in this case, the goal is to detect deviations from the expected number of visits to various states in the random walk.

2.4 The Randomness of IPI Sequences

This section describes our experiments to analyze the randomness of the IPI values and a discussion of the obtained results.

2.4.1 Dataset

For consistency with previous research in this area, we have first downloaded the mitdb, ptbdb and mghdb Physionet[8] databases from [57] and we have tried to replicate the experimental setting used by both Rostami et al. in [147] and Xu et al. in [195]. The results, however, were impossible to reproduce due to the lack of information that authors provide in the original papers. The downloaded databases contain the information of several subjects and we do not know how the original experiments were run, e.g., i) by acquiring the 4 LSBs of the ECG of each one of the subjects and after that running (a subset) of the NIST STS tests per person; ii) if the authors generated one single file with the information of all subjects belonging to the same database and then this file was used as input of some of the NIST STS tests; or, iii) if the authors generated one single file with the information of all subjects of all databases and then they run (a subset) of the NIST STS tests.

Due to the fact that only one single value was given in [147] regarding the final results of the NIST STS and also that at certain moment, authors claim that they use an aggregate of different databases for the error generation, we assume that authors used the iii) approach: they created one single file with the 4 LSBs of the IPI of different people belonging to different datasets. Nevertheless, we consider that this is not a realistic experiment because of the heterogeneous of the databases (see Table 2.2) as it was also pointed out by [20]. On the contrary, authors in [195] neither provide the achieved results of the NIST STS nor they say which database(s) they use for testing.

For these reasons, we have substantially extended this setting to 16 additional datasets of ECGs also present in the Physionet repository.

[8] The software package to access the data repository can be found at `https://physionet.org/physiotools/wfdb.shtml`

All these datasets contain ECG records obtained from a variety of real subjects with different heart-related pathologies in many cases. Table 2.2 shows the main features of the 19 datasets used in this work. Also, we have computed the median value of the extracted IPIs per file (person) per database. For instance, it is easy to argue that heart signals acquired from people equipped with holters (cdb) cannot be used to prove that the heart signal is random enough. Similar cases occur with iafdb, ptdb or twadb databases with medians of 37, 68 and 87 IPIs respectively.

In order to avoid the aforementioned problems and to allow other researchers to reproduce the results, we have split up the results in their corresponding databases. After that, we have extracted the 4 LSBs of each subject and run the random tests (NIST STS and ENT suites) to each individual file (corresponding to each subject of each database) to evaluate how good the generated random numbers are. Finally, the results are grouped per pathology (database) and we give a percentage of the files (persons) which successfully passed the random tests.

2.4.2 IPI Extraction

Previous works in this area found out that the four LSBs of each IPI are highly entropic [147, 195]. We replicated this process as follows. We first used a Matlab script available at the Physionet repository[9] to obtain the ECG signal for each record (person) in each one of the 19 datasets. We next applied the following steps:

1. Get the sampling frequency for each signal, which is available in an associated description record.
2. Run Pan-Tomkins's QRS detection algorithm [132] over the ECG signal to extract the R-peaks.
3. Get the timestamp of each R-peak and calculate the difference between each pair of consecutive R-peaks to obtain the sequence of raw IPI values.
4. Apply a dynamic quantization algorithm to each IPI to decrease the measurement errors. This process consists in generating discrete values from an ECG (continuous signal).
5. Apply a Grey code to the resulting quantized IPI values to increase the error margin of the physiological parameters.

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

Dataset	#Records	Frequency (Hz)	Median (IPIs)	Pathology
aami-ec13 [76]	10	720	48.5	Tachycardia
apnea-ecg [135]	77	100	15786	Tachycardia
cdb [121]	53	250	12	Holter recordings
cebsdb [54]	54	5,000	175	Healthy
cudb [128]	9	250	415	Ventricular problems
edb [170]	90	250	4405	Myocardial and hypertension
iafdb [139]	5	1,000	37	Atrial fibrillation or flutter
mitdb [120]	46	360	1113	Arrhythmia
mghdb [192]	202	360	2426	Unstable patients in ICU
nstdb [122]	14	360	1246	Mitdb with noise
ptbdb [24]	545	1,000	68	Myocardial and Healthy controls
qtdb [96]	104	250	520.5	Holter recordings
shareedb [116]	23	128	46910	Hypertension
slpdb [74]	17	250	11517	Sleep Apnoea syndrome
stdb [6]	28	360	1243	Stress tests
svdb [63]	47	128	1192	Partial epilepsy
szdb [5]	7	200	4439	Partial epilepsy
twadb [123]	5	500	87	Myocardial problems
vfdb [62]	17	250	1800	Tachycardia

Table 2.2: The 19 datasets used in this work. For each dataset the table provides the number of records (subjects), the sampling frequency, the median value of IPIs per database and the pathology (if any) of the subjects involved in each dataset.

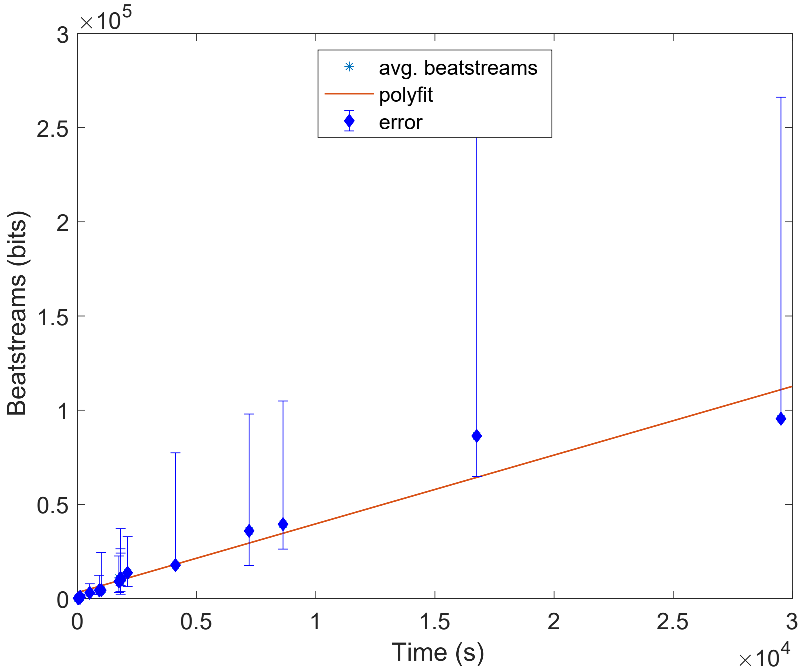


Figure 2.3: Statistical analysis of beatstreams (in bits) and time (in seconds).

6. Extract the 4 LSB from each coded IPI value.

Each sequence of extracted bits per record of each dataset is stored in separate files for subsequent analysis.

Additionally, we have also conducted one more experiment in Matlab under a MacPro laptop with 4Gb of RAM to estimate how long the signal should be to extract a stream of length x bits. To do so, we have computed the average number of IPIs and the length average of the signal of the nineteen databases. In Figure 2.3 these results can be seen where we can conclude that the relation between time and the length of bits is linear and for instance, after almost 4 hours we will have approximately 60,000 bits which can be used as random numbers. It is also noticeable that these results are consistent with the hypothesis that in order to extract a valid cryptographic key, only a few seconds are enough. In other words, to generate a cryptographic key of 128 bits, a device should wait between 20 and 50 seconds to create that key. It is also remarkable that depending on the scenario this time constraint might be not feasible to be deployed—e.g., a person who is suffering from a heart attack cannot wait for a minute to authenticate its pacemaker with the caregiver device.

Test	**Optimal value**	**Threshold**	**Counter**
Entropy	1.0	>0.85	0.99
Optimum compression	<0%	<5%	0%
Chi square	5% < $\tilde{\chi}^2$ <95%	5% < $\tilde{\chi}^2$ <95%	1%
Arithmetic mean	0.5	0.4< $\bar{x}$ <0.6	0.46
Monte Carlo value for π	error=0%	error <5%	12.38%
Serial Correlation coefficient	0	$< 10^{-1}$ or $< 10^{-2}$	0.012

Table 2.3: ENT tests: optimal values, thresholds used to consider that a sequence passes the test, and results obtained for a counting sequence.

2.4.3 Measuring Randomness

In this section we discuss the results of applying both the NIST STS and ENT test suites to the datasets discussed above.

2.4.3.1 ENT

As described in Section 2.3.1, the ENT suite comprises 6 tests of randomness. Table 2.3 shows the optimum value for each one of them. Along with this, we also provide two additional values for each test: i) a threshold, which constitutes a more affordable value for each test since the optimal output is quite restrictive and most sequences would fail the tests otherwise, and; ii) the test result obtained for an input sequence consisting of a simple counter value from 0 to 2^{14}. The purpose of this experiment is just to demonstrate that the result of a single test cannot be used alone to claim evidence of randomness; see, e.g., the output achieved by the counting sequence for the entropy, the arithmetic mean, the serial correlation coefficient and the optimum compression.

The results obtained after applying the 6 ENT tests to each one of the files (persons)—with the IPIs of their ECG signals in our 19 datasets—can be seen in Table 2.4. Each cell in the table provides the percentage of persons who pass the test using the threshold shown in Table 2.3. For instance, in the case of mitdb database, we have generated 46 files, belonging to 46 persons involved in this database, with a median of 1113 IPIs per file. The results for this database are that all persons pass both the entropy and optimum compression tests (100%) but none of them pass the chi square test (0%); 45 out 46 pass the

arithmetic mean and the serial correlation tests (97.83%); and, 22 out of 46 pass the Monte Carlo value for π (47.83%).
Overall, the first noticeable observation is that these results are quite good across all datasets in the entropy, optimum compression, and serial correlation whereas for the chi square the results are catastrophic. The situation is similar for the Monte Carlo for π test where all databases fail but szdb, slpdb, edb and shareedb achieving 71.43%, 74.47%, 60% and 55.52%, respectively. On the contrary, the arithmetic mean test achieve good results but the vfdb, and the cudb fail that test with 17% and 44.44%, respectively.
Looking at the results from a dataset perspective, we were not able to identify if there exists some correlation among the tests results with the information available to us (number of samples, sampling frequency, signal length, IPIs per file or characteristics of the subjects). See also the discussion provided later on in Section 2.4.4 for an additional analysis on this.

2.4.3.2 NIST STS

In Section 2.3.2 a description of all the fifteen tests that comprises this suite can be read. As a common feature, all NIST STS tests are parameterized by a variable n which means the length of bits of the processed bitstream. Additionally, some of the tests can also detect local non-randomness: frequency test within block, overlapping and non-overlapping template matching, Maurer's "Universal Statistical", linear complexity, serial and approximate entropy tests. These tests are also parameterized by a second variable denoted as m or M. Those tests which use m parameter are mainly focused on detection of m-bit patterns in the stream whereas those tests which use M parameter, check the distribution of the specific feature across n/M blocks of equal size (M bits). In Table 2.5 the minimum requirements in terms of length can be seen.
Furthermore, if we take into account the values of Table 2.2 regarding the length (median) of our datasets, we cannot run the original NIST STS with enough confidence level. The Physionet datasets are irregular in their size, with several of them having too small size to be used with the original tests. In order to circumvent the length constraints that the original NIST STS has, we have used a variant [56] of the original software package.
Table 2.6 provides the success rate obtained for the 15 NIST STS

Dataset	**Entropy**	**Optimum Compression**	**Chi square**	**Arithmetic Mean**	**Monte Carlo value for π**	**Serial Correlation**
cebsdb	100%	100%	0%	50%	10%	60%
ptbdb	99.82%	100%	0%	97.98%	22.20%	99.63%
twadb	100%	100%	0%	80%	0%	100%
iafdb	100%	100%	0%	100%	40%	100%
cdb	100%	100%	0%	81.13%	1.89%	96.23%
nstdb	100%	100%	0%	92.86%	35.71%	100%
mitdb	100%	100%	0%	97.83%	47.83%	97.83%
qtdb	99.04%	100%	0%	96.15%	38.46%	100%
stdb	100%	100%	0%	100%	35.71%	100%
cudb	100%	100%	0%	44.44%	11.11%	100%
aami-ec13	80%	100%	0%	50%	10%	60%
svdb	100%	100%	0%	97.87%	42.55%	97.87%
vfdb	83%	100%	0%	17%	6%	94%
szdb	85.71%	100%	0%	85.71%	71.43%	85.71%
slpdb	100%	100%	0%	100%	74.47%	100%
edb	98.89%	100%	0%	98.89%	60%	100%
mghdb	72.28%	100%	0%	59.41%	22.28%	86.14%
apnea-ecg	75.32%	100%	0%	62.34%	29.87%	81.82%
shareedb	95.65%	100%	0%	95.65%	55.52%	100%

Table 2.4: Results of the ENT tests expressed as the percentage of subjects that pass each test per database.

Test Name	n	m or M
Frequency (Monobit)	$n \geq 100$	-
Frequency Test within a Block	-	$20 \leq M \leq n/100$
Run	$n \geq 100$	-
Longest Run of Ones in a Block		
Binary Matrix Rank	$n \geq 38912$	-
Discrete Fourier Transform (Spectral)	$n \geq 1000$	-
Non-Overlapping Template Matching		$2 \leq m \leq 21$
Overlapping Template Matching		$1 \leq m \leq n$
Maurer's "Universal Statistical"		$1 \leq m \leq n$
Linear Complexity	$n \geq 10^6$	$500 \leq M \leq 5000$
Serial		$3 \leq m \leq \lfloor log_2 n \rfloor - 3$
Approximate Entropy		$m \leq \lfloor log_2 n \rfloor - 6$
Cumulative Sums	$n \geq 100$	
Random Excursions	$n \geq 10^6$	
Random Excursions Variant	$n \geq 10^6$	

Table 2.5: NIST STS requirements in terms of length [168].

tests to the files (subjects) of each dataset. In this case, we used the pass criteria included in each test, which are based on an analysis of the yielded p-values. In other words, p-values of less than 0.01 are considered to reject. Overall, the results are similar to those obtained for ENT, although in this case the success rate is generally higher in most cases. Also, there are substantial differences across datasets. For instance, iafdb, ptbdb, and twadb obtain success rates higher than 80% in 12, 12, and 11 out of the 15 tests, respectively. Contrarily, the performance of many datasets is considerably poor, with less than 50% of their records not passing a majority of the tests: see, for example, the cases of apnea-ecg and cudb (more than 50% of the records fail 9 out of 15 tests); svdb (more than 50% of the records fail 19 out of 15 tests); edb, slpdb, szdb and vfdb (more than 50% of the records fail 11 out of 15 tests); mghdb (more than 50% of the records fail 12 out of 15 tests); and lspdb (more than 50% of the records fail 13 out of 15 tests). In the case of slpdb and szdb the results are very deficient, with all signals in both datasets failing 9 out of the 15 tests (i.e., 0% of success rate) .

In terms of performance against individual tests, the results are rather diverse, with a few exceptions. The case of the linear complexity test stands out, as most datasets exhibit an extremely poor result. This suggests the existence of patterns than can be modelled by linear prediction functions, which undoubtedly implies predictability. Similarly, most datasets perform badly in the monobit and block frequency tests, which reveals a non-negligible imbalance of zeroes and ones (monobit frequency) and, more generally, all possible M-block bit patterns (block frequency).

Finally, it is worth noting that there *seems* to be some correlation among the tests results, particularly for datasets that do and do not perform well. Consider, for example, the case of cdb (11/15), iafdb (13/15), ptdb (13/15), twadb (13/15), and cebsdb (14/15) which obtain extremely good results (at least pass 11 out of 15 tests in the worst case, i.e., cdb) for all tests. All these databases have in common that the number of IPIs in median is less than 175—note that an IPI is made of 4 bits. Contrarily, shareedb (2/15), apnea-ecg (3/15), mghdb (3/15), edb (4/15), slpdb (4/15), szdb (4/15), and vfdb (4/15) achieve very poor results (only pass 4 out of 15 in the best case) in the NIST STS having 46910, 15786, 2426, 4405, 11517, 4439 and 1800 IPIs in median respectively.

2.4.4 Discussion

In Table 2.7 a summary of all tested databases can be seen with the typology of each dataset in order to find out some relations between them. Notice that if we analyze the results in average, all databases achieve reasonable results in the ENT suite whereas 8 out of 19 pass the NIST STS tests. Nevertheless, this is not true at all as we can see in Table 2.4 that none of the tests pass the Chi Square test which is crucial due to this test checks if the sequence is random or not [189]. Moreover, the Monte Carlo test achieves 36.45%, i.e., only edb, shareedb, slpdb and szdb databases pass the Monte Carlo test.

It has been previously pointed out (see Table 2.2) that many authors only use the mitdb dataset which is true that passes most of the tests of both suites but not all of them. Thus, it is not a real assumption to claim that ECG can be considered to be random only by taking the entropy results. We have proven that a counter achieve similar results and it is well known that it cannot be used as a random generator (Table 2.3).

On the one hand, we have run all ENT tests—6 out of 6—to all databases with different samples per signal of each one of the sub-

Dataset	Monobit frequency	Block frequency	Runs	Longest run ones	Binary matrix rank	Spectral	Non overlapping template matching	Overlapping template matching	Universal statistic	Linear complexity	Serial	Approximate entropy	Cumulative sums	Random excursions	Random excursions variant
cebsdb	94%	81%	85%	87%	98%	100%	96%	72%	100%	20%	85%	96%	96%	98%	100%
ptbdb	89%	89%	89%	89%	85%	92%	98%	84%	1%	0%	100%	85%	98%	99%	65%
twadb	80%	60%	80%	100%	80%	80%	100%	80%	20%	0%	100%	60%	100%	80%	80%
iafdb	100%	100%	100%	80%	100%	100%	100%	100%	20%	0%	100%	100%	80%	100%	40%
cdb	94%	25%	92%	94%	100%	94%	96%	81%	0%	0%	100%	77%	100%	100%	0%
nstdb	14%	21%	7%	64%	21%	50%	71%	57%	86%	57%	7%	93%	93%	71%	100%
mitdb	46%	33%	33%	50%	35%	59%	91%	52%	89%	39%	9%	87%	96%	85%	100%
qtdb	47%	41%	44%	54%	25%	53%	92%	56%	89%	0%	0%	77%	98%	81%	95%
stdb	50%	7%	29%	54%	21%	21%	64%	32%	64%	21%	4%	71%	100%	32%	100%
cudb	0%	11%	0%	56%	11%	22%	11%	67%	67%	0%	11%	56%	100%	22%	78%
aami-ec13	10%	20%	30%	40%	20%	10%	90%	90%	0%	0%	100%	60%	100%	0%	40%
svdb	23%	11%	19%	28%	6%	9%	77%	43%	77%	21%	0%	77%	100%	28%	94%
vfdb	29%	12%	12%	41%	18%	12%	29%	29%	88%	18%	0%	71%	100%	24%	100%
szdb	14%	0%	0%	29%	0%	0%	43%	29%	71%	0%	0%	86%	100%	0%	86%
slpdb	24%	0%	6%	35%	6%	12%	76%	12%	24%	0%	0%	76%	94%	6%	94%
edb	23%	1%	14%	29%	3%	7%	62%	21%	43%	9%	0%	86%	100%	2%	94%
mghdb	22%	14%	14%	33%	11%	12%	35%	34%	34%	7%	19%	57%	99%	15%	60%
apnea-ecg	5%	4%	4%	17%	1%	0%	27%	26%	23%	0%	9%	68%	96%	1%	75%
shareedb	9%	0%	0%	4%	0%	0%	17%	0%	0%	0%	0%	48%	100%	0%	96%
Average	36.8%	21.0%	26.3%	52.6%	26.3%	42.1%	68.4%	52.6%	47.3%	5.2%	31.5%	94.7%	100%	42.1%	84.2%

Table 2.6: Results of the NIST STS tests expressed as the percentage of subjects that pass each test.

jects, however here we have only focused on mitdb database because it has been commonly used in the literature. Figure 2.4(a) shows that when the length of the IPIs (number of bits in the file) is less than 6,000 bits, the probability of being success is less than 0,5 in average whereas when the length is greater than 6,000 the probability is between 0.1 and 0.8. Those results corroborate the same results previously got in Table 2.4 where Chi-Square test achieves a 0% of success whereas the optimum compression test nearly has a 100% of success.

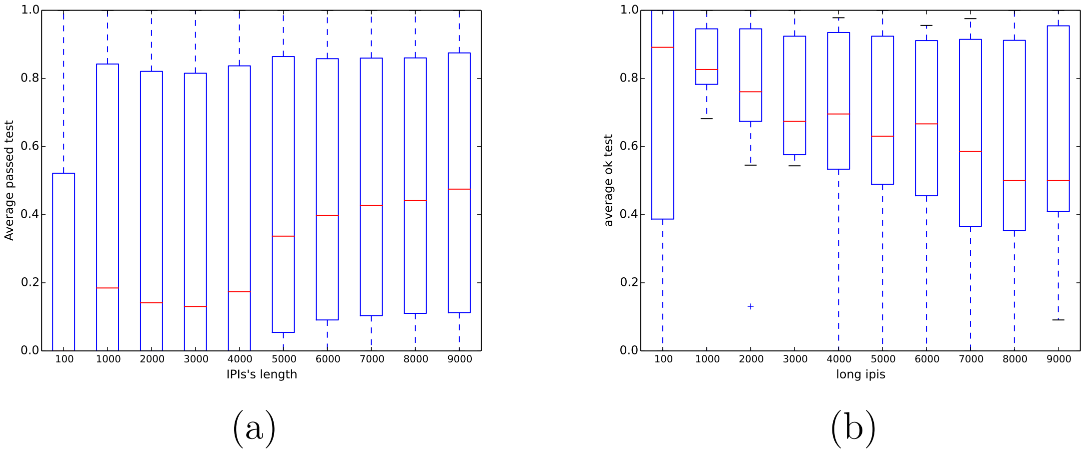


Figure 2.4: distribution of the fraction of tests passed for the mitdb dataset as a function of the number of bits used. (a) ENT suite, and (b) NIST STS suite.

On the other hand, we have run all NIST STS tests—15 out of 15—to all databases with different IPIs in median. Nevertheless, similarly to the ENT experiment, we have only focused on mitdb test instead of the rest of databases. Contrarily to the results obtained in option a) in [147], Figure 2.4b shows that when the length of the IPI (number of bits in the file) increases the results are worse and even when the length is higher than 7,000 bits, the probability of being successful is close to 0.5.

After analyzing carefully Table 2.7 where the average of the results can be seen, we extract the following information:

- When the number of IPIs in median is higher than 1800, then databases achieve extremely poor results (2 passed tests out of 15 in the worst case) in the NIST STS. Examples of these databases are vfdb, szdb, slpdb, mghdb, edb, apnea-ecg and shareedb.
- When the number of IPIs in median is between 1800 and 415, then databases are in the borderline of passing (at least) half

Dataset	ENT	NIST STS	Avg. No. Samples	Median (IPI)	Pathology
cebsdb	66.6%	93.3%	4,968,780	175	Healthy volunteers
ptbdb	66.6%	86.6%	108,818	68	Myocardial problems and Healthy controls
twadb	66.6%	86.6%	59,770	87	Myocardial problems
iafdb	66.6%	80.0%	19,707,034	37	Atrial fibrillation or flutter
cdb	66.6%	73.3%	5,120	12	Holter recordings
nstdb	66.6%	66.6%	650,000	1246	Physically active volunteers
mitdb	66.6%	60.0%	650,000	1113	Arrhythmia
qtdb	66.6%	60.0%	224,999	520.5	Holter recordings
stdb	66.6%	46.6%	624,166	1243	Stress tests
cudb	50.0%	40.0%	127,232	415	Ventricular problems
aami-ec13	66.6%	33.3%	55,522	48.5	Tachycardia
svdb	66.6%	33.3%	230,400	1192	Partial epilepsy
vfdb	50.0%	26.6%	525,000	1800	Tachycardia
szdb	83.3%	26.6%	17,245,701	4439	Partial epilepsy
slpdb	83.3%	26.6%	4,188,530	11517	Sleep Apnoea syndrome
edb	83.3%	26.6%	1,800,000	4405	Myocardial and hypertension
mghdb	66.6%	20.0%	1,479,358	2426	Critical Care Units
apnea-ecg	66.6%	20.0%	11,930	15786	Tachycardia
shareedb	83.3%	13.3%	10,553,116	46910	Hypertension

Table 2.7: Characteristics vs. success rate datasets.

of the NIST STS. Examples of these databases are svdb, cudb, stdb, qtdb, mitdb and nstdb. There is also one exception to this rule: aami-ec13 which has 48.5 IPIs in median and it achieves a 33.3% (5 passed tests out of 15) which is similar to svdb results.

- When the number of IPIs in median is between 415 and 37, the databases achieve extremely good results (14 passed tests out of 15 in the best case) in the NIST STS. Examples of these databases are cdb, twadb, pbdb, iafdb, cebsdb. As before, there is an exception to this rule: aami-ec13 which has 48.5 IPIs in median and it only passes 5 out of 15 tests.

We have tested 19 public databases from the Physionet repository. This has turned recently a common practice in security proposals and mitdb has been used as a starting point for authentication and security based protocols. According to the results presented in this work, we can claim that mitdb is not the best database for this purpose but cebsdb. However, other tests such as Diehard was impossible to be run with these databases because of the length of the signals—Diehard needs binary files that usually go from 10 to 12 million bytes.

2.5 Conclusions

In this Chapter, we have addressed the random number generation issue by using heart signals—in particular, ECG records are used. Some authors have claimed that the 4 LSBs of the IPI values have certain entropy level. Despite we have proven they have some entropy degree, we have also showed that ECG records, and consequently IPI values derived from them, should not be considered a good source of randomness only by observing that value. We have used both ENT and NIST STS test suites to evaluate the randomness property of 19 public and well-known ECG databases and results point to the fact that IPIs values are not as random as supposed. The database that achieves better results is cebsdb (healthy volunteers records) instead of mitdb (arrhythmia record) which is the most common database used in the literature. The use of cebsdb database seems more appropriate since users do not suffer any medical condition and no defect (or bias) is a priory expected in the signals—in addition, the size of the database is more appropriate.

The results obtained through the conducted in-depth analysis clearly point two conclusions out: 1) a short burst of bits derived from an ECG record may seem random, but; 2) large files derived from long ECG records should not be used for security purposes (e.g., key generation algorithms). These conclusions should be taken with caution since these are conditioned to: 1) IPI extraction algorithm described in Section 2.4.2, and; 2) the 19 public databases studied. Finally, we highlight here that all the necessary scripts to reproduce our experiments are public available on (`https://github.com/aylara/Random_ECG`).

3

Feasibility Analysis of Inter-Pulse Intervals Based Solutions for Cryptographic Token Generation by Two Electrocardiogram Sensors

3.1 Introduction

Interest in biometrics has gained momentum in the last years mostly due to the massive use of daily life devices like smartwatches, smartphones and laptops [66, 95]. This technology identifies and authenticates people in an automatic way based on biological and behavioral traits [191]. This interest is not temporary. According to a recently published report, global biometric market revenues will reach $34.6 billion annually in 2020, especially in mobile devices [78].

From a technical point of view, biometrics can be classified into two main groups depending on whether they use physiological or behavioral signals. Examples of physiological signals include fingerprints, iris, retina, heart and brain signals, whereas voice, signature analysis or keystroke dynamics are behavioral signals. The main reason why such signals can be easily included in authentication systems is be-

This chapter is based on this [131] publication cause they exhibit a number of desirable features: they are universal, collectible, unobtrusive, permanent, unique, and difficult to circumvent [49].

The research outcome in this area is that most gadgets, such as smartphones, tablets, wearables and IMDs, have been equipped with one or more embedded sensors with the ability to measure biometric parameters from the bearer. Besides having biometrics sensors, most (if not all) of these devices are enhanced with some wireless communication technology, e.g., Bluetooth, WiFi or Radio Frequency (RF), allowing them to share data and to perform remote reconfiguration [44]. All the above has given birth to the so-called WBAN.

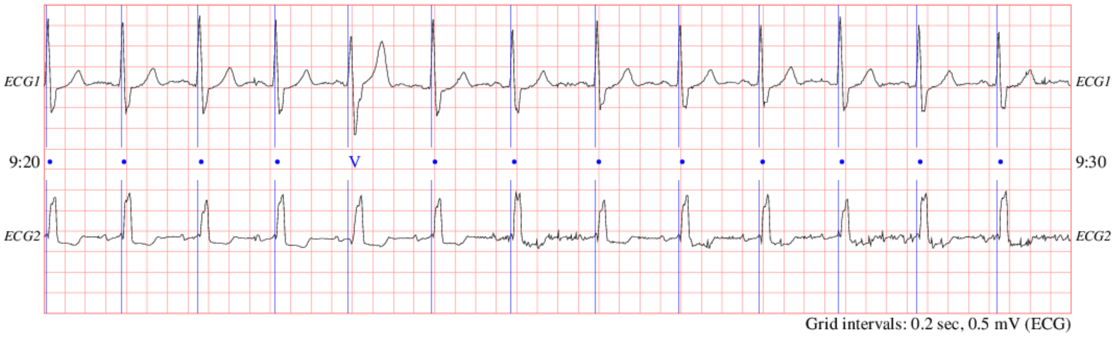


Figure 3.1: Two ECG signals from svdb [63] database.

In the last years, several works have focused on using the heart signal as part of either authentication protocols [144, 147, 162], human identification [17, 29], or as a key generation algorithm [58, 160, 184, 200] to enable secure communications. More concretely, authors use the ECG to extract the time difference between two consecutive heartbeats (R-peaks). These time intervals are referred to as IPIs or RR-intervals and have been shown to contain some degree of entropy after applying a quantization algorithm (see Section 3.3.2.1). This makes the IPI values an ideal candidate to generate tokens to be used in cryptographic solutions (e.g., [8, 147, 162, 185, 205]).

In order to obtain a biometric signature based on the heart signal, different sensors such as ECG, PPG or BP can be used. The ECG signal is measured using electrodes usually placed on the chest which detect the tiny electrical changes in the heart and generate a complex digital signal. The PPG detects the pulse of the heart by measuring the amount of light which is reflected in the skin to a photodiode. As a light source, most of the commercial gadgets have a LED on them, e.g., smartwatches and sport wrists. As an example on how these advances may be used for new purposes, some researchers have recently used a BP sensor to get the bearer's heart signal [162]: this sensor can measure the pressure in large arteries in the systemic circulation, so the signal reflects the up and down fluctuation of the arterial pressure which is related to each heartbeat.

Using these sensors is not trivial though, as there are some technical difficulties due to different factors. For example, even when two similar sensors—from the same manufacturer, having the same brand, and with the same capabilities—are measuring the same heart signal in the same part of the body, the resulting signal would likely be different in both sensors due to the noise of the signals, missed data during the gathering phase, delays, or simply because of the bearer's movements [158].

Along the same lines, it has been reported in [104] that both HRV and Inter-Sensor Variability (VAR_{is}) measurements directly affect the processing of the heart signal and, in particular, the peak detection procedure. These issues become crucial when a cryptographic protocol entirely relies on biometric data acquisition to generate random tokens, e.g., random seeds or fresh nonces, to be used for key generation [196] or in authentication procedures [147].

In particular, the problem of signal synchronization is quite relevant in the health sector where expensive medical electrodes are used. Let us consider a real example of measuring ECGs using two different sensors. Figure 3.1 shows two ECG signals, channel 1 (ECG_1) and channel 2 (ECG_2), taken from the public database svdb [63]. This database is composed of 78 half-hour ECG recordings of supraventricular arrhythmias. The Bits per Minute (bpm) in both signals are the same, or, in the worst case, show a difference of a few bpm. However, at low level, the time differences between two consecutive heartbeats (R-peaks), are slightly different in ECG_1 and ECG_2. Thus, despite sensing the same ECG from the same patient, both channels have different signals, and it is easy to see that even by shifting any of the signals it could not be possible to fully synchronize them.

Authors are somehow aware of this problem and for instance in [158], a miss-detection algorithm is proposed that given two ECGs, authors "manually" add a peak in the place where it was supposed to be whenever it is detected that a peak was missing in order to generate the same token in different devices. Some years later, in [163], authors propose a key-exchange protocol among a Programmer[1] and an IMD where both devices generate the same key from the heart signal. After gathering the same signal, authors apply a BCH, which is an ECC, to the generated keys in both devices to finally get the same value.

3.1.1 Our Work

No matter if authentication protocols for WBAN were published [17, 18, 147], if key distribution schemes based on the heart signals were proposed [144, 163] or whether authors assumed that there is a secure communication channel and a shared key is derived from the heart signal to be used afterwards in a cryptographic protocol [36, 158, 162, 200], all these proposals rely on the same assumption: there are two sensors measuring the heart signal and they can derive the same cryptographic token under an IPI-based approach and after applying an ECC algorithm like BCH. Unfortunately, after an in depth analysis (19 databases), we show that the above claim does not hold when only an ECC algorithm is used to correct errors between the two generated tokens.

Motivated by this, we carry out an analysis on the (open) question concerning the generation of a cryptographic token based on the analysis of IPI values from different ECG devices that are sensing the same heart signal. Our analysis is based on the use of a run-time monitor, extracted from a formal model, i.e., a timed automaton, that is verified against predefined properties, combined with a fuzzy extractor (i.e., an ECC) to improve the final result. We show that it is impossible, in general, to correct the differences between the two captured signals when using only the fuzzy extractor, thus being impossible to corroborate previous claims on the feasibility of the approach.

Our proposed method can successfully synchronize two heart signals through IPI values and extract a common token that can be used afterwords as part of a cryptographic protocol, as one more security check in order to proof that both devices are attached to the same body by proving that they are listening to the same heart signal, i.e., they are attached to the same body.

To the best of our knowledge, this is the first work to use a run-time monitor in combination with a fuzzy extractor. In addition, to demonstrate the validity of our approach, we provide a large-scale

[1] Device used to (re)configure IMDs.

evaluation of the proposed method over 19 public databases containing heart signals. However, we do not evaluate how good or bad the IPI-based generated random tokens are from a cryptographic point of view; we urge the reader to consult [130] for an in-depth analysis of this issue.

After applying our proposed solution to public databases containing at least two measurements of heart signals (ECG_1 and ECG_2), we conclude that a fuzzy extractor (or another error correction technique) is not enough to correct the synchronization errors between the IPI values derived from two ECG signals captured via two sensors placed on different positions (Section 3.3). In particular, we show that a pre-processing of the heart signal must be performed before the fuzzy extractor is applied.

3.1.2 Contributions

In summary, our contributions are:

- We perform an in-depth analysis of the problem of how to synchronize two cryptographic tokens generated by two different ECG sensors that record the same heart signal and use the IPIs as the basics for generating the mentioned tokens. We show how an initial signal pre-processing step is necessary for the error correction algorithm (e.g., fuzzy extractor) to work properly. Our results show that it is not possible to assume that two sensors can derive a common token just by applying an error correction algorithm without having previously synchronized both signals. In summary, this first result gives evidence that the assumptions under which previous IPI-based solutions operate are not correct and does not guarantee that the same token can be extracted from two ECGs sensors (Section 3.3.2).
- In order to perform the synchronization (at IPI values level) between two ECGs sensors, we have generated a run-time monitor from a timed automaton, which has been verified correct with respect to predefined timing properties. We compare our results before and after applying a fuzzy extractor and demonstrate our improvement in performance (Section 3.3.3).
- We modified our timed automaton and the monitor in order to extract a token with a given accuracy (namely 32, 64 and 128 bits), in order to gather statistical information on how long it

would take (median) to get a token with the requested accuracy. We found that to generate a 32, 64 and 128 bits tokens, a sensor should wait on median 13, 28 and 56.5 seconds, respectively (for individual with a heart rate of 80 beats-per-minute), instead of 6, 12, and 24 seconds as reported in previous works, i.e., [17, 124, 147, 195] (Section 3.3.3.2).

- We have developed a proof-of-concept implementation of an ECG-based token generator by using a BITalino shield[2] (Section 3.4). This shield has two ECG channels connected using wires and the pre-processing is executed before the token generation (IPI-based approach in our particular case) takes place. The purpose of this proof-of-concept is to shed further light on the technical real difficulties in getting a fully working implementation of such a solution.
- As it was previously stated, the contributions in this Chapter shed light on the feasibility of IPI-based solutions, where two sensors obtain such values from the same organ (in our case the heart). On the other hand, in this article we do not analyze the security of IPI values, which has been widely studied in the literature (e.g., [130, 147, 162, 185]).

The rest of this Chapter is organized as follows, in Section 3.2 we provide some basic knowledge in order to facilitate the reading of the rest of the Chapter. Section 3.3 presents the core of our work, while Section 3.4 introduces our proof-of-concept implementation of the proposed solution. Section 3.5 contains a summary of the main published papers in this research area. Finally, we conclude in the last Section.

3.2 Background

In this Section we provide some preliminaries on BANs and we give a brief overview of the datasets used for the experiments. After this, we yield an overview of related work that has explored how heart signals can be applied to biometrics and cryptography. We also discuss why fuzzy extractors are often used in the literature together with biometrics. Finally, we give some background about modeling and verification of real-time systems focusing on how formal verification is

[2] http://bitalino.com/en/

used to verify the run-time monitor that we use in to synchronize two ECG signals.

3.2.1 Body Area Networks

With the recent advances on technology, manufacturers are creating small and affordable sensors that people can be equipped with in order to acquire different parameters from their vital signs. For instance, athletes usually wear chest band to measure the heart beats while training or even when they are competing. In the case of elderly people, they might be remotely monitored without the need to be in a medical center. Moreover, nowadays it is common to have smartwatches or sport gadgets equipped with accelerometers, Global Positioning System (GPS), and PPG to measure the heart rate. These devices also have communication modules such as WiFi, Bluetooth or RF.

When all these gadgets are working together, it is said that they are part of a Wireless Body Area Network (WBAN) (Figure 3.2). That is, a WBAN is a private network composed of sensors and/or actuators that measure different vital signs and send this information to a central node, typically the bearer's smartphone—which is assumed to be trusted—that acts as a gateway between the WBAN and the Internet [12, 79].

3.2.2 The Physionet Repository

Physionet [57] is a public repository composed of different databases about physiologic signals of healthy and patients with diseases. The main purpose of this repository is to allow and encourage researchers to investigate in the study of diseases and physiologic signals. Specifically, in this work we are only focusing on heart signals and, more precisely, in those databases with at least two ECG channels. That being said, popular databases such as fantasia [80] or apnea-ecg [135] are not considered in our study because there is only one ECG channel in their records. On the contrary, when more than two ECG signals are found in the same file, we are taking the first two signals we found (in a sequential order) in the *.hea* file which is a special file where the metadata of the record is stored.

In order to automate the process, we implemented a script to down-

Database	**Files**	**Peaks in $\mathbf{ECG_1}$**	**Peaks in $\mathbf{ECG_2}$**	**Heart condition**
afdb [55]	23	49003	48294	Atrial fibrillation
afpdb [119]	300	1817	1797	Paroxysmal atrial fibrillation
ahadb [77]	2	8473	8183	Healthy and ventricular ectopy
cebsdb [54]	60	360	360	Healthy
edb [170]	90	8852	881	Myocardial and hypertension
iafdb [139]	32	91	88	Atrial fibrillation or flutter
incartdb [140]	75	2263	2327	Coronary artery disease
ltafdb [138]	84	110632	108205	Paroxysmal
mitdb [120]	48	2204	2227	Arrhythmia
nsrdb [141]	18	99746	10066	No significant arrhythmias
nstdb [122]	15	2556	2544	Mitdb with noise
prcp [122]	10	4310	3355	Healthy
qtdb [96]	105	1044	1044	Holter recordings
sddb [61]	22	25969	36615	Arrhythmia
shareedb [116]	139	95809	95896	Hypertension
slpdb [74]	18	21087	23892	Sleep apnea syndrome
svdb [63]	70	2322	2323	Partial epilepsy
twadb [123]	100	185	184	Myocardial problems
vfdb [62]	22	3457	3457	Tachycardia

Table 3.1: Summary of the databases.

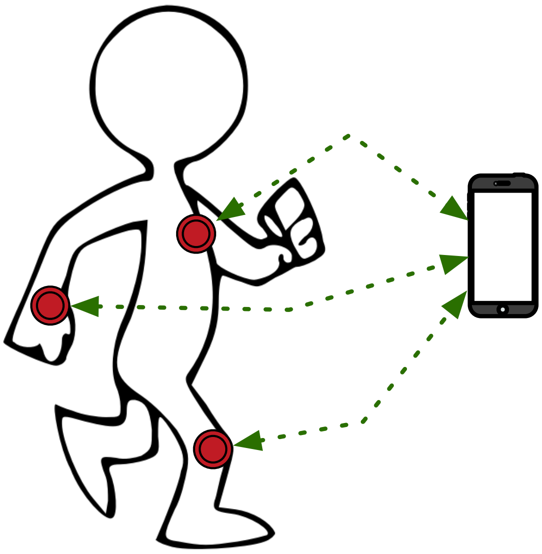


Figure 3.2: Body Area Network.

load 19 databases from the Physionet repository. A description of the databases can be seen in Table 3.1 where the number of files represents the number of patients we used in our experiments. Apart from that, we computed the average number (median) of R-peaks (heartbeats) that both the first channel of the ECG (ECG_1) and the second channel of the ECG (ECG_2) have. For each database we also included the heart condition (if any) of the patients.

From that table it is interesting to see that the number of peaks, using the well-established Pan-Tompkins algorithm for peak detection [183], is only equal in three databases: cebsdb, qtdb and vfdb whereas the values are almost equal in the iafdb and twadb databases. All the rest of the databases (14 out of 19) have different number of peaks.

Finally, Figure 3.3 shows the number of patients that cannot be considered part of the dataset because they do not reach the minimum number of IPIs which is 8, 16 and 32 to compute the tokens of 32, 64 and 128 bits respectively.

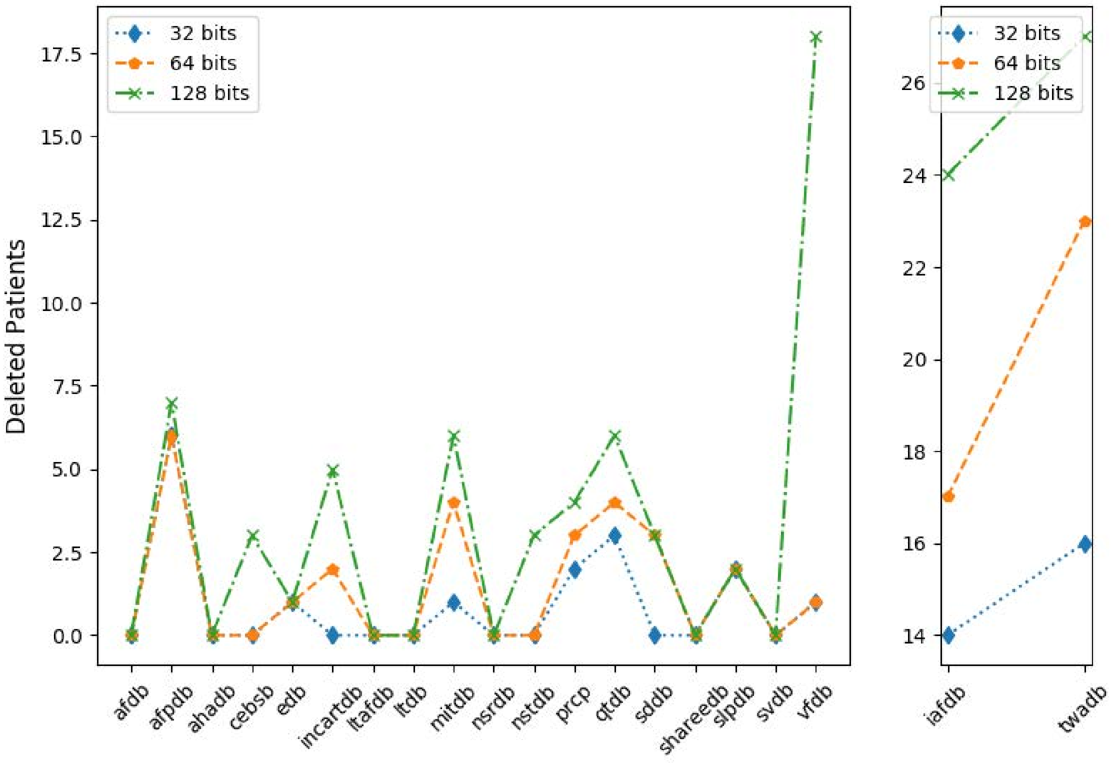


Figure 3.3: Deleted patients vs tokens length.

3.2.3 Heart Signals in Cryptography

The use of ECG signals and IPI-based approaches for cryptographic applications has been widely studied in the literature. Even though some researchers take more than the 4 LSBs of the IPI to generate cryptographic tokens, e.g., [137, 143]), the vast majority of the research community, e.g., [7, 8, 17, 34, 136, 147, 158, 160, 162, 169, 185, 200, 205] use the 4 LSBs extracted after applying the quantization algorithm explained in Section 3.3.2.1, or a slight variation of it proven to contain some degree of entropy. As we try to be as general as possible, we use the 4 LSBs to generate tokens in our experiments. It is worth mentioning that although our work is focused on IPI values, which is the most widely used approach, some authors have proposed alternative solutions which work in a transform domain (e.g., [58] or [187]).

In most of the aforementioned IPI-based works it is assumed that there are two devices listening to the heart signal and they extract a random token which is used afterwards in a cryptographic protocol. However, to the best of our knowledge, no one has performed an in-depth empirical analysis to check if it is indeed possible to extract a common token from the same signal (particularly, the ECG) gathered from different devices over the same body. Our work aims to fill in this gap and focused exclusively on IPI-based approaches.

3.2.4 Fuzzy Extractor

Juels and Wattenberg were the first who introduced the term fuzzy commitment in [84], where a cryptographic key is extracted from a biometric signal such as an ECG or an EEG. The process of generating this key is through an algorithm called *fuzzy extractor*.

Fuzzy extractors are not only applied to key generation protocols based on biometrics [45, 86, 100] but also for generating keys for authentication purposes, by using Physical Unconlable Functions (PUFs) [70, 155], and for key generation in Vehicular Ad-Hoc Networkss (VANETs) [102].

Formally, a fuzzy extractor is a function f which takes as input a biometric signal w, and produces a random string R and a public parameter P. Fuzzy extractors are particularly suitable for cryptographic protocols because when the input w' changes slightly, i.e., $w' = w + \epsilon$ for a very small ϵ, the random output R remains invariant [46].

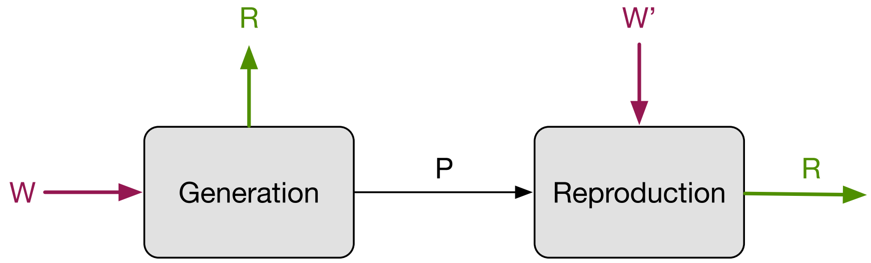


Figure 3.4: Scheme of fuzzy extractor [126].

Typically, a fuzzy extractor is composed of two main phases: *generation* and *reproduction* [46]. As it can be seen in Figure 3.4, in the generation phase, a biometric signal w is received as input and two parameters are given as output: a secret value R and a public value P. In the reproduction phase, a fresh biometric signal w' is given as input together with the public parameter P, previously generated in the generation phase. If and only if the distance between these two biometric signals—typically the Hamming distance—is less than a given threshold t_r ($Hamming(w, w') < t_r$), then the same output R will be retrieved.

3.2.5 Modelling and Verification of Real-Time Systems

Our application is a typical example of a real-time system, where a number of real-time constraints must be satisfied. Our proposed solution is based on the satisfaction of three important real-time properties concerning: i) the time between two consecutive peaks of each ECG signal; ii) the relative time between peaks from the different heart signals; iii) the total sampling time to return back a valid token. Note that this final requirement is to force the algorithm to finish its execution after a fixed time. We give some upper-bounds of these times in Section 3.3.3.2.

The design, reasoning and implementation of real-time systems have been addressed by different communities, in particular by formal methods researchers and more specifically those concerned with real-time verification [10]. In that community, the idea is to make an abstract model to represent the real-time system or some specific time constraints of the system, and apply tools to increase the confidence that the model satisfies some properties. One of the most broadly used formalism to model real-time systems is *timed automata* [9], for which reasonable mature tools have been developed to reason about, e.g., UPPAAL [21] and KRONOS [25]. In those tools, one specifies the model as a timed automata and writes properties about it on a real-time logic called Timed Computation Tree Logic (TCTL) [11].

The idea is that after performing such verification on the model, one may then write an implementation by taking the timed automaton as a starting point. Depending on the abstraction level of the model, the implementation might be more or less difficult to obtain. Though there is a gap between the model and the implementation, and errors might be introduced when an implementation is obtained from the model, it is clearly an advantage to have a verified model in the first place. As we will see later, in our case the implementation is directly obtained from the model, which gives us quite a high confidence on the correctness of our solution with respect to the specified timing constraints.

3.3 ECG-Based Token Generation Procedure

In this Section, we first explain the methodology we have followed to carry out our research. We then explain in detail how we generated tokens from different ECG signals, and demonstrate how a pre-processing phase is needed to agree on the same token generated. Finally, we propose a timed automaton satisfying our properties and create the corresponding monitor in order to synchronize the signals (and thus generate the same token).

3.3.1 Our Methodology

All the experiments presented in this section were run on a Macbook Pro 2.4Ghz with 4Gb of RAM. The processing of all the patients' signals were implemented on Matlab.

We analyze all the performed experiments and discuss the results obtained after generating two tokens independently (emulating different sensors) in 4 scenarios: 1. running a quantization algorithm (Section 3.3.2.1); 2. running a fuzzy extractor algorithm (Section 3.3.2.2); 3. running a run-time monitor (Section 3.3.3.1), and; 4. running a run-time monitor and a fuzzy extractor algorithm (Section 3.3.3.2). Finally, from our results we conclude that synchronization of the signals is a must if we want the sensors to derive the same token from the ECG signal.

Regarding the run-time monitor, two specific values need to be computed beforehand: the time between two consecutive peaks from both the same ECG channel and from different channels. To generate an upper-bound of those values with statistical significance, we require the person to be quite and calm. However, due to the fact that we are using the Physionet repository with all the signals already measured, we decided to use the mean of the time interval between R-peaks of each one of the signals as an upper-bound which is a common technique used in medical research [175]. Additionally, we set the maximum time between consecutive peaks for different signals in the case of Physionet repository to $\frac{1}{f_s}$ where f_s is the reading frequency of the device where the signal is gathered. This is due to the fact that this parameter is determined by the physical distance between sensors and in the case of the Physionet databases, all the patients were monitored using wired ECG sensors attached to their chest.

Having computed those numbers, we have verified three main properties: i) the time between two consecutive peaks of each ECG signal; ii) the relative time between peaks from the different heart signals, and; iii) the total sampling time. Similarly [158], we consider that when the time interval between two consecutive peaks from the same signal is longer than the computed upper-bound, then the monitor resets its clocks and considers that there a miss-detected peak was found. Also, when the time interval between two consecutive peaks from different signals is longer than $\frac{1}{f_s}$ then the monitor resets its clocks and considers that those peaks are not synchronized. Finally, we have proved that after t seconds, the final state is always reached and if and only if there are enough synchronized IPI then a token is computed.

3.3.2 Debunking ECG-Based Token Generation Myths

3.3.2.1 Token Generation Algorithm

Our first experiment goal was to generate as many tokens of 128 bits as possible from both channels (ECG_1 and ECG_2) of the patients of all the databases to know how different they are. In order to process the ECG signal—which is a continuous signal, it must be transformed to a discrete one. This process is known as *quantization* (e.g., uniform or dynamic quantization) and it is one of the most important steps in the token generation based on heart signals [26].

As far as we know, in the context of ECG IPI-based approaches, the dynamic quantization firstly proposed by Rostami et al. in [147] is the most extended in the literature. In a nutshell, their algorithm works as follows. First, the ECG signal is cleaned (i.e., the DC component is eliminated and then the ECG signal is passed through a pass-band filter with 0.67Hz and 45Hz cut-off frequencies [27]). Second, R-peaks are extracted from the heart signal by using Pan-Tompkins algorithm [132] and the time difference between R-peaks are computed and thus, the IPIs are generated. Third, the IPIs values are dynamically transformed into values between 0 and 1. Then the data are multiplied by 256 and rounded to the nearest integer. Finally, the Gray code encoding scheme with 8-bit of precision is used to fa-

cilitate error correction and the 4 LSBs of each IPIs are extracted to generate a token. This token is computed by appending these 4 LSBs [8, 205]; in order to create a 128-bit number at least 32 IPIs should be processed. The source code of the dynamic quantization is freely available at `https://github.com/aylara/synchro` (i.e., see `getIPIsSignal.m` file).
The pseudocode of the aforementioned IPI extraction algorithm, which is also used in this Chapter to process the signal and extract the 4 LSBs of the IPIs, is shown as Algorithm 1.

Algorithm 1 IPIs' extraction.

```
procedure IPIGENERATION(record)
    signal ← get_signal(record)
    freq ← get_sampling_frequency(record)
    cleaned_signal ← ECG_pre-processing(record)
    IPIs ← Pan_Tompkins(cleaned_signal,freq)
    IPIs ← dynamic_quantization(IPIs)
    result ← [ ]
    for ipi ∈ IPIs do
        grey← grey_code(ipi)
        $IPI_{NEW}$ ← get_LSB(gray)
        result.append($IPI_{NEW}$)
    return result
```

As mentioned before, we generate as many 128 bits tokens as possible per user per database and the result of this analysis can be seen in the second column of Table 3.2. Note that this column contains the sum of all the tokens extracted per database for only one channel (ECG_1 or ECG_2). Also, we computed how many of these tokens are similar by calculating the Hamming distance between each pair of tokens from both channels (ECG_1 and ECG_2) before performing any signal processing and compared them pairwise. The results can be seen in the third column of Table 3.2. It is interesting to see that the number of similar tokens is extremely low in all databases which means that the output of the quantization algorithm cannot directly be used to generate similar tokens in different devices.
As a conclusion, we corroborate our claim that just applying an IPI extraction algorithm like the one presented in Algorithm 1) composed of the Pan-Tompkins algorithm plus a dynamic quantization to gen-

DB	**Tokens (Alg 1)**	**Similar tokens**	**Similar tokens (FE)**	**Tokens (RM)**	**Similar tokens (RM)**	**Similar tokens (RM+FE)**
afdb	35690	8 (0.02%)	77 (0.2%)	1549 (%)	847 (54.6%)	1495 (96.5%)
afpdb	14505	40 (0.27%)	740 (5.1%)	9251 (%)	45 (0.48%)	1196 (12.9%)
ahadb	511	0 (0%)	0 (0%)	15 (%)	2 (13.3%)	14 (93.3%)
cebsdb	2577	2 (0.07%)	1360 (52.7%)	839 (%)	59 (7.0%)	835 (99.5%)
edb	24262	21 (0.08%)	497 (2.0%)	3769 (%)	1995 (52.9%)	3706 (98.3%)
iafdb	207	0 (0%)	34 (16.4%)	23 (%)	11 (47.8%)	23 (100%)
incartdb	5127	0 (0%)	69 (1.3%)	1117 (%)	4 (0.3%)	241 (21.5%)
ltafdb	271605	185 (0.06%)	1188 (0.4%)	21337(%)	92 (0.4%)	870 (4.0%)
mitdb	3198	0 (0%)	45 (1.4%)	770 (%)	0 (0%)	110 (14.2%)
nsrdb	52290	1980 (3.7%)	4072 (7.7%)	13965(%)	26 (0.1%)	1176 (8.4%)
nstdb	1160	0 (0%)	8 (0.6%)	171 (%)	0 (0%)	25 (14.6%)
prcp	825	0 (0%)	0 (0%)	22 (%)	2 (9.0%)	20 (90.9%)
qtdb	3413	1 (%)	216 (6.3%)	1346 (%)	695 (51.6%)	1315 (97.6%)
sddb	21280	1212 (5.6%)	1732 (8.1%)	1364 (%)	572 (41.9%)	1029 (75.4%)
shareedb	405775	2638 (0.6%)	4758 (1.1%)	60440(%)	297 (0.4%)	3229 (5.3%)
slpdb	8860	0 (0%)	0 (0%)	172 (%)	27 (15.6%)	169 (98.2%)
svdb	5710	2 (0.03%)	105 (1.8%)	2984 (%)	9 (0.3%)	315 (10.5%)
twadb	528	0 (0%)	31 (5.8%)	204 (%)	107(52.4%)	203 (99.5%)
vfdb	2144	0 (0%)	49 (2.2%)	221 (%)	93 (42.0%)	216 (97.7%)

Table 3.2: Number of tokens of 128-bit tokens generated by Algorithm 1 (column 2); Number of similar tokens after running Algorithm 1 (column 3); Number of similar tokens after running Algorithm 1 + Fuzzy Extractor (FE) (column 4); Number of tokens after running Algorithm 1 + Run-time Monitor (RM) (column 5); Number of similar tokens after running Algorithm 1 + Run-time Monitor (RM) (column 6); Number of similar tokens after running Algorithm 1 + Run-time Monitor + Fuzzy Extractor (RM+FE) (column 7).

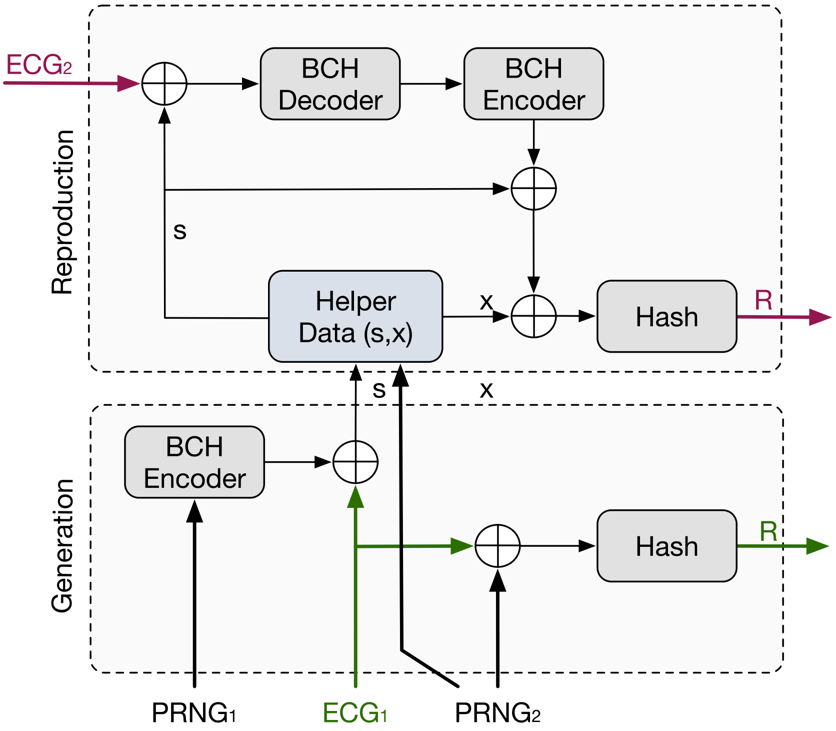


Figure 3.5: Fuzzy extractor.

erate tokens is not enough to guarantee that the same token will be generated in two different sensors.

3.3.2.2 Fuzzy Extractor

Following the scheme presented in [86], we implemented a fuzzy extractor algorithm, which was specifically adapted to work with ECG signals. The scheme of the fuzzy extractor can be seen in Figure 3.5. The fuzzy extractor takes as input two ECG signals (ECG$_1$ and ECG$_2$) and two random numbers (PRNG$_1$ and PRNG$_2$). The ECG$_1$ and the mentioned random numbers are provided in the generation phase since they are needed in the computation of the *Helper Data* (i.e., (s, x)), which is used in the reproduction phase together the ECG$_2$ signal. The result of the fuzzy extractor is a pair of identical values R. Our fuzzy extractor is publicly available at `https://github.com/aylara/synchro` (i.e., see `simulation_fuzzyextractor.m` file).
We assign the following values for the parameters m, n, k and t of the BCH: $m = 7$, $n = 127$, $k = 50$, $t = 13$), following the guidelines given in [86].The parameter $t = 13$ represents a trade-off between the correction capability and the the ability the adversary has to break

the protocol. Thus this parameter should not be increased arbitrarily since it would increase the success probability of an adversary. This means that the BCH can recover at most 13 different bits from words of 128 bits (i.e., a 10% of the bits). We urge the reader to consult [67] for a detailed description of BCH parameters and their implications. An additional argument for this 10% value ($t = 13$) is that we have empirically demonstrated that is not possible to achieve 90% of similarity in the tokens generated without our run-time monitor together with the fuzzy extractor (see columns 3 and 4 from Table 3.2).

In order to check how the fuzzy extractor behaves, we used the output of the Algorithm 1 as input of the fuzzy extractor and computed the Hamming distance between each pair of tokens compared tokens pairwise and the results can be seen in the third column of Table 3.2. Even if our fuzzy extractor produces a slight improvement, with respect to the results obtained without performing any pre-processing of the signal (see column 2), the results are far from being the expected ones. For instance, the cebsdb database which achieves a 52.7% of the similar tokens in both channels is not a good result, i.e., 1 out of 2 generated tokens is random. The reason for getting these poor results stems from the fact that the distance between the IPIs calculated from each sensor clearly exceeds the correction capacity of the fuzzy extractor (BCH encoder). In our experimentation, for words of 128 bits the correction capacity is set to $t = 13$.

3.3.3 How to Generate ECG-Based Tokens

3.3.3.1 Timed Automata

Timed automata are composed of five main parts: clocks, time-checks, actions, events and states. For our timed automaton, we have defined three different clocks, namely c_1, c_2 and c_3, which are in charge of checking the time properties of the heart beats in our model. Concretely, c_1 checks an upper bound for the execution of the automaton, that is, how long the automaton should be executed; c_2 checks when the peaks from both signals are synchronized or not, and; c_3 checks when there are missed peaks in the same signal.

All the time checks used in the automaton were obtained after having analyzed all databases. We show in Figure 3.6 a representation of these time checks. More concretely:

t_c This value varies in time and between each person. In order

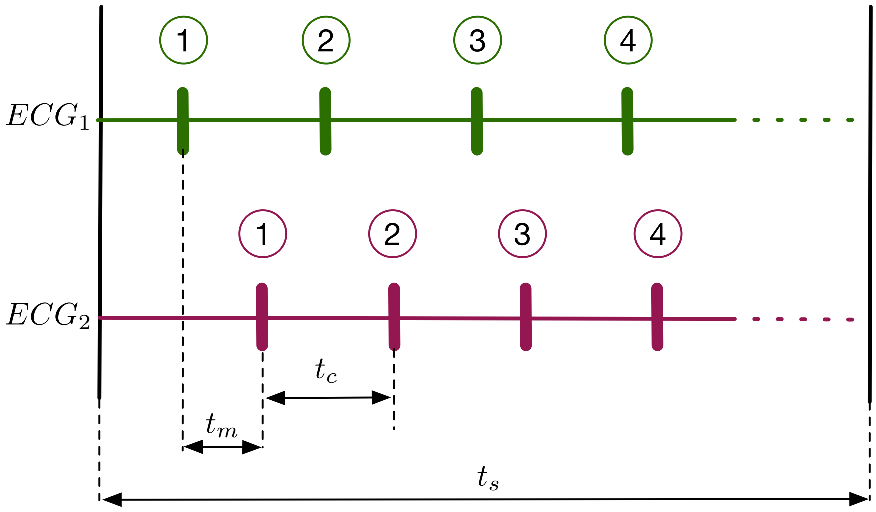


Figure 3.6: Time-checks used in the timed automaton.

to compute t_c, as stated in Section 3.3.1, the person should be in a quite and peaceful environment. For our experiments and following the similar technique proposed in [175], we calculated the mean time between R-peaks of each pair of ECGs (ECG$_1$ and ECG$_2$), which is the value assigned to the time-check t_c.

t_m This value is determined by the physical distance between sensors and hence, it is directly affected by the speed of the blood pumped from the heart to the rest of the body. In our particular case, all the databases of the Physionet repository always consider electrodes attached to the chest of the patients, so we forced this value to be less than $\frac{1}{f_s}$ where f_s is the sampling rate. So, $t_m < t_c$, otherwise a missed peak is detected and discarded by the automaton.

t_s This value is a bound that determines how long each "session" of the execution of the monitor should be. We set this value to be equal to the longest signal encountered in our databases, in order to ensure we consider all the signals.

Table 3.3 shows all the variables and constants of our automaton (shown in Figure 3.7), defined as a tuple $\mathcal{A} = \{L, X, \Sigma, \Delta, F\}$, where:

- $L = \{E_0, E_1, E_2, E_3, E_4\}$ is the set of locations (with E_0 and E_4 the initial and final states, respectively);
- $\varSigma = \{Log, Reset, ReturnPeaks, Sync\}$ are all the actions;
- $X = \{c_1, c_2, c_3\}$ is the set of clocks;
- $\Delta \subseteq L \times X \times \Sigma \times 2^X \times L$ is the transition relation, where $F \subseteq L$ is a set of accepting locations.

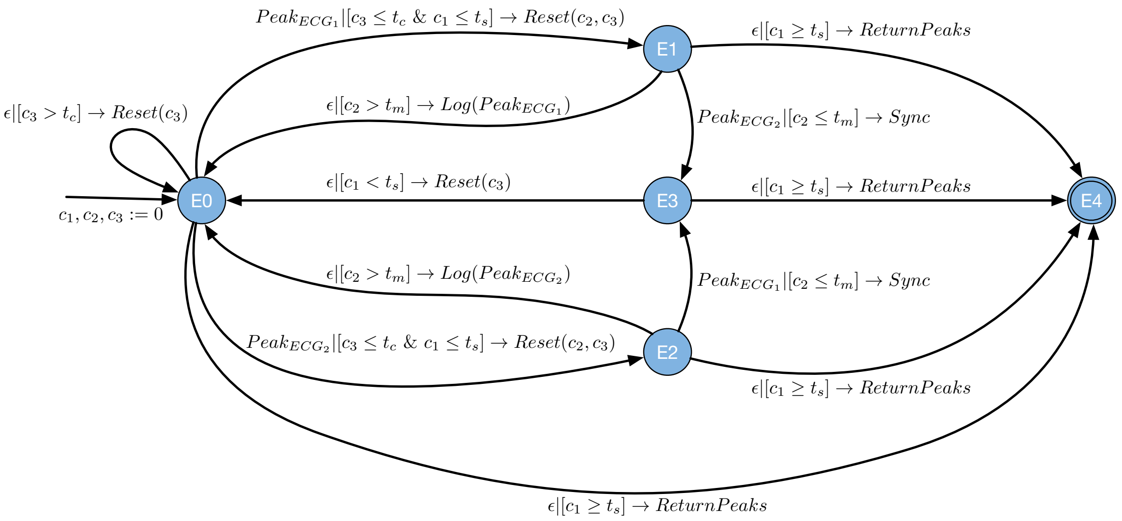

$Peak_{ECG_1}|[c_3 \le t_c \ \& \ c_1 \le t_s] \rightarrow Reset(c_2, c_3)$
$\epsilon|[c_1 \ge t_s] \rightarrow ReturnPeaks$
$\epsilon|[c_3 > t_c] \rightarrow Reset(c_3)$
$\epsilon|[c_2 > t_m] \rightarrow Log(Peak_{ECG_1})$
E1
$Peak_{ECG_2}|[c_2 \le t_m] \rightarrow Sync$
$\epsilon|[c_1 < t_s] \rightarrow Reset(c_3)$
$\epsilon|[c_1 \ge t_s] \rightarrow ReturnPeaks$
E0
E3
E4
$c_1, c_2, c_3 := 0$
$\epsilon|[c_2 > t_m] \rightarrow Log(Peak_{ECG_2})$
$Peak_{ECG_1}|[c_2 \le t_m] \rightarrow Sync$
$Peak_{ECG_2}|[c_3 \le t_c \ \& \ c_1 \le t_s] \rightarrow Reset(c_2, c_3)$
E2
$\epsilon|[c_1 \ge t_s] \rightarrow ReturnPeaks$
$\epsilon|[c_1 \ge t_s] \rightarrow ReturnPeaks$

Figure 3.7: Heart based timed automaton.

The Log action keeps a list of those IPIs which are not synchronized according to our time constraints. Reset initializes the clocks given as input. *ReturnPeaks* returns the list of non-synchronized IPIs. Finally, the Sync action computes the list of IPIs which are synchronized.

Regarding the *events*, we have two types: when a peak comes from the ECG$_1$ or from the ECG$_2$. Additionally, we have defined ϵ which means that we do not wait for any event to occur and we force the runtime monitor to check if the condition is satisfied to perform the transition to the corresponding state.

The automaton has 5 *states*. All the clocks and variables are set to 0 in the initial state E_0. Note that whenever $c_1 \ge t_s$ then the computation finishes (the automaton is in state E_4). The rest are intermediate states, ensuring progress in the computation provided the relevant timing constraints are respected ("accepting" or "rejecting" peaks).

We implemented our timed automaton in Uppaal [21], allowing us to validate and verify our model in a formal way. Our verified model was then translated into a *runtime monitor* implemented as Matlab code. The source code of both implementations are available at `https://github.com/aylara/synchro` (i.e., see `Automaton` and `UPAAL` folders). We tested our generated runtime monitor with the output of the Algorithm 1. The number of tokens has decreased considerably as it can be seen in the fifth column of Table 3.2. After that, we then computed the Hamming distance between each pair of tokens compared pairwise and the results can be seen in the sixth column of Table 3.2. Note that, in general, the number of similar tokens has increased considerably af-

Clocks	
c_1	Sampling time
c_2	Time between peaks of two signals
c_3	Time between two consecutive peaks (same signal)
Time-checks	
t_c	Time between two consecutive peaks (same signal)
t_m	Time between peaks of two signals
t_s	Sampling time
Actions	
Log	Stores those IPIs which are not synchronized
$Reset$	Initializes the clocks given as input
$ReturnPeaks$	Returns the non-synchronized IPI set
$Sync$	Checks what IPIs are synchronized
Events	
$Peak_{ECG_x}$	R-Peak of ECG_x, where $x \in [1, 2]$
ϵ	No event
States	
E_0	Initial state
E_1	When a peak of the first signal is detected
E_2	When a peak of the second signal is detected
E_3	When one peak of each signal is detected
E_4	When the max time is detected ($c_1 \geq t_s$)

Table 3.3: Properties of the automaton (Figure 3.7).

ter running the run-time monitor with respect to the third column. However, this improvement does not come for free. The penalty we have to pay is that the number of tokens has decreased per database as it can be seen in fifth column of such a Table.

3.3.3.2 Timed Automaton & Fuzzy Extractor

As already explained, our approach consists of combining our monitor (extracted from our verified timed automaton, for synchronizing the tokens (based on IPI values) extracted from two different ECG signals) and the fuzzy extractor (to correct some bits).

The results can be seen in the last column of Table 3.2. After applying this solution we can successfully generate the same token from different sensors with high probability in the majority of the databases, i.e., 10 out of 19 databases have a probability higher than 90% of taking two similar tokens generated on different sensors. However, despite of our method improves the current state of the art, it will remains low for 8 databases, namely afpdb, incartdb, ltafdb, mitdb, nsrdb, nstdb, shareedb and svdb whereas sddb achieves a 75.4% of probability that two arbitrary tokens be similar.

From the above results, we can clearly conclude that the best databases to be used to extract cryptographic tokens are the ones with healthy patients. Moreover, our method seems to work reasonably well with those patients whose disease is not severe. Hence, we recommend not to use databases such as mitdb which is widely used in the research community for security purposes [14, 143, 147, 158, 206] or nsrdb [204, 205] to mention a few.

Having empirically demonstrated the effectiveness of our proposed method, the only question that remains uncovered yet is how long the run-time monitor needs to listen to the heart signal in order to obtain a token which can be used later on as part of a cryptographic protocol.

We conducted an additional experiment to measure how long the run-time monitor needs to keep listening an ECG signal in order to produce a token of 1. 32 bits (Figure 3.8); 2. 64 bits (Figure 3.9), and; 3. 128 bits (Figure 3.10). To carry out this test, we modified the original timed automaton (Figure 3.7) in such a way that, instead of having 3 different clocks (c_1, c_2 and c_3), we only keep c_2 and c_3, and replace c_1 by a counter. By doing so, as soon as the automaton detects that the length of the token is 32, 64 or 128 respectively, then the final state E_4

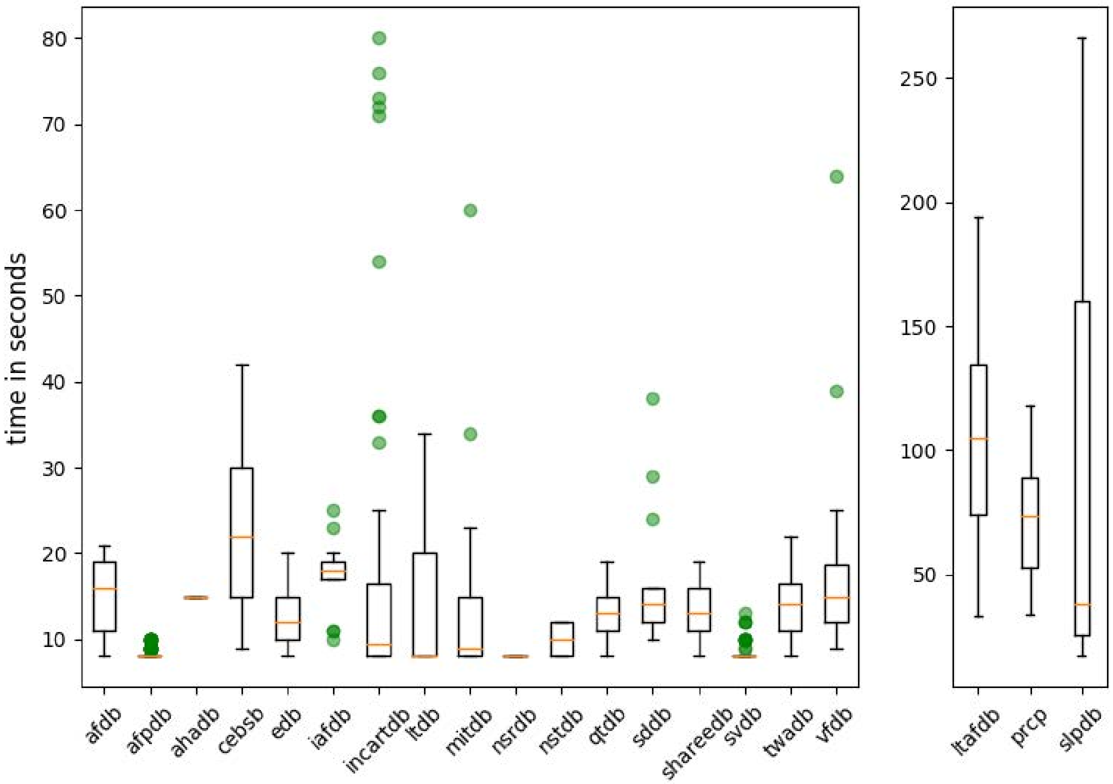


Figure 3.8: Time needed to generate a 32-bit token.

is reached. Roughly speaking, taking into account that we can only extract the 4 LSBs from an IPI, the automaton will stop when it finds 8, 16 or 32 synchronized IPIs. We have then re-implemented our new automaton in Matlab, getting a new monitor.

Furthermore, in order to make Figures 3.8 to 3.10 more readable, we decided to discard some of the outliers and we kept the 70% of the original data. It can be observed that, in order to get a 32-bit token, sensors need to listen approximately for 13 seconds on median. Similarly, to get a 64-bit or 128-bit token, they should listen the ECG for 28 and 56.5 seconds, respectively, on median. It is also noticeable that although at a first sight the above mentioned timing values might appear to be excessive, this generation process will only be executed once, typically in the setup phase of the cryptographic protocol (e.g., key generation and synchronization processes between two sensors).

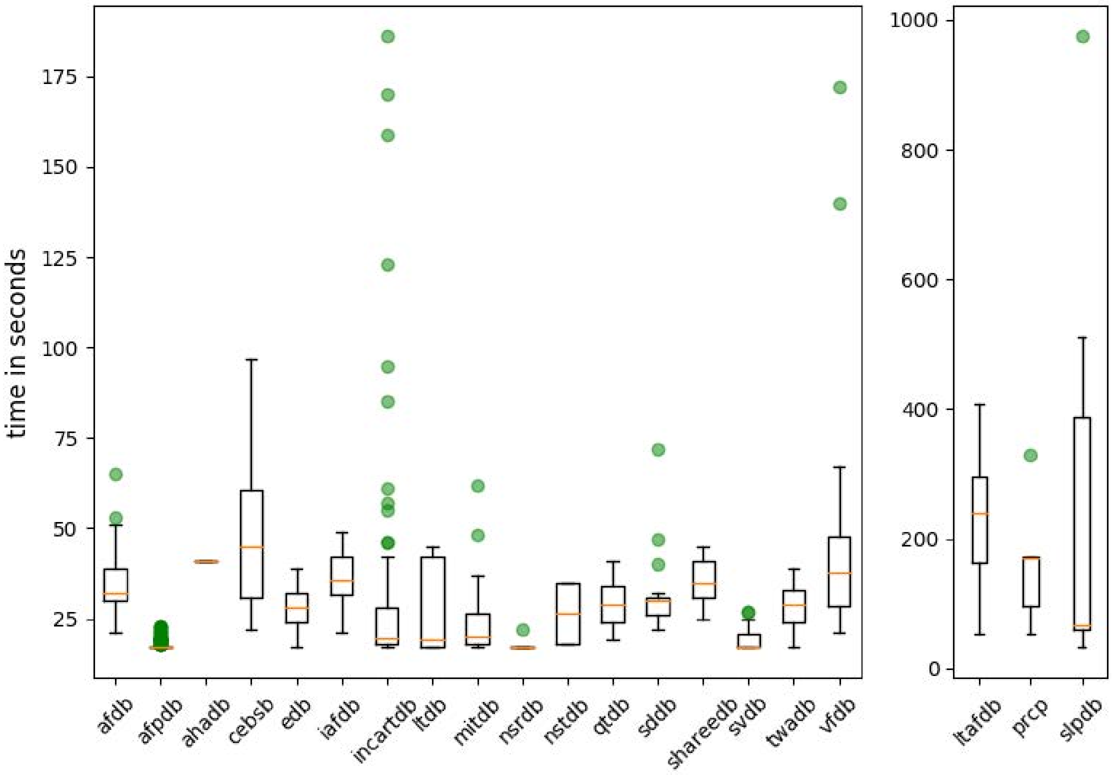


Figure 3.9: Time needed to generate a 64-bit token.

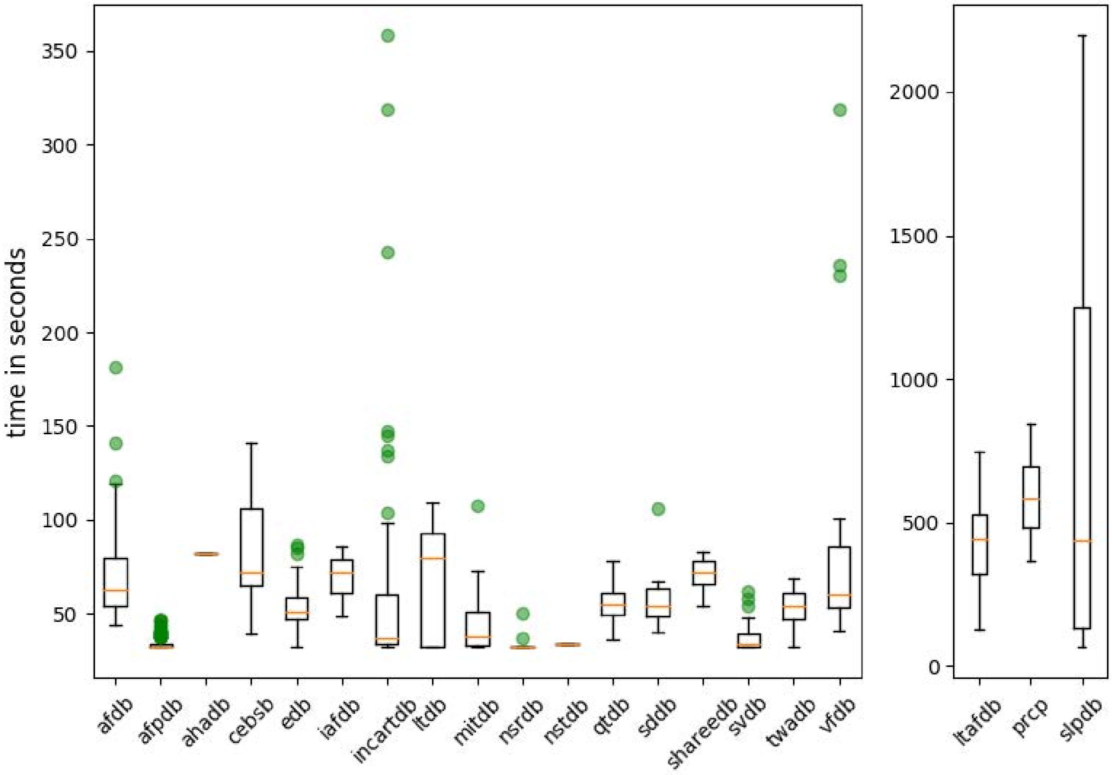


Figure 3.10: Time needed to generate a 128-bit token.

3.4 Proposed Solution

In this Section we provide a proof-of-concept implementation to demonstrate whether two sensors can derive the same token from the heart signal using real hardware. The purpose of the presented proof-of-concept is to show the feasibility of our solution as well as the minimum requirements for generating common tokens on different ECG sensors. For this first approach, and similarly to previous proposals (e.g., [154, 163, 202]), we assume the communications to be secure between the sensors and the gateway (at least during the set-up phase). Alternatively, we could have used noisy cryptography [177] to share sensitive information (the two ECGs in our particular case) via insecure channels.

A real example in which the above scenario can occur is as follows. Imagine that Alice is wearing a smart T-shirt with an ECG monitor similar to the one proposed in [194]. This T-shirt is already paired with her smartphone which makes the communication channel secure. Additionally, she has a pacemaker which is as well paired with the smartphone. In this scenario, the smartphone is acting as a WBAN gateway due to its computational resources in terms of CPU, storage, memory and communication capabilities. The above example is integrated within what is called body area networks. Another example, perhaps more futurist and within the area of the intelligent and connected cars could be the following. A driver (Bob) holds a smart-watch with an ECG sensor—note that this type of product is already on the market [31]. As for the car, the steering wheel, and as a novelty, also has an ECG sensor [32, 156]. As in the above example, both ECG sensors are securely connected to the car's central control system that acts as gateway. The two described examples are completely different but in both scenarios the two sensors must calculate the same cryptographic token (derived from the ECG recorded by each sensor) with the help of the gateway.

Summarizing, our system has three entities: a gateway and two ECG sensors (e.g., smart T-shirt and pacemaker).[3] Figure 3.11 provides an architectural view of the system. Once the ECGs signals have been gathered by the sensors, they are sent to the gateway in order to be synchronized by using the timed-automation (see Section 3.3.3.1 and Figure 3.7). After that, the already synchronized signals are sent

[3] Note that it could have also been possible to use one of the sensors as a gateway.

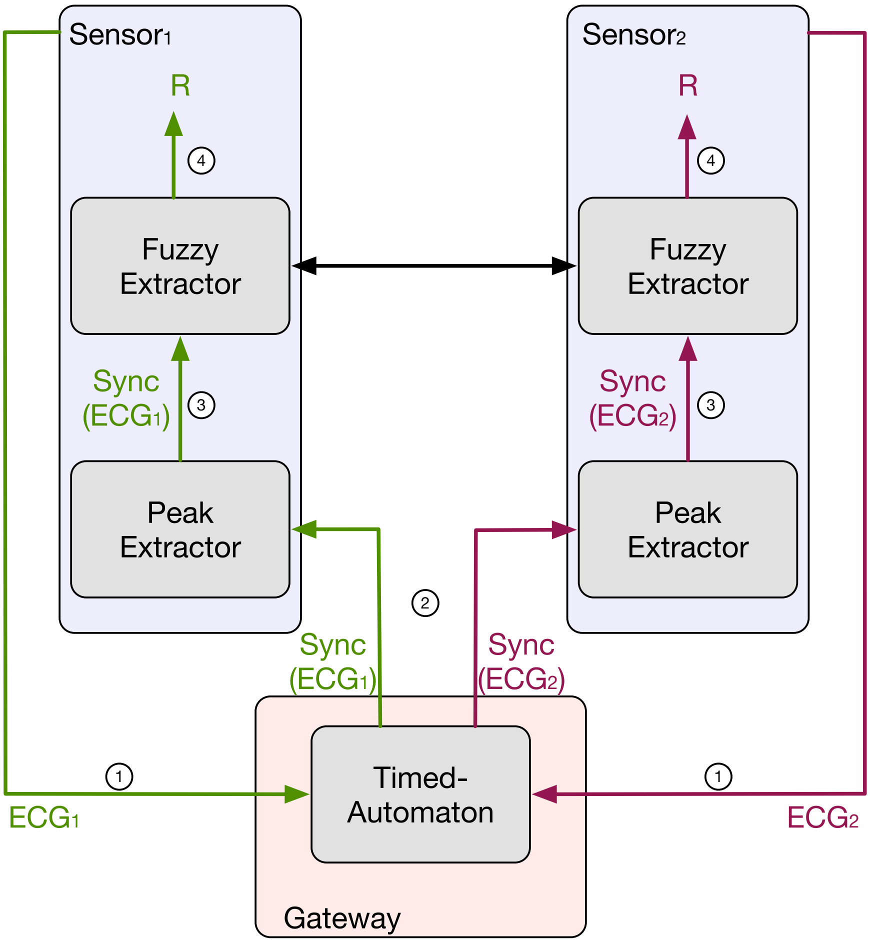


Figure 3.11: System model using both a timed automaton and a fuzzy extractor.

back to the sensors, the peak extractor procedure (Algorithm 1) is executed to extract the tokens, and finally the fuzzy extractor (see Section 3.3.2.2 and Figure 3.5) is applied to the processed signals in order to generate the same cryptographic token.

We have basically deployed the scheme presented in Figure 3.11 by using a low-cost hardware dedicated for research purposes named BITalino[4]. This shield has two ECG channels and a Bluetooth connection. As in the Alice example, we had to pair our hardware with the gateway which, in our case, was a laptop as can be seen in Figure 3.12.

More concretely, it works as follows: 1) First, sensors measure the heart signal and send the gathered ECGs to the gateway via a secure communication channel. Once the signal is received by the gateway, the run-time monitor is executed in order to synchronize both signals.

[4] `http://bitalino.com/en/`

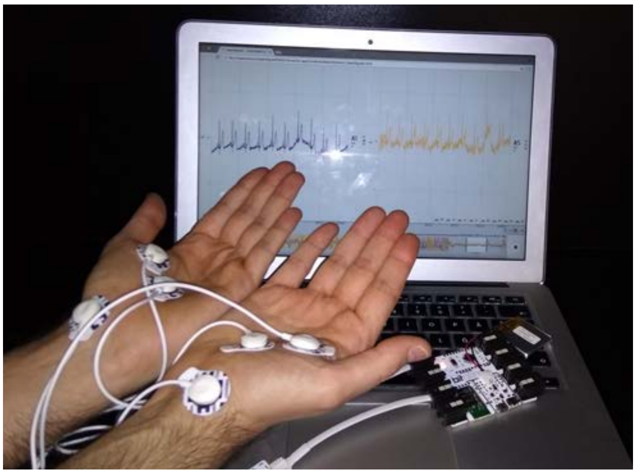


Figure 3.12: Proof-of-concept based on the BITalino platform.

2) After the signals are synchronized, the sensors receive the position of the peaks that must be removed by the gateway. For this communication to occur a secure channel is also needed. 3) When the the list of peaks to be removed is received, the sensors proceed to delete them—note that at this point both signals are synchronized in terms of R-peaks—in order to proceed with the token generation. 4) Finally, a fuzzy extractor is applied to the processed signal in order to generate the same token. The helper data can be transmitted from one sensor (generation process) to the other one (reproduction process) without the necessity of a secure channel.

It is worth mentioning that the gateway cannot generate a token by itself to be used in the WBAN; its role is to synchronize the signal and thus helping the sensors to generate a shared token. However, at the end of this protocol, not only the sensors can generate the same token but also the gateway may do it. The source code of our implementation can be downloaded from `https://github.com/aylara/synchro`.

3.4.1 Security Analysis

In order to solve the problem of creating the same token in two sensors, two aspects need to be addressed: i) a secure communication channel (sensor(s)–gateway) must be used, and; ii) sensors have to share their gathered ECGs with the gateway to synchronize them.

Traditionally, authors have assumed that different sensors can extract the same token by measuring the same signal from an organ (in our case the heart). Most authors rely on the fact that, in order to derive the same token, the (active) attacker must be reading the ECG of the bearer at the same time as the devices are and such probability is almost negligible [198]. On the other hand, other proposals (e.g., [147, 158]) assumed that the communication can only be established if the devices are close enough (commonly named as neighborhood area): the attacker should be a few centimeters from them and would be easily detected. In these security protocols, the ECG is used to derive a common secret between different sensors and thus, the ECG can be considered as the secret key. The signal must therefore be transmitted over a secure communication channel.

A more recent approach was presented in [154]. In this work, authors use the ECG to securely distribute symmetric cryptographic keys. However, authors use a trusted central node in charge of establishing a secure communication between sensors that want to share some data. In this approach all the devices (two sensors and the gateway) are on the body as they need to record the ECG at the same time and by themselves—in our case only the two sensors have to collect the ECG signal—and the secure channel is established by using a fuzzy commitment scheme [84]. In this Chapter, we demonstrated that only by using a fuzzy commitment scheme is not enough to generate the same token in different devices. In [154] and our proposal, once the cryptographic token synchronized between the sensors is established, they can exchange information directly without involving anyone else. For instance, after the setup phase, smart-watches, wrist-bands and IMD can securely exchange data with each other regardless of the brand, manufacturer or the purpose of the device.

It is important to note that our proof-of-concept implementation is secure if and only if the communication channel between the sensors and the gateway is secure (like most of the proposed solutions in this field). This is because both sensors are sending the ECG to the gateway and thus, an attacker ($\mathcal{A}$) can sniff the communication channel, extract the signals and perform the matching operation. The only extra information that $\mathcal{A}$ would need is the specific instant the tokens have started to be computed. It is also remarkable that the secure channel requirement is only used for the very first time (set-up phase of the protocol); once the two ECG signals are synchronized, there is no longer any need to keep the channel secure.

In this Chapter we empirically demonstrated that the assumption of two sensors deriving the same token from the heart (ECG signal), is at least questionable. We showed that, in addition to the error correction techniques, a new step is needed before extracting such a token: the synchronization of the signal. To achieve this, there are two options: 1) one of the sensors sends the ECG to the other one in order to synchronize the signal and the latter sends the synchronized signal back to the first one, or; 2) a trusted and external party is used to synchronize the signal and communicate the final decision to the sensors. Either way, the main consequence from a security point of view is that now Eve—a passive attacker, just by eavesdropping on the communication channel might synchronize both ECG signals and extract a common key. To combat this we proposed two main approaches: 1) assume a secure channel in the set-up phased, or; 2) assume that the channel is insecure all the time and use some protection mechanism such as solutions based on noisy cryptography [177].

3.5 Related work

Several studies have been done in the area of security and privacy applied to biometrics, and in particular where heart signals are involved (e.g., [144, 147, 205]] In most of these works, there are three main assumptions: 1) bits extracted from the heart signal can be considered random [147, 162]; 2) two sensors placed in the same body can generate the same random token from the heart signal [17, 158, 185, 206], and; 3) two sensors should gather 32 consecutive peaks in order to generate a 128 bits nonce which is approximately a 32 seconds signal [200, 205]. As far as we are concerned, this work is the first one that empirically demonstrates that the usual assumptions made in the aforementioned papers regarding the token generation in different devices at the same time are at least questionable.

On the one hand, it is usually assumed that the random numbers derived from IPIs can be directly used on cryptographic applications because of the high entropy degree that the 4 LSBs have. Additionally, some researchers have tried to improve the strength of the entropy per IPI in order to guarantee a higher security level [160, 162, 185]. However, Ortiz et al. questioned the entropy quality of the IPI values and the dependence of the results on the dataset used [130]. In this Chapter, the involvement of the two sensors is necessary because thanks to this (and the acquisition of the same IPI values derived from the recorded ECGs) they mutually verify they are close to each other (i.e., that they are in the neighbourhood area as is commonly named in distance bounding protocols) and additionally they can authenticate each other. Therefore, this kind of distance checking (and mutual authentication if necessary) is made by the participation of both sensors (on the same organ).

On the other hand, researchers have usually assumed that a person equipped with different heart sensors can extract the same nonce from the ECG by using a fuzzy extractor (see [204] for a comparison between fuzzy commitment and fuzzy vault schemes). For example, authors in [147] propose a security protocol where a patient equipped with an IMD and a doctor with a Programmer can extract the same nonce from the patient's ECG. Similar assumptions are made in [144, 195, 206], just to cite a few of them. Contrarily to any prior proposals, in this Chapter we demonstrated that error correction algorithms, such as fuzzy extractors, are not enough to claim that sensors placed in different parts of the body can generate the same token using an IPI-based approach.

Recently, some authors have taken into account that some events may occur during the measurement process that increase the difference between the generated tokens. For instance, it is possible that noise appear in the extraction of the signal and the detection of heart peaks will be affected [158]. Because of that, [158] proposes a mechanism to statistically calculate where a peak should be in the ECG signal and they manually add it so that the entropy degree is not affected. Also, other parameters such as HRV and VAR_{is} can alter the peak detection algorithm [104] and they must be taken into account to ensure that the keys are equal enough. In this Chapter, we not only take those issues into consideration but also the heart signal is never modified so that other computations can be applied over the signal such as medical checks of the heart.

Finally, it was stated in [17, 124, 147, 195], that sensors have to keep listening the ECG for about 30 seconds to generate a 128 bits token. In our work we have proved that in order to generate a 128 bits token, two sensors should be reading the heart signal for almost 1 minute (56.6 seconds) on median, time which we have obtained experimentally and it is in general much larger that it was previously claimed.

3.6 Conclusion

In this Chapter we tested whether error corrections algorithms, including fuzzy extractors, can be used alone to claim that two different sensors are able to derive equal tokens from two ECG signals measured at different parts of the body by using an IPI-based approach as proposed in many previous works [17, 147, 160, 162, 195]. We run the experiments against 19 public databases from Physionet repository, and we can clearly conclude that a pre-processing of the heart signal is mandatory for generating the same token. Because of that, we proposed a run-time monitor, based on a timed automaton, to synchronize both ECG signals before the peaks are computed and before the fuzzy extractor takes place. Finally, we run once again the same experiments and errors are reduced to zero in many of the tested databases.

Additionally, we also conducted one more experiment to check how long the sensors should record the heart signal in order to generate tokens of 32, 64 and 128 bits and, contrarily to what it is usually assumed (6, 12, and 24 seconds for individual with a heart rate of 80 bpm), the sensors have to wait 13, 28 and 56.5 seconds on median respectively to derive the same token from two ECG sensors.

4

Are the Interpulse Intervals of an ECG Signal a Good Source of Entropy? An In-depth Entropy Analysis Based on NIST 800-90B Recommendation

4.1 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 [14, 58]. It can be seen how since the first paper appeared in 2004, proposing that the heart signal might be applied to cryptography [18], 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. [18] and later improved by Xu et al. [195] in 2011, the most common one was proposed by Rostami et al. [147] two years later based on the previous ones. After then, many authors have used such quantization algorithm [8, 91, 159, 161] or a slight modification of it [88] to extract a subset of the LSBs from each Inter-Pulse Interval

(IPI) (i.e., time interval between two R-peaks or heartbeats) due to its claimed entropy property [147].

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 [159, 88], authentication protocols [147, 186, 22] or peak misdetection algorithms [91, 157]. 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 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 [143]. 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 [8, 159, 161, 142]. On the contrary, there are others who just say that they test the entropy, providing no more information [93, 88] or even there are some authors who directly do not check the entropy but run some random test instead like the NIST STS [17, 71, 201] which, as Rushanan et al. [149] pointed out, is not enough to claim that the 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 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.

4.1.1 Overview of Our Results

In this work, we analyze the entropy of the LSB values extracted from a heart signal according to the NIST STS recommendation (i.e., SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation [176]). To facilitate the replication of our results by other researchers, we downloaded and used 19 databases from the Physionet public repository[1] [57]. 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 [176]. 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 4.2 provides some background on biometric authentication using an ECG signal and also a necessary explanation of some random tests. Section 4.3 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 4.4 while this paper ends with some conclusions and future work in Section 4.5.

4.2 Background

4.2.1 Dataset and IPI Extraction

Dataset We first downloaded 19 databases from the Physionet repository [57] which contain the heart information of several subjects/patients and their ECG signals in one or several channels. In these datasets, we can find from healthy people as in cebsdb to patients with myocardial diseases as in edb. Table 4.1 shows a summary of the main characteristics of the 19 datasets used all throughout this work.

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

Database	#Patients	Pathology
afdb [55]	23	Atrial fibrillation
afpdb [119]	300	Paroxysmal atrial fibrillation
cebsdb [54]	60	Healthy volunteers
edb [170]	90	Myocardial and hypertension
fantasia [80]	40	Healthy
iafdb [139]	32	Atrial fibrillation or flutter
incartdb [140]	75	Coronary artery disease
ltafdb [138]	84	Paroxysmal
mitdb [120]	48	Arrhythmia
nsrdb [141]	18	No significant arrhythmias
nstdb [122]	15	Mitdb with noise
prcp [69]	10	Healthy
qtdb [96]	105	Holter recordings
sddb [61]	22	Arrhythmia
shareedb [116]	139	Hypertension
slpdb [74]	18	Sleep apnea syndrome
svdb [63]	70	Partial epilepsy
twadb [123]	100	Myocardial problems
vfdb [62]	22	Tachycardia

Table 4.1: For each database the number of patients and the pathology (if any) of the patients involved.

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 [195, 147]. There are, on the contrary, some authors who use more bits than the four LSBs [143, 137]. 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 4.3) 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 [132] 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

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

obtain the so-called IPI values.

Once we computed the IPIs, we run the *quantization* algorithm proposed by Rostami et al. [147], which is a slight variation of the quantization algorithm first proposed by Bao et al. [18] and later improved by Xu et al. [195]. This algorithm fundamentally transforms a continuous signal into a discrete one and applies a Grey code to decrease the errors of the signal.

We took the public source code recently released by Ortiz et al. [131] 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.

4.2.2 Entropy & NIST

The concept of entropy was first introduced in 1948 by Shannon in [164]. 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, then the information entropy is low. 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 4.3 to see the complete example).

In 2012, the NIST STS published a draft with some recommendations for the entropy sources used for random bit generation [19]. The final document (NIST SP 800-90B) was recently published—early 2018—

and can be seen in [176]. 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 STS 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 [150]. This last concept has been recently used by Chizari and Lupu [38] to measure the entropy of the heart signal.

Regarding the size of the dataset, the NIST STS 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 1,000 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 4.2, 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 [147, 143, 124, 196].

In our work we are using the min-entropy estimators proposed by the NIST STS 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 STS.

The Most Common Value Estimate This test first finds the pro-

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

Table 4.2: 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.

.

portion 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 [65], 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 [114], 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.

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 [153] 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 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 4.3. 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`

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

Table 4.3: 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.

seems a good source of entropy.

4.3 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.

Figure 4.1 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,

[3] See [147, 131] for more details about the quantization algorithm as well as for the source code.

IPI	1	0	1	0	1	0	1	0
	IPI_1	IPI_2	IPI_3	IPI_4	IPI_5	IPI_6	IPI_7	IPI_8

Figure 4.1: Position of bits in IPIs.

6th and 8th positions of the IPIs (i.e., $\text{IPI}_2||\text{IPI}_6||\text{IPI}_8$) belonging to a concrete database, e.g., "000" in the example of Figure 4.1. Another example might be the combination of bits 78 (i.e., $\text{IPI}_7||\text{IPI}_8$), which can also be read as the file made of the last 2 LSB of IPIs, e.g., "10" in the example of Figure 4.1.

After running the min-entropy (using ten estimators) in our previous example against three well-known examples (see Table 4.3), 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 4.3.5, 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 [8, 159, 143, 124, 206].

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 STS 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 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 4.2.2) 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 STS 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 40Gb 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.

4.3.1 $\mathbf{V_2(8)}$ Variations of two bits without repetition

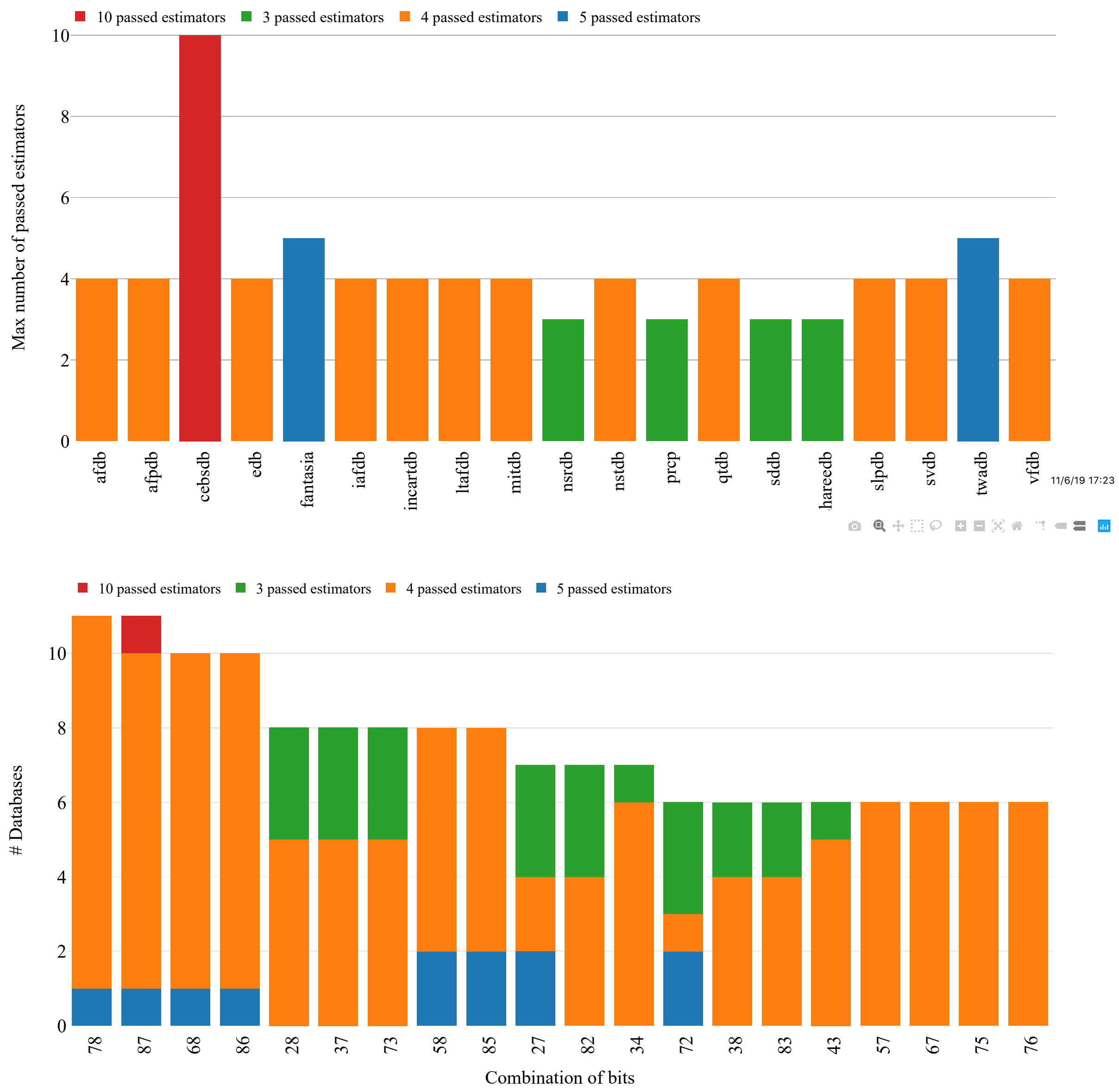
We analyzed the results after running the min-entropy estimators for variations without repetition of 2 bits and the results are shown in Figures 4.2 and 4.3. In average, it can be seen how in most of the databases, the maximum number of successful tests (estimator higher than 0.7) is four (see Figure 4.2a). 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 Figure 4.2b.

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 Figure 4.3a 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 STS 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, Figure 4.3b 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.

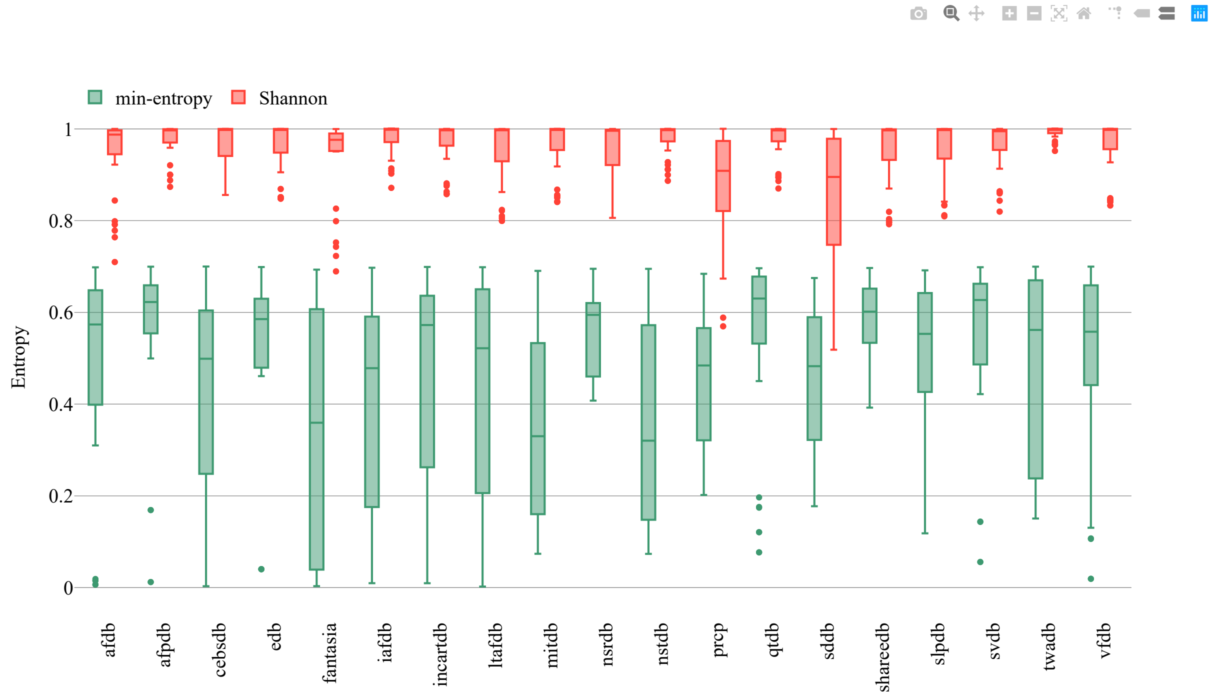


(b) Best combinations.

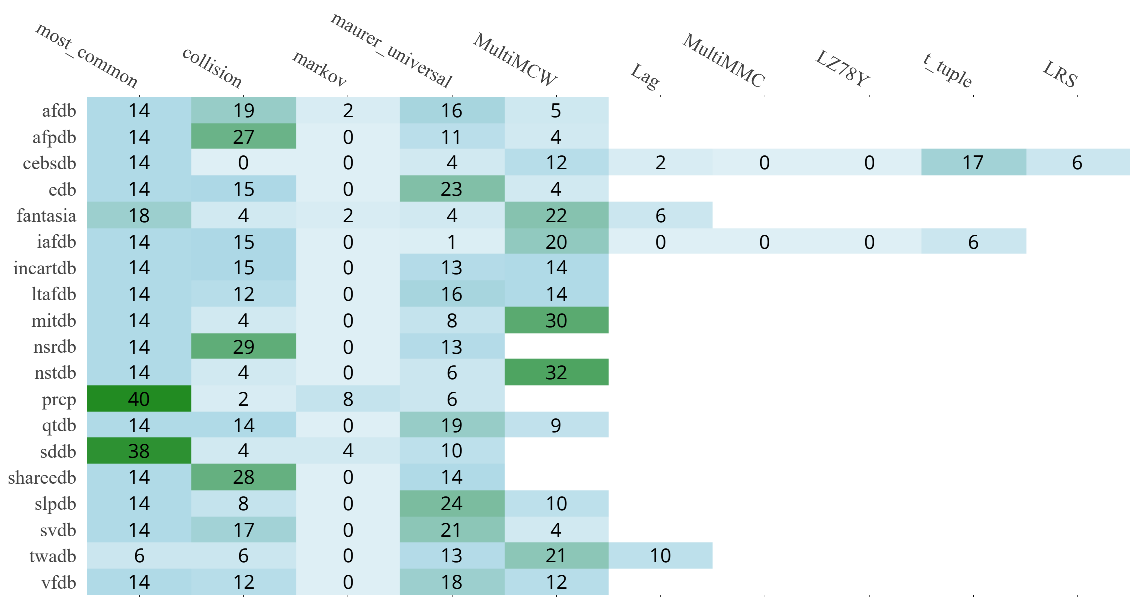
Figure 4.2: Entropy analysis of files generated by extracting 2 bits from IPIs. Figure 4.2a represents the maximum number of passed estimators that achieves at least one combination of bits. Figure 4.2b shows the best and most common combination of bits of databases.

file:///Users/pablop/Documents/MATLAB/combinatorial_ipi/2_bits_most_common.html Página 1 de 1

15/4/2019 2_bits_shannon.html



15/4/2019 heatmap_2_bits.html



	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.

Figure 4.3: Entropy analysis of files generated by extracting 2 bits from IPIs. Figure 4.3a depicts a comparison of the min-entropy and the Shannon entropy. Figure 4.3b shows a heatmap of the most failed estimators per database.

file:///Users/pablop/Documents/MATLAB/combinatorial_ipi/heatmap_2_bits.html 1/1

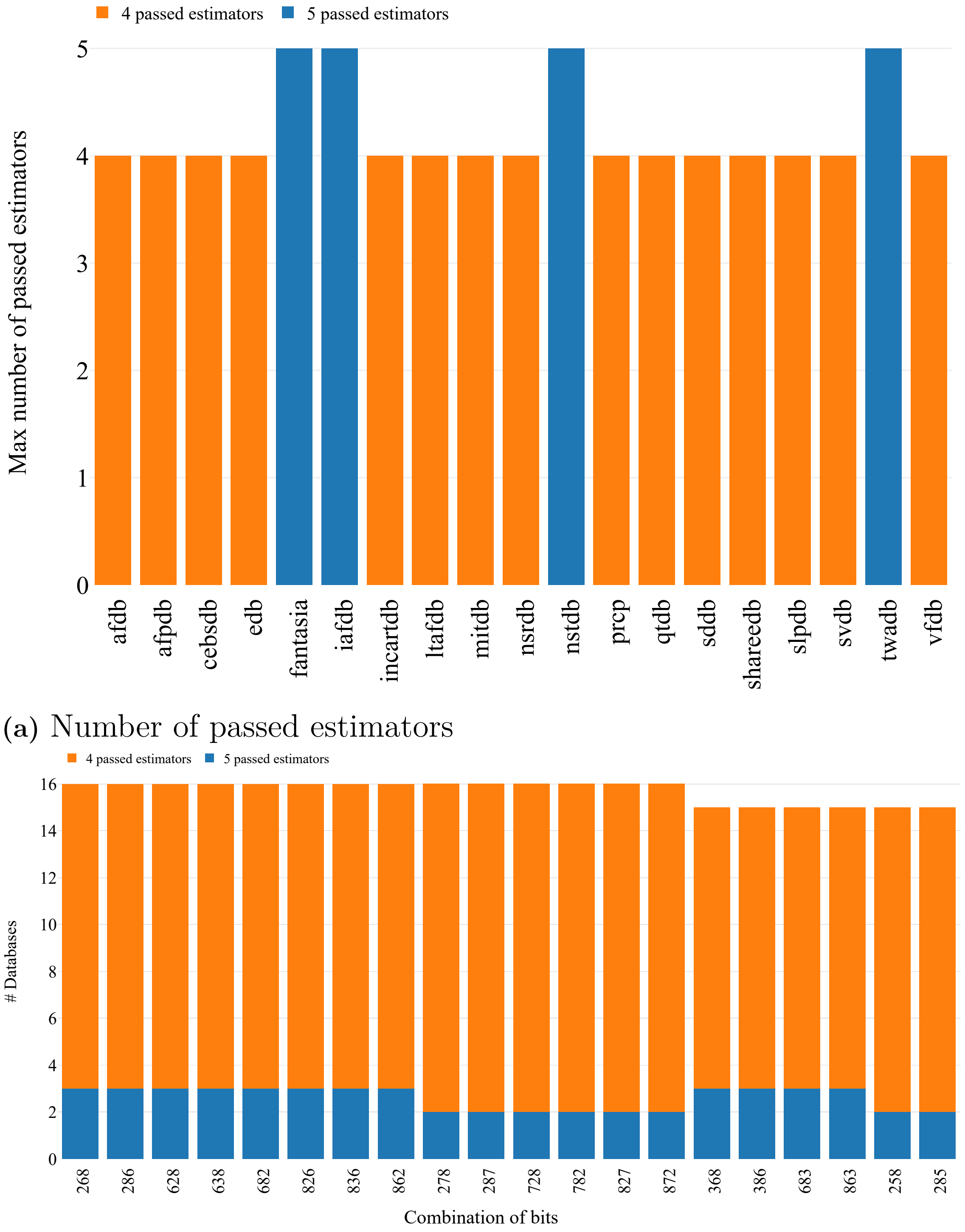
4.3.2 $V_3(8)$ Variations of three bits without repetition

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 Figure 4.4a. In particular, we can assort 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.

Figure 4.4b 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 STS recommendation (see Table 4.2), 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: $\{P_3\{2, 6, 8\}\} \cup \{P_3\{2, 7, 8\}\} \cup \{638, 836\}$, apart from the aforementioned databases, they also have in common nstdb and nsrdb databases (16 out of 19).

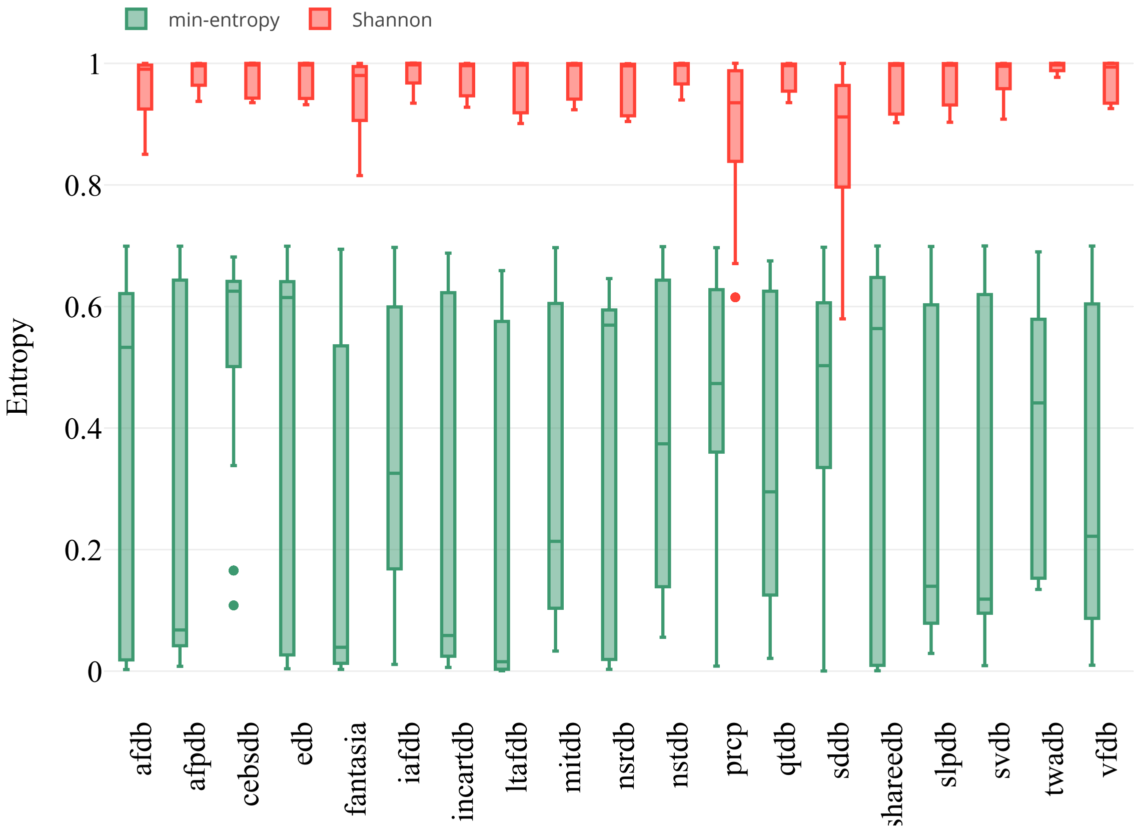
It can be seen in Figure 4.5a a comparison between values obtained from running the Shannon entropy and the min-entropy estimators for all the generated $V_3(8)$ variations and grouped per databases. In this plot, it can be observed how, just by calculating the Shannon



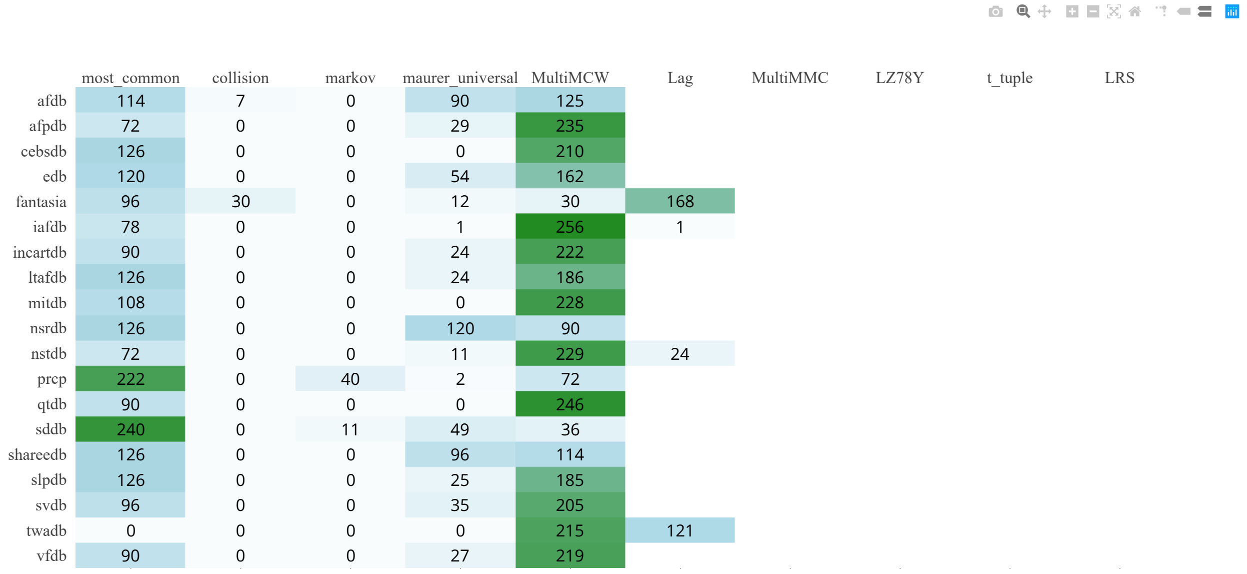
(a) Number of passed estimators

(b) Best combinations

Figure 4.4: Entropy analysis of files generated by extracting 3 bits from IPIs. Figure 4.4a represents the maximum number of passed estimators that achieves at least one combination of bits. Figure 4.4b shows the best and most common combination of bits of databases.



23/3/19 17:18



	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

Figure 4.5: Entropy analysis of files generated by extracting 3 bits from IPIs. Figure 4.5a depicts a comparison of the min-entropy and the Shannon entropy. Figure 4.5b shows a heatmap of the most failed estimators per database.

file:///Users/pablo/Documents/MATLAB/combinatorial/ipi/heatmap_3_bits.html

Página 1 de 1

entropy, cannot be claimed that the heart signal 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.

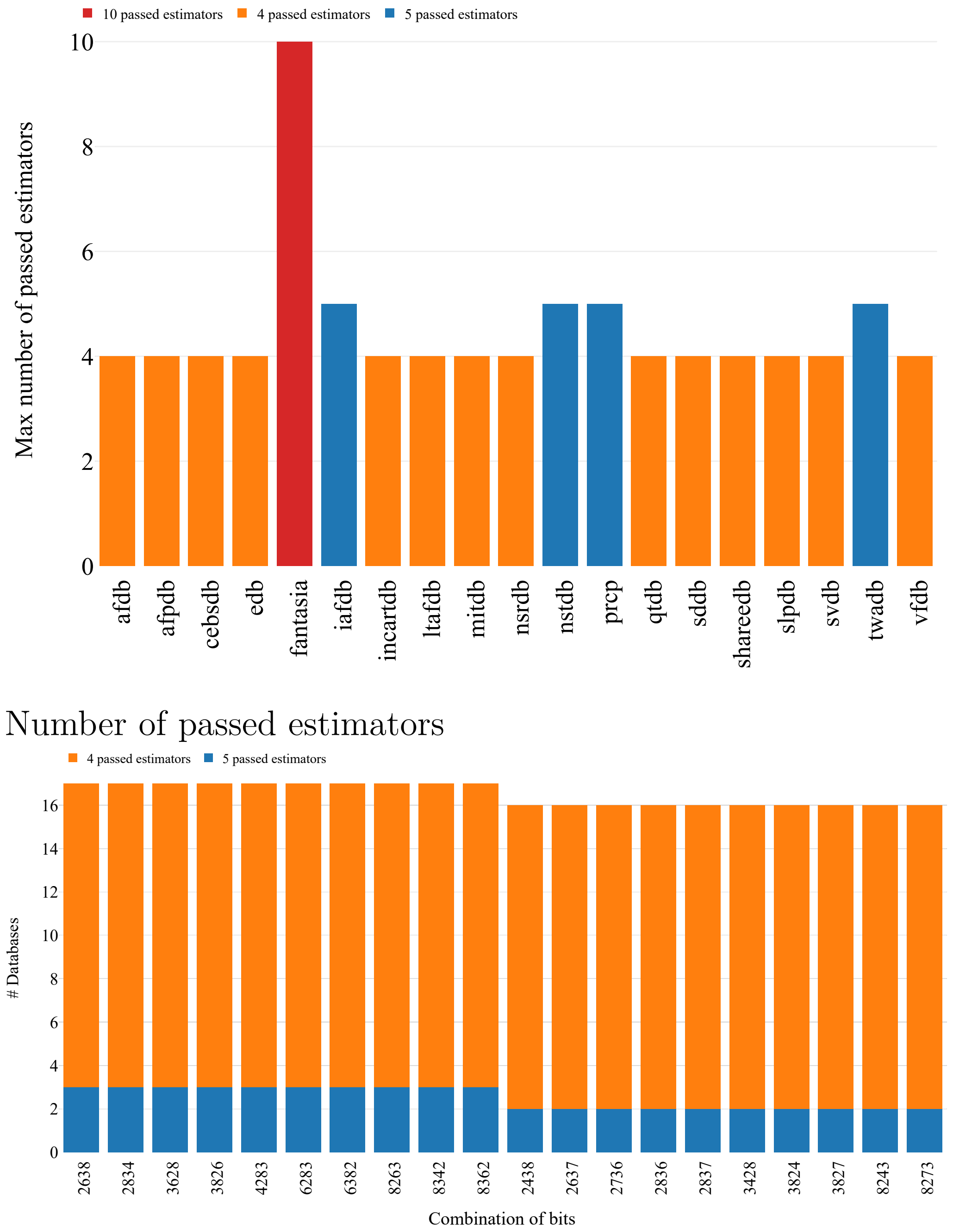
In Figure 4.5b we can see that in general, most of the databases fail in the MultiMCW. Besides, the files of fantasia database usually fail in the Lag estimator—which is an extended and improved version of the MultiMCW estimator. It is also worth mentioning that prcp and sddb databases fail in the first 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.

4.3.3 $V_4(8)$ 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 Figures 4.6 and 4.7.

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 Figure 4.6b, the set of combinations of bits that passes more tests are the following ones: {2638, 2834, 3628, 3826, 4283, 6283, 6382, 8263, 8342, 8362}.

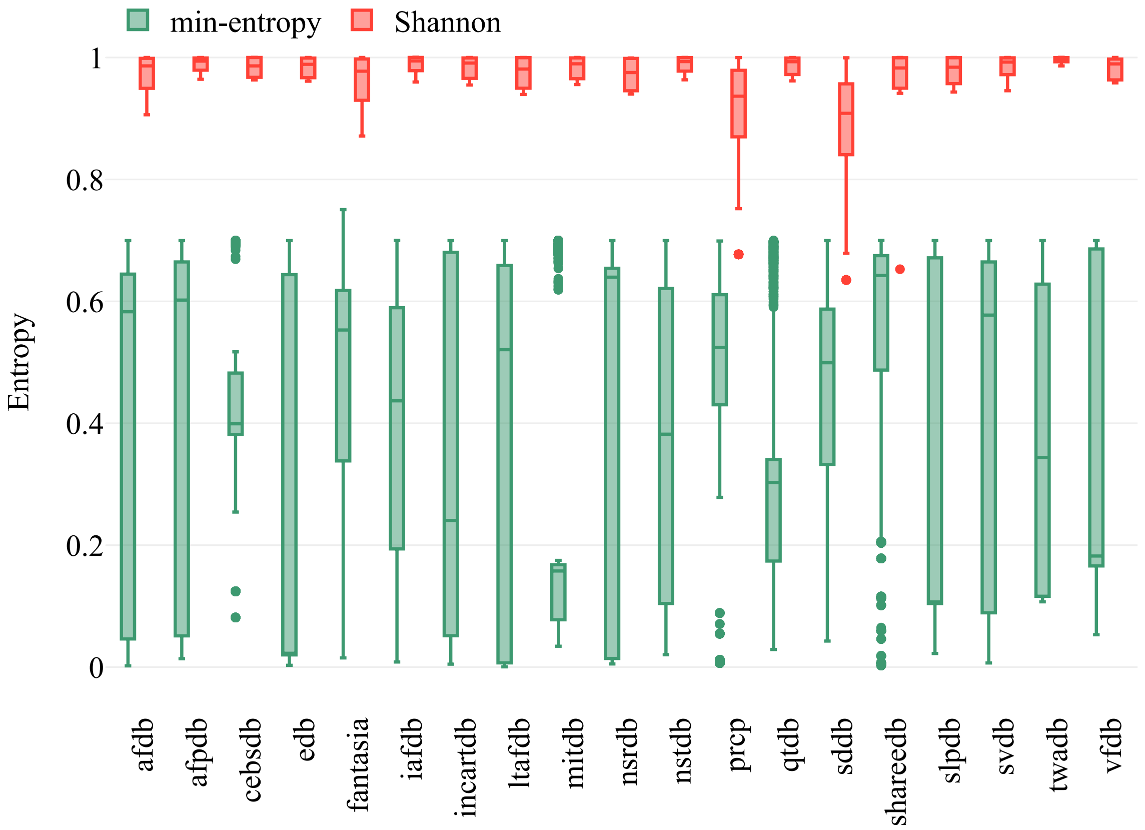
In this case, it is remarkable that none of the databases of the group that passes five estimators (i.e., {iafdb, nstdb, prcp, twadb}), achieves the minimum requirement in terms of size that the NIST STS recommendation establishes (see Table 4.2). Despite that, and given the fact that we found some works in the literature which directly or indirectly use some of these databases for security purposes [97, 194], we decided to keep them in the interpretation of the results.

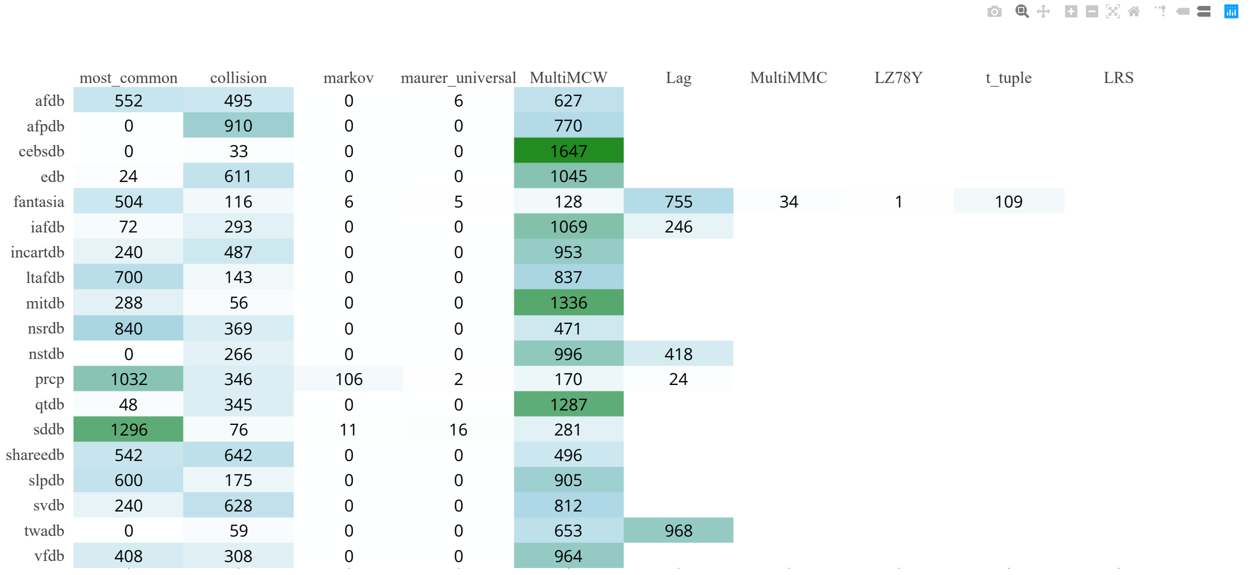


(a) Number of passed estimators

(b) Best combinations.

Figure 4.6: Entropy analysis of files generated by extracting 4 bits from IPIs. Figure 4.6a represents the maximum number of passed estimators that achieves at least one combination of bits. Figure 4.6b shows the best and most common combination of bits of databases.




23/3/19 17:16

	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

Figure 4.7: Entropy analysis of files generated by extracting 4 bits from IPIs. Figure 4.7a depicts a comparison of the min-entropy and the Shannon entropy. Figure 4.7b shows a heatmap of the most failed estimators per database.

file:///Users/pablop/Documents/MATLAB/combinatorial_ipi/heatmap_4_bits.html Página 1 de 1

Figure 4.6b 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 [147].

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 STS recommendation.

In Figure 4.7a, 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, Figure 4.7b 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 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.

4.3.4 $V_5(8)$ Variations of five bits without repetition

The last variation that we computed is $V_5(8)$. In general, it can be seen in Figure 4.8a 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 seconds to generate the same key.

Regarding the best combination of bits (see Figure 4.8b), 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., 26387). Another similar case can be seen for the combination of 36287 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 26387 is the best one possible in any of the tested databases. Additionally and for completeness, we looked for the usually assumed combination of bits 45678 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.

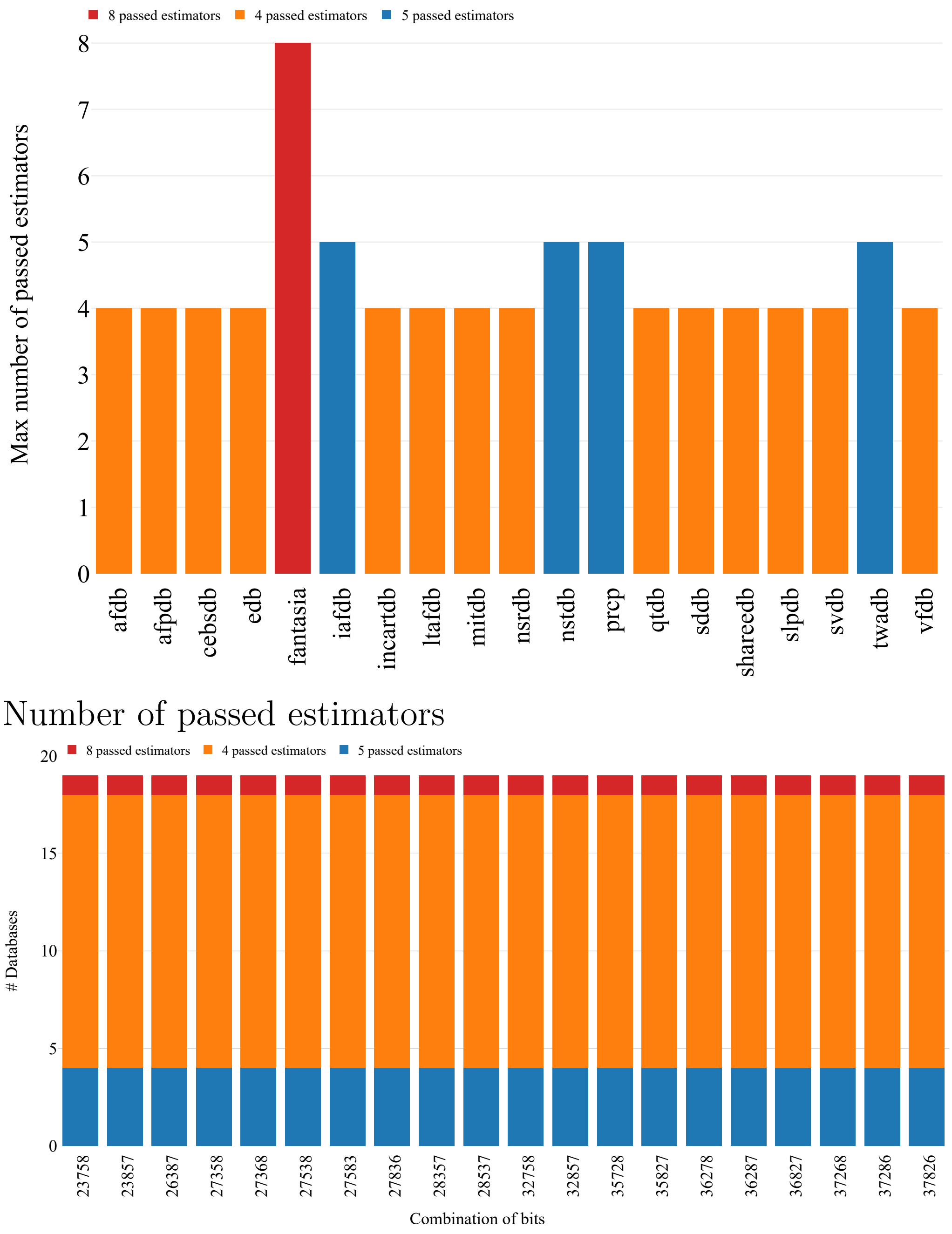
Figure 4.9a shows both Shannon and the min-entropy values. The results follow the same pattern as in the previous experiments in the sense that there is a considerable distance between them. However, we can now see how the min-entropy values of the databases are less spread than in any of the previous 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 Figure 4.9b, 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.

4.3.5 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 [8, 159, 143, 124, 206] as well as the result we obtained after running the `urand` function (see Table 4.3) and we summarize the results in Figure 4.10. 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.

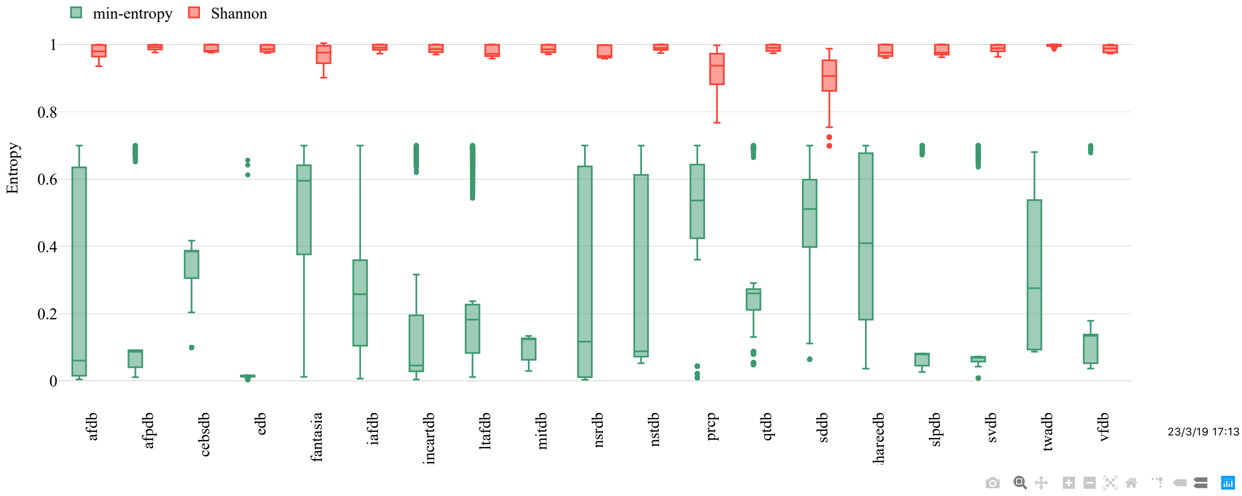
Finally, in Table 4.4, we summarize the conclusions we got from all the experiments executed. It is also interesting to mention that we limited the representation of the results to the top 20. This has direct repercussions in $V_5(8)$, both Figure 4.8b and Table 4.4 where only the first 20 combinations are shown. However, more combinations are not there and achieve the same results.



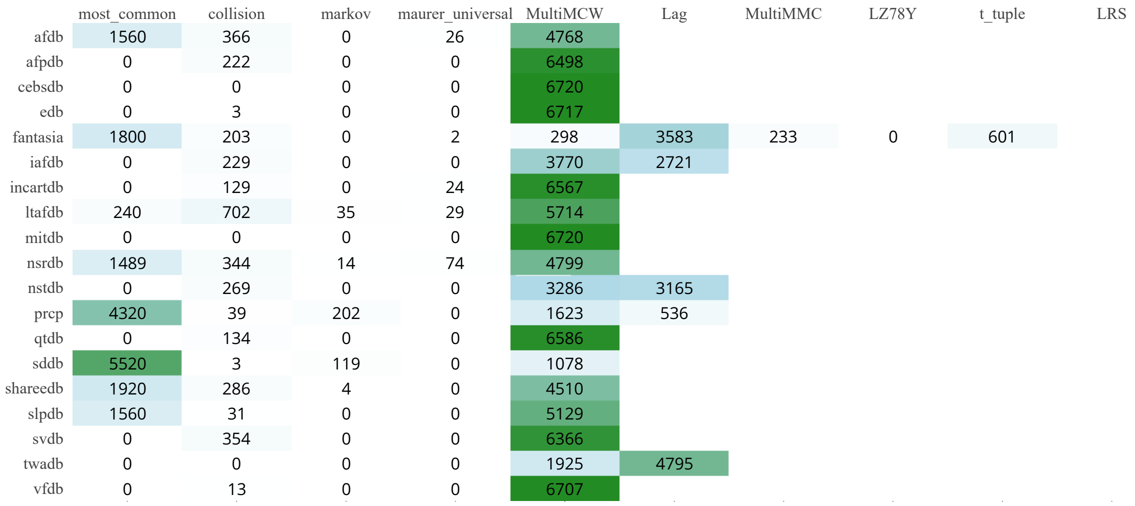
(a) Number of passed estimators

(b) Best combinations.

Figure 4.8: Entropy analysis of files generated by extracting 5 bits from IPIs. Figure 4.8a represents the maximum number of passed estimators that achieves at least one combination of bits. Figure 4.8b shows the best and most common combination of bits of databases.



23/3/19 17:13



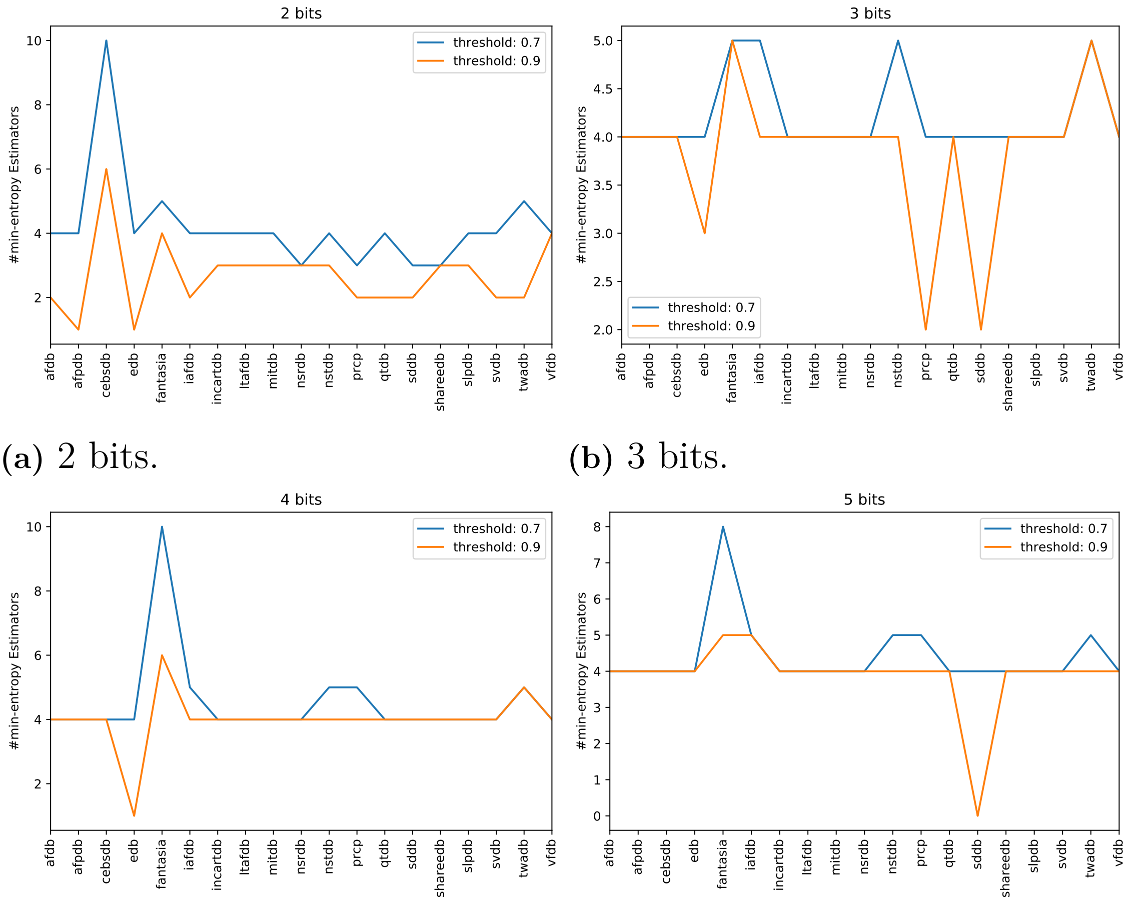
	most_common	collision	markov	maurer_universal	MultiMCW	Lag	MultiMMC	LZ78Y	t_tuple	LRS
afdb	1560	366	0	26	4768					
afpdb	0	222	0	0	6498					
cebsdb	0	0	0	0	6720					
edb	0	3	0	0	6717					
fantasia	1800	203	0	2	298	3583	233	0	601	
iafdb	0	229	0	0	3770	2721				
incartdb	0	129	0	24	6567					
ltafdb	240	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

Figure 4.9: Entropy analysis of files generated by extracting 5 bits from IPIs. Figure 4.9a depicts a comparison of the min-entropy and the Shannon entropy. Figure 4.9b shows a heatmap of the most failed estimators per database.

file:///Users/pablop/Documents/MATLAB/combinatorial_ipi/heatmap_5_bits.html

Página 1 de 1



(a) 2 bits.

(b) 3 bits.

(c) 4 bits.

(d) 5 bits.

Figure 4.10: Min-entropy comparison with thresholds equal to 0.7 and 0.9.

Variations	Best Combinations	Databases
$V_2(8)$	$\{P_2\{7,8\}\}$	11 out of 19
$V_3(8)$	$\{P_3\{2,6,8\}\} \cup \{P_3\{2,7,8\}\} \cup \{638,836\}$	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} ∪ {27368,27538,27583,27836,28357,28537} ∪ {32758,32857,35728,35827,36278,36287}∪ {36827,37268,37286,37826}	19 out of 19

Table 4.4: 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).

4.4 Related Work

Bao et al. [18], Poon et al. [144] and Bao et al. [17] proposed in 2004, 2006 and 2008 respectively, different protocols to secure 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 about how long the sequence is [196, 198] or those which do not use the quantization algorithm proposed by Rostami et al. [147] using wavelets instead [88, 28].

In 2010, Venkatasubramanian and Gupta [186] proposed an IPI-based protocol to secure communications between sensors in a BSN. This protocol was based on Poon et al.'s work [144] and authors did not corroborate the entropy of IPI values. A year later, in 2011 Xu et al. [195] proposed IMDGuard, a security scheme for 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 [20] and stated that the four LSBs are random.

Based on previous works [195, 186, 144], in 2013, Rostami et al. [147] 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 [73], 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. [159] 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

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

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. [8] based on IPIs. Authors used to test their proposal 50 subjects from the MIMIC II Waveform [151], 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. [91] 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 and all the tests provided by AIS.31 [133].

Koya et al. [93] proposed a hybrid mutual authentication and key agreement scheme for 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.

4.5 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 [8, 91, 159, 161, 88], 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.

5

Conclusions

This Thesis analyzes the security and privacy issues in services based on biosignals for implantable medical and wearable devices. In the following, the main conclusions and future work.

5.1 Summary and Conclusions

In Chapter 1 it is shown a summary of the main concepts that this Thesis is based on, i.e., basic concepts of biometrics; the use of heart signals as a biosignal for biometrics; how ECG signal is processed and handled, and; where the data come from to test different proposals. In the final part of this Chapter, the objectives that this dissertation tackles, namely **O1**, **O2**, **O3** and **O4**, are introduced as well as the main motivation for this PhD.

In Chapter 2, we executed two of the most representative randomness suites, i.e., ENT and NIST STS, to the generated IPI values (random number generated by heart signals) of 19 databases downloaded from the Physionet public repository. The analyzed results lead to the following conclusions:

1. IPI values are not as random as were supposed to be.
2. The database that achieves better results is cebsdb, composed of users who do not suffer any medical condition, instead of mitdb which is the most common database used in the literature.
3. A short burst of bits derived from a heart record may seem random, but when large files derived from the heart signal are analyzed, they show that heart signal should not be used for security purposes by using the Rostami et al. methodology [147].
4. An in depth analysis of the random number generation by using heart signals was covered in Chapter 2, addressing thus the first objective (**O1**).

In Chapter 3, we applied an IPI-based token generation approach to obtain equal tokens from the same cardiac signal measured at the same time in two different parts of the body. To get both tokens, it was necessary to process the signals by using a fuzzy extractor algorithm and a time automata algorithm. In total, nineteen databases of the Physionet public repository were used to this experiment. We filtered out and selected those databases with at least two cardiac signals taken from different sensors. Regarding this work, the next conclusions are stated:

1. A pre-processing of the heart signal is mandatory for generating the same token. Using error corrections algorithms or do not solve the synchronization of the signals alone. Because of that, we proposed a run-time monitor, based on a timed automaton, to synchronize both ECG signals.
2. In our experiments, by using both time automata and fuzzy extractors algorithms together, errors are reduced to zero in many of the tested databases.
3. The time needed to generate 32, 64 and 128 bit tokens is increased by 12 to 56 seconds on average to obtain the tokens due to the pre-processing step to synchronize both signals, contrarily to what it is usually assumed (6, 12, and 24 seconds for individual with a heart rate of 80 bpm).
4. A new system to generate cryptographics keys coming from different sensors measuring the same heart signal has been proposed in Chapter 3, addressing thus the second and the third objectives (**O2**, **O3**) of this dissertation.

Finally, in Chapter 4 we carried out an in-depth study, analyzing the entropy of the heart signal. In particular, we evaluated the IPI values extracted from a heart signal according to the NIST STS recommendation. This is the first work that aims at extracting the best combination of bits of the IPIs in terms of min-entropy. The results obtained from this work can be summarized as:

1. We demonstrated that the four LSBs are not the best bits to be used in cryptographic applications.
2. We empirically analyzed more than 160,000 files and proposed different combinations for generating tokens of 2, 3, 4 and 5 bits length which are, in general, much better than taking the LSBs.
3. We analyzed the entropy of the LSBs values extracted from a heart signal addressing thus the fourth objective (**O4**) of this dissertation.

As a global conclusion, one of the difficulties encountered was that many of the previous proposals in this field did not provide enough information about either the databases authors used for testing or if the dataset is private, or even the methodology used to generated the IPIs. Therefore, in order to advance in this line of research, it would be a good starting point to either use public repositories such as Physionet or make the datasets public.

5.2 Future Work

There are two main decisions we took which might be improved in the future.

Extend to other biometrical signals. We have focused on heart signals by using two channels (ECG_1 and ECG_2). We plan to extend our analysis and proposal to other physiological signals like Photoplethysmograms (PPGs), Blood Pressure (BP) or even using the Electroencephalograms (EEGs), as proposed in [13].

Timed Automaton. The mean value has been used as the upper-bound time for RR intervals as it has been previously proposed in medial research [175], and the consequence is that certain peaks are missed and the run-time monitor does not synchronize as many peaks as it might do. We plan to further research on this line and refine our timed automaton to achieve better results, specially in those databases that do not perform as well.

Bibliography

[1] J. Ackleson. Border security in risk society. *Journal of Borderlands Studies*, 20(1):1–22, 2005.

[2] P. Adey. Facing airport security: affect, biopolitics, and the preemptive securitisation of the mobile body. *Environment and Planning D: Society and Space*, 27(2):274–295, 2009.

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

[4] M. Akhbari, M. B. Shamsollahi, O. Sayadi, A. A. Armoundas, and C. Jutten. Ecg segmentation and fiducial point extraction using multi hidden markov model. *Computers in Biology and Medicine*, 79:21 – 29, 2016.

[5] I. Al-Aweel, K. Krishnamurthy, J. Hausdorff, J. Mietus, J. Ives, A. Blum, D. Schomer, and A. Goldberger. Postictal heart rate oscillations in partial epilepsy. *Neurology*, 53(7):1590–1590, 1999.

[6] P. Albrecht. *ST segment characterization for long term automated ECG analysis.* Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1983.

[7] R. Altawy and A. M. Youssef. Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices. *IEEE Access*, 4:959–979, 2016.

[8] D. K. Altop, A. Levi, and V. Tuzcu. Deriving cryptographic keys from physiological signals. *Pervasive and Mobile Computing*, 2016.

[9] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[10] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In *REX Workshop*, volume 600, pages 74–106, 1991.

[11] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[12] M. Ambigavathi and D. Sridharan. Energy efficient and load balanced priority queue algorithm for wireless body area network. *Future Generation Computer Systems*, 88:586 – 593, 2018.

[13] P. Bagade, A. Banerjee, J. Milazzo, and S. K. S. Gupta. Protect your bsn: No handshakes, just namaste! In *BSN*, pages 1–6, 2013.

[14] T. Bai, J. Lin, G. Li, H. Wang, P. Ran, Z. Li, D. Li, Y. Pang, W. Wu, and G. Jeon. A lightweight method of data encryption in bans using electrocardiogram signal. *Future Generation Computer Systems*, 92:800 – 811, 2019.

[15] S. Banerjee, R. Gupta, and M. Mitra. Delineation of ecg characteristic features using multiresolution wavelet analysis method. *Measurement*, 45(3):474 – 487, 2012.

[16] S. Bao, Z. He, R. Jin, and P. An. A compensation method to improve the performance of ipi-based entity recognition system in body sensor networks. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1250–1253, July 2013.

[17] S.-D. Bao, C. C. Poon, Y.-T. Zhang, and L.-F. Shen. Using the timing information of heartbeats as an entity identifier to secure body sensor network. *Trans. Info. Tech. Biomed.*, 12(6):772–779, 2008.

[18] S.-D. Bao, L.-F. Shen, and Y.-T. Zhang. A novel key distribution of body area networks for telemedicine. In *Workshop on Biomedical Circuits and Systems*, pages 1–17, 2004.

[19] E. Barker and J. Kelsey. Recommendation for the entropy sources used for random bit generation. *Draft NIST Special Publication*, pages 800–900, 2012.

[20] 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, and S. Vo. Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, National Institute of Standards & Technology, 2010.

[21] G. Behrmann, A. David, K. G. Larsen, J. Hakansson, P. Petterson, W. Yi, and M. Hendriks. Uppaal 4.0. In *International*

Conference on the Quantitative Evaluation of Systems, pages 125–126, 2006.

[22] T. Belkhouja, X. Du, A. Mohamed, A. K. Al-Ali, and M. Guizani. Biometric-based authentication scheme for implantable medical devices during emergency situations. *Future Generation Computer Systems*, 98:109 – 119, 2019.

[23] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.*, 5(4):367–397, 2002.

[24] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik/Biomedical Engineering*, 40(s1):317–318, 1995.

[25] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In *Computer Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 546–550, 1998.

[26] A. Calleja, P. Peris-Lopez, and J. E. Tapiador. Electrical heart signals can be monitored from the moon: Security implications for ipi-based protocols. In R. N. Akram and S. Jajodia, editors, *Information Security Theory and Practice*, pages 36–51, 2015.

[27] C. Camara, P. Peris-Lopez, L. Gonzalez-Manzano, and J. Tapiador. Real-time electrocardiogram streams for continuous authentication. *Applied Soft Computing*, 68:784 – 794, 2018.

[28] C. Camara, P. Peris-Lopez, H. Martín, and M. Aldalaien. Ecg-rng: A random number generator based on ecg signals and suitable for securing wireless sensor networks. *Sensors*, 18(9), 2018.

[29] C. Camara, P. Peris-Lopez, and J. E. Tapiador. Human identification using compressed ecg signals. *Journal of Medical Systems*, 39(11):148, Sep 2015.

[30] E. Camlikaya, A. Kholmatov, and B. Yanikoglu. Multi-biometric templates using fingerprint and voice. In *Biometric technology for human identification V*, volume 6944, page 69440I. International Society for Optics and Photonics, 2008.

[31] A. Carman. Withings' new smartwatch has an ekg sensor to compete with the apple watch, 2019.

[32] C. Carreiras, A. Lourenço, A. Fred, and R. Ferreira. Ecg signals for biometric applications – are we there yet? In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*, pages 765–772. SciTePress, 2014.

[33] S. A. Chaudhry, K. Mahmood, H. Naqvi, and M. K. Khan. An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. *Journal of Medical Systems*, 39(11):1–12, 2015.

[34] G. Chen. Are electroencephalogram (eeg) signals pseudorandom number generators? *Journal of Computational and Applied Mathematics*, 268:1 – 4, 2014.

[35] X. Chen, Y. Zhang, G. Zhang, and Y. Zhang. Evaluation of ecg random number generator for wireless body sensor networks security. In *BMEI*, pages 1308–1311, 2012.

[36] S. Cherukuri, K. K. Venkatasubramanian, and S. K. S. Gupta. Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In *Conference on Parallel Processing Workshops*, pages 432–439, 2003.

[37] J.-C. Chien, J.-P. Wang, C.-L. Cho, and F.-C. Chong. Security biosignal transmission based on face recognition for telemedicine. *Biomedical Engineering: Applications, Basis and Communications*, 19(01):63–69, 2007.

[38] H. Chizari and E. Lupu. Extracting randomness from the trend of IPI for cryptographic operators in implantable medical devices. *CoRR*, abs/1806.10984, 2018.

[39] H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang. Outsourceable two-party privacy-preserving biometric authentication. In *CCS*, pages 401–412, 2014.

[40] R. Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology & People*, 7(4):6–37, 1994.

[41] K. Côté-Boucher. The diffuse border: intelligence-sharing, control and confinement along canada's smart border. *Surveillance and Society*, 2008.

[42] J.-P. Couderc. The telemetric and holter ecg warehouse initiative (THEW): a data repository for the design, implementation and validation of ecg-related technologies. In *IEEE Engineering in Medicine and Biology*, pages 6252–6255. IEEE, 2010.

[43] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, Sep. 1990.

[44] T. Denning, A. Borning, B. Friedman, B. T. Gill, T. Kohno,

and W. H. Maisel. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *CHI*, pages 917–926. ACM, 2010.

[45] L. M. Dinca and G. Hancke. User-centric key entropy: Study of biometric key derivation subject to spoofing attacks. *Entropy*, 19(2), 2017.

[46] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, pages 523–540, 2004.

[47] R. Doroz and P. Porwik. Handwritten signature recognition with adaptive selection of behavioral features. In *Computer Information Systems–Analysis and Technologies*, pages 128–136. Springer, 2011.

[48] N. Duta. A survey of biometric technology based on hand shape. *Pattern Recognition*, 42(11):2797 – 2806, 2009.

[49] A. Eng and L. A. Wahsheh. Look into my eyes: A survey of biometric security. In S. Latifi, editor, *ITNG*, pages 422–427, 2013.

[50] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *PETS*, pages 235–253, 2009.

[51] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, and A. Schclar. User identity verification via mouse dynamics. *Information Sciences*, 201:19–36, 2012.

[52] P. Flandrin, G. Rilling, and P. Goncalves. Empirical mode decomposition as a filter bank. *IEEE Signal Processing Letters*, 11(2):112–114, Feb 2004.

[53] B. Ganeshan, D. Theckedath, R. Young, and C. Chatwin. Biometric iris recognition system using a fast and robust iris localization and alignment procedure. *Optics and Lasers in Engineering*, 44(1):1 – 24, 2006.

[54] M. A. García-González, A. Argelagós-Palau, M. Fernández-Chimeno, and J. Ramos-Castro. A comparison of heartbeat detectors for the seismocardiogram. In *CinC*, pages 461–464, 2013.

[55] M. GB and M. RG. A new method for detecting atrial fibrillation using r-r intervals. *Computers in Cardiology*, 1983.

[56] I. Gerhardts. Ilja gerhardt - random number tests, 2017.

[57] 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, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[58] L. González-Manzano, J. M. de Fuentes, P. Peris-Lopez, and C. Camara. Encryption by heart (ebh)—using ecg for time-invariant symmetric key generation. *Future Generation Computer Systems*, 77:136 – 148, 2017.

[59] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*, pages 89–98, 2006.

[60] S. Graja and J. . Boucher. Hidden markov tree model applied to ecg delineation. *IEEE Transactions on Instrumentation and Mbeasurement*, 54(6):2163–2168, Dec 2005.

[61] S. D. Greenwald. *The development and analysis of a ventricular fibrillation detector*. PhD thesis, Massachusetts Institute of Technology, 1986.

[62] S. D. Greenwald. *The development and analysis of a ventricular fibrillation detector*. PhD thesis, Massachusetts Institute of Technology, 1986.

[63] S. D. Greenwald, R. S. Patil, and R. G. Mark. Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information. In *Computers in Cardiology*, pages 461–464, 1990.

[64] Z. Guo, Y. Xin, and Y. Zhao. Cancer classification using entropy analysis in fractional fourier domain of gene expression profile. *Biotechnology & Biotechnological Equipment*, 0(0):1–5, 2017.

[65] P. Hagerty and T. Draper. Entropy bounds and statistical tests. In *NIST Random Bit Generation Workshop*, pages 1319–1327, 2012.

[66] H. Hamidi. An approach to develop the smart health using internet of things and authentication based on biometric technology. *Future Generation Computer Systems*, 91:434 – 449, 2019.

[67] Y. S. Han. BCH codes. In *Graduate Institute of Communication Engineering, National Taipei University*, 2016.

[68] M. A. Hanson, H. C. Powell Jr., A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach. Body area sensor networks: Challenges and opportunities. *Computer*, 42(1):58–65, Jan. 2009.

[69] T. Heldt, M. B. Oefinger, M. Hoshiyama, and R. G. Mark. Circulatory response to passive and active changes in posture. In *Computers in Cardiology, 2003*, pages 263–266, 2003.

[70] C. Herder, L. Ren, M. van Dijk, M. D. Yu, and S. Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing*, 14(1):65–82, 2017.

[71] T. Hong, S. D. Bao, Y. T. Zhang, Y. Li, and P. Yang. An improved scheme of ipi-based entity identifier generation for securing body sensor networks. In *EMBS*, pages 1519–1522, 2011.

[72] S. S. Hosseini and S. Mohammadi. Review banking on biometric in the world's banks and introducing a biometric model for iran's banking system. *Journal of Basic and Applied Scientific Research*, 2(9):9152–9160, 2012.

[73] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen. Opfka: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks. In *INFOCOM*, pages 2274–2282, 2013.

[74] Y. Ichimaru and G. Moody. Development of the polysomnographic database on cd-rom. *Psychiatry and Clinical Neurosciences*, 53(2):175–177, 1999.

[75] G. Inc. Google scholar, 2018. Accessed: 2018-04-30.

[76] A. N. S. Institute, A. for the Advancement of Medical Instrumentation, et al. *Cardiac Monitors, Heart Rate Meters and Alarms*. Association for the Advancement of Medical Instrumentation, 2002.

[77] E. Institute. Aha database sample excluded record, 2018.

[78] A. M. Intelligence. The Global Biometrics and Mobility Report: The Convergence of Commerce and Privacy . Technical report, Acuity Market Intelligence, 2015.

[79] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[80] N. Iyengar, C. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz. Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 271(4):R1078–R1084, 1996.

[81] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008:113:1–113:17, 2008.

[82] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

[83] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 476–482. ACM, 2011.

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

[85] A. D. Jurik and A. C. Weaver. Securing mobile devices with biotelemetry. In *ICCCN*, pages 1–6, 2011.

[86] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura. Performance analysis for puf data using fuzzy extractor. In Y.-S. Jeong, Y.-H. Park, C.-H. R. Hsu, and J. J. J. H. Park, editors, *CUTE*, pages 277–284, 2014.

[87] M. Karnan, M. Akila, and N. Krishnaraj. Biometric personal authentication using keystroke dynamics: A review. *Applied Soft Computing*, 11(2):1565 – 1573, 2011. The Impact of Soft Computing for the Progress of Artificial Intelligence.

[88] M. V. Karthikeyan and J. M. L. Manickam. Ecg-signal based secret key generation (eskg) scheme for wban and hardware implementation. *Wireless Personal Communications*, Sep 2018.

[89] A. Kharb, V. Saini, Y. Jain, and S. Dhiman. A review of gait cycle and its parameters. *IJCEM International Journal of Computational Engineering & Management*, 13:78–83, 2011.

[90] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *DSN*, pages 125–134, 2009.

[91] J. Kim, K. Cho, Y.-K. Kim, K.-S. Lim, and S. U. Shin. Study on peak misdetection recovery of key exchange protocol using heartbeat. *The Journal of Supercomputing*, Sep 2018.

[92] B. . Kohler, C. Hennig, and R. Orglmeister. The principles of software qrs detection. *IEEE Engineering in Medicine and Biology Magazine*, 21(1):42–57, Jan 2002.

[93] A. M. Koya and D. P. P. Anonymous hybrid mutual authentica-

tion and key agreement scheme for wireless body area network. *Computer Networks*, 140:138 – 151, 2018.

[94] M. Kumar, R. B. Pachori, and U. R. Acharya. Automated diagnosis of myocardial infarction ecg signals using sample entropy in flexible analytic wavelet transform framework. *Entropy*, 19(488), 2017.

[95] S. Kumar and S. K. Singh. Monitoring of pet animal in smart cities using animal biometrics. *Future Generation Computer Systems*, 83:553 – 563, 2018.

[96] P. Laguna, R. Mark, A. Goldberg, and G. Moody. Database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In *Computers in Cardiology*, volume 1997, pages 673 – 676, 10 1997.

[97] R. Lazzeretti, J. Guajardo, and M. Barni. Privacy preserving ECG quality evaluation. In *MM & Sec*, pages 165–174, 2012.

[98] R. Lennox, M. L. Dennis, C. K. Scott, and R. Funk. Combining psychometric and biometric measures of substance use. *Drug and Alcohol Dependence*, 83(2):95–103, 2006.

[99] M. Li, S. Yu, J. D. Guttman, W. Lou, and K. Ren. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Trans. Sen. Netw.*, 9(2):18:1–18:35, 2013.

[100] N. Li, F. Guo, Y. Mu, W. Susilo, and S. Nepal. Fuzzy extractors for biometric identification. In *ICDCS*, pages 667–677, 2017.

[101] T. Li and M. Zhou. Ecg classification using wavelet packet entropy and random forests. *Entropy*, 18(8):285, 2016.

[102] X. Li, J. Liu, Q. Yao, and J. Ma. Efficient and consistent key extraction based on received signal strength for vehicular ad hoc networks. *IEEE Access*, 5:5281–5291, 2017.

[103] C. Lin, C. Mailhes, and J. Tourneret. P- and t-wave delineation in ecg signals using a bayesian approach and a partially collapsed gibbs sampler. *IEEE Transactions on Biomedical Engineering*, 57(12):2840–2849, Dec 2010.

[104] W.-H. Lin, D. Wu, C. Li, H. Zhang, and Y.-T. Zhang. Comparison of heart rate variability from ppg with that from ecg. In *ICHI*, pages 213–215, 2014.

[105] S. Liu and M. Silverman. A practical guide to biometric security technology. *IT Professional*, 3(1):27–32, 2001.

[106] H. Löhr, A.-R. Sadeghi, and M. Winandy. Securing the e-health

cloud. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 220–229, 2010.

[107] M. Lucchini, N. Pini, W. P. Fifer, N. Burtchen, and M. G. Signorini. Entropy information of cardiorespiratory dynamics in neonates during sleep. *Entropy*, 19(225), 2017.

[108] J. P. Madeiro, W. B. Nicolson, P. C. Cortez, J. A. Marques, C. R. Vázquez-Seisdedos, N. Elangovan, G. A. Ng, and F. S. Schlindwein. New approach for t-wave peak detection and t-wave end location in 12-lead paced ecg signals based on a mathematical model. *Medical Engineering & Physics*, 35(8):1105 – 1115, 2013.

[109] M. I. Malik, S. Ahmed, A. Dengel, and M. Liwicki. A signature verification framework for digital pen applications. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 419–423. IEEE, 2012.

[110] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, 2nd edition, 2009.

[111] E. Marasco and A. Ross. A survey on antispoofing schemes for fingerprint recognition systems. *ACM Comput. Surv.*, 47(2):28:1–28:36, 2014.

[112] A. Martínez, R. Alcaraz, and J. J Rieta. Automatic electrocardiogram delineator based on the phasor transform of single lead recordings. In *Computing in Cardiology*, pages 987–990, Sep. 2010.

[113] J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna. A wavelet-based ECG delineator: evaluation on standard databases. *IEEE Transactions on Biomedical Engineering*, 51(4):570–581, April 2004.

[114] U. M. Maurer. A universal statistical test for random bit generators. *Journal of cryptology*, 5(2):89–105, 1992.

[115] S. Mehta, D. Shete, N. Lingayat, and V. Chouhan. K-means algorithm for the detection and delineation of qrs-complexes in electrocardiogram. *IRBM*, 31(1):48 – 54, 2010.

[116] P. Melillo, R. Izzo, A. Orrico, P. Scala, M. Attanasio, M. Mirra, N. De Luca, and L. Pecchia. Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis. *PloS one*, 10(3):e0118504, 2015.

[117] F. Miao, L. Jiang, Y. Li, and Y. T. Zhang. Biometrics based

novel key distribution solution for body sensor networks. In *EMBC*, pages 2458–2461, 2009.

[118] F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *CCS*, pages 48–56, 1997.

[119] G. Moody, A. Goldberger, S. McClennen, and S. Swiryn. Predicting the onset of paroxysmal atrial fibrillation: the computers in cardiology challenge 2001. In *Computers in Cardiology*, pages 113–116, 2001.

[120] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45–50, 2001.

[121] G. B. Moody, R. G. Mark, and A. L. Goldberger. Evaluation of thetrim'ecg data compressor. In *Computers in Cardiology. Proceedings.*, pages 167–170, 1988.

[122] G. B. Moody, W. Muldrow, and R. G. Mark. A noise stress test for arrhythmia detectors. *Computers in cardiology*, 11(3):381–384, 1984.

[123] T. P. . C. i. C. C. Moody GB. T-wave alternans. computers in cardiology. *Neurology*, 35:505–508, 2008.

[124] S. R. Moosavi, E. Nigussie, S. Virtanen, and J. Isoaho. Cryptographic key generation using ecg signal. In *CCNC*, pages 1024–1031, 2017.

[125] V. S. Murthy, S. Ramamoorthy, N. Srinivasan, S. Rajagopal, and M. M. Rao. Analysis of photoplethysmographic signals of cardiovascular patients. In *EMBS*, volume 3, pages 2204–2207, Oct 2001.

[126] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood. Protection of privacy in biometric data. *IEEE Access*, 4:880–892, 2016.

[127] M. Nieto, K. Johnston-Dodds, and C. W. Simmons. *Public and private applications of video surveillance and biometric technologies*, volume 2. California State Library, California Research Bureau Sacramento, CA, 2002.

[128] F. Nolle, F. Badura, J. Catlett, R. Bowser, and M. H. Sketch. CREI-GARD, a new concept in computerized arrhythmia monitoring systems. In *Computers in Cardiology*, 1986.

[129] D. Oberoi, W. Y. Sou, Y. Y. Lui, R. Fisher, L. Dinca, and G. P. Hancke. Wearable security: Key derivation for body area sensor

networks based on host movement. In *ISIE*, pages 1116–1121, 2016.

[130] L. Ortiz-Martin, P. Picazo-Sanchez, P. Peris-Lopez, and J. Tapiador. Heartbeats do not make good pseudo-random number generators: An analysis of the randomness of inter-pulse intervals. *Entropy*, 20(2), 2018.

[131] L. Ortiz-Martin, P. Picazo-Sanchez, P. Peris-Lopez, J. Tapiador, and G. Schneider. Feasibility analysis of inter-pulse intervals based solutions for cryptographic token generation by two electrocardiogram sensors. *Future Generation Computer Systems*, 96:283 – 296, 2019.

[132] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985.

[133] H. Park, J.-S. Kang, and Y. Yeom. Probabilistic analysis of AIS. 31 statistical tests for TRNGs and their applications to security evaluations. *Journal of the Korea Institute of Information Security and Cryptology*, 26(1):49–67, 2016.

[134] M. Patel and J. Wang. Applications, challenges, and prospective in emerging body area networking technologies. *Wireless Commun.*, 17(1):80–88, 2010.

[135] T. Penzel, G. B. Moody, R. G. Mark, A. L. Goldberger, and J. H. Peter. The apnea-ecg database. In *Computers in cardiology*, pages 255–258, 2000.

[136] B. Petchlert and H. Hasegawa. Using a low-cost electroencephalogram (eeg) directly as random number generator. In *IIAIAAI*, pages 470–474, 2014.

[137] S. Peter, B. Pratap Reddy, F. Momtaz, and T. Givargis. Design of secure ecg-based biometric authentication in body area sensor networks. *Sensors*, 16(4), 2016.

[138] S. Petrutiu, A. V. Sahakian, and S. Swiryn. Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans. *Europace*, 9(7):466–470, 2007.

[139] Physionet. Intracardiac atrial fibrillation database, 2018.

[140] Physionet. St.-petersburg institute of cardiological technics 12-lead arrhythmia database, 2018.

[141] Physionet. The MIT-BIH Normal Sinus Rhythm Database, 2018.

[142] S. Pirbhulal, O. W. Samuel, W. Wu, A. K. Sangaiah, and G. Li.

A joint resource-aware and medical data security framework for wearable healthcare systems. *Future Generation Computer Systems*, 95:382 – 391, 2019.

[143] S. Pirbhulal, H. Zhang, W. Wu, S. C. Mukhopadhyay, and Y. Zhang. Heart-beats based biometric random binary sequences generation to secure wireless body sensor networks. *IEEE Transactions on Biomedical Engineering*, pages 1–1, 2018.

[144] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine*, 44(4):73–81, 2006.

[145] K. B. Rasmussen, M. Roeschlin, I. Martinovic, and G. Tsudik. Authentication using pulse-response biometrics. In *NDSS*, 2014.

[146] F. Rincón, J. Recas, N. Khaled, and D. Atienza. Development and evaluation of multilead wavelet-based ecg delineation algorithms for embedded wireless sensor nodes. *IEEE Transactions on Information Technology in Biomedicine*, 15(6):854–863, Nov 2011.

[147] M. Rostami, A. Juels, and F. Koushanfar. Heart-to-heart (H2H): authentication for implanted medical devices. In *CCS*, pages 1099–1112, 2013.

[148] R. E. Rowe. Casino gambling system with biometric access control, Oct. 24 2006. US Patent 7,125,335.

[149] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson. SoK: Security and privacy in implantable medical devices and body area networks. In *Security & Privacy*, pages 524–539, May 2014.

[150] 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*, pages 547–561, Berkeley, Calif., 1961. University of California Press.

[151] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[152] I. Saini, D. Singh, and A. Khosla. Delineation of ecg wave components using k-nearest neighbor (knn) algorithm: Ecg wave de-

lineation using knn. In *Conference on Information Technology: New Generations*, pages 712–717, April 2013.

[153] D. Salomon. *Data compression: the complete reference.* Springer Science & Business Media, 2004.

[154] A. Sammoud, O. Hamdi, M. A. Chalouf, and A. Bouallegue. A new protocol for an efficient and green biometric-based security key establishment in wban's. In *IWCMC*, pages 762–767, 2018.

[155] A. Schaller, T. Stanko, B. Škorić, and S. Katzenbeisser. Eliminating leakage in reverse fuzzy extractors. *IEEE Transactions on Information Forensics and Security*, 13(4):954–964, 2018.

[156] B. Secure. Ecg biometrics for the connected car (white paper), 2018.

[157] R. M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, and C. I. De Zeeuw. Peak misdetection in heart-beat-based security: Characterization and tolerance. In *EMBC*, pages 5401–5405, 2014.

[158] R. M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, and C. I. D. Zeeuw. Peak misdetection in heart-beat-based security: Characterization and tolerance. In *EMBC*, pages 5401–5405, 2014.

[159] R. M. Seepers, C. Strydis, I. Sourdis, and C. D. Zeeuw. Enhancing heart-beat-based security for mhealth applications. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2015.

[160] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. D. Zeeuw. On using a von neumann extractor in heart-beat-based security. In *Trustcom/BigDataSE/ISPA*, volume 1, pages 491–498, 2015.

[161] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. D. Zeeuw. On using a von neumann extractor in heart-beat-based security. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 491–498, 2015.

[162] R. M. Seepers, C. Strydis, I. Sourdis, and C. I. D. Zeeuw. Enhancing heart-beat-based security for mhealth applications. *IEEE Journal of Biomedical and Health Informatics*, 21(1):254–262, 2017.

[163] R. M. Seepers, J. H. Weber, Z. Erkin, I. Sourdis, and C. Strydis. Secure key-exchange protocol for implants using heartbeats. In *Conference on Computing Frontiers*, pages 119–126, 2016.

[164] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[165] C. Shen, Z. Cai, X. Guan, Y. Du, and R. A. Maxion. User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security*, 8(1):16–30, 2013.

[166] B. Shi, Y. Zhang, C. Yuan, S. Wang, and P. Li. Entropy analysis of short-term heartbeat interval time series during regular walking. *Entropy*, 19(10), 2017.

[167] K. A. Sidek, I. Khalil, and H. F. Jelinek. Ecg biometric with abnormal cardiac conditions in remote monitoring system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11):1498–1509, 2014.

[168] M. Sýs and Z. Říha. Faster randomness testing with the nist statistical test suite. In *SPACE*, pages 272–284. Springer International Publishing, 2014.

[169] J. Szczepanski, E. Wajnryb, J. Amigó, M. V. Sanchez-Vives, and M. Slater. Biometric random number generators. *Computers & Security*, 23(1):77 – 84, 2004.

[170] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. Moody, C. Zeelenberg, and C. Marchesi. The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography. *European heart journal*, 13(9):1164–1172, 1992.

[171] P. S. Teh, A. B. J. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013, 2013.

[172] X. F. Teng and Y. T. Zhang. Study on the peak interval variability of photoplethysmogtaphic signals. In *EMBS*, pages 140–141, Oct 2003.

[173] H. M. Thang, V. Q. Viet, N. D. Thuc, and D. Choi. Gait identification using accelerometer on mobile phone. In *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 344–348. IEEE, 2012.

[174] C. Torrence and G. P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1):61–78, 1998.

[175] 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.

[176] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish,

and M. Boyle. Recommendation for the entropy sources used for random bit generation. *NIST Special Publication*, 2018.

[177] P. Tuyls, B. Skoric, and T. Kevenaar. *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting.* Springer-Verlag, 2007.

[178] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, and K. S. Kwak. A comprehensive survey of wireless body area networks. *Journal of medical systems*, 36(3):1065–1094, 2012.

[179] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[180] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar. Blind authentication: A secure crypto-biometric verification protocol. *Trans. Info. For. Sec.*, 5(2):255–268, 2010.

[181] S. Urie and B. Martin. Biometric enabled casino gaming system, June 1 2004. US Patent 6,743,098.

[182] I. Van der Ploeg. The illegal body:eurodac'and the politics of biometric identification. *Ethics and Information Technology*, 1(4):295–302, 1999.

[183] C. Vastarouchas, S. Kapoulea, and C. Psychalinos. Ecg signal acquisition for the pan-tompkins algorithm using current-mirror filters. In *ICECS*, pages 317–320, 2016.

[184] I. Vasyltsov and C. Bak. Method for seamless unlock function for mobile applications. In *EMBC*, pages 2614–2617, 2016.

[185] I. Vasyltsov and S. Lee. Entropy extraction from bio-signals in healthcare iot. In *IoTPTS*, pages 11–17. ACM, 2015.

[186] K. K. Venkatasubramanian and S. K. S. Gupta. Physiological value-based efficient usable security solutions for body sensor networks. *ACM Trans. Sen. Netw.*, 6(4):31:1–31:36, 2010.

[187] K. K. Venkatasubramanian, Venkatasubramanian, A. Banerjee, and S. K. S. Gupta. Ekg-based key agreement in body sensor networks. In *INFOCOM Workshops*, pages 1–6, 2008.

[188] G. A. von Graevenitz. Biometric authentication in relation to payment systems and atms. *Datenschutz und Datensicherheit-DuD*, 31(9):681–683, 2007.

[189] J. Walker. A pseudorandom number sequence test program, 2017.

[190] S. Wang, Y. Zhang, X. Yang, P. Sun, Z. Dong, A. Liu, and T.-F.

Yuan. Pathological brain detection by a novel image feature – fractional fourier entropy. *Entropy*, 17(12):8278–8296, 2015.

[191] J. Wayman, A. Jain, D. Maltoni, and D. Maio. *An Introduction to Biometric Authentication Systems*, pages 1–20. Springer, 2005.

[192] J. Welch, P. Ford, R. Teplick, and R. Rubsamen. The massachusetts general hospital-marquette foundation hemodynamic and electrocardiographic database–comprehensive collection of critical care waveforms. *Clinical Monitoring*, 7(1):96–97, 1991.

[193] R. P. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[194] W. Wu, S. Pirbhulal, A. K. Sangaiah, S. C. Mukhopadhyay, and G. Li. Optimization of signal quality over comfortability of textile electrodes for ecg monitoring in fog computing based medical applications. *Future Generation Computer Systems*, 86:515 – 526, 2018.

[195] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li. Imdguard: Securing implantable medical devices with the external wearable guardian. In *INFOCOM*, pages 1862–1870, 2011.

[196] L. Yao, B. Liu, G. Wu, K. Yao, and J. Wang. A biometric key establishment protocol for body area networks. *International Journal of Distributed Sensor Networks*, 2011, 2011.

[197] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. Svc2004: First international signature verification competition. In *International conference on biometric authentication*, pages 16–22. Springer, 2004.

[198] E. K. Zaghouani, A. Jemai, A. Benzina, and R. Attia. Elpa: A new key agreement scheme based on linear prediction of ecg features for wban. In *EUSIPCO*, pages 81–85, 2015.

[199] G. H. Zhang, C. C. Y. Poon, and Y. T. Zhang. A fast key generation method based on dynamic biometrics to secure wireless body sensor networks for p-health. In *EMBC*, pages 2034–2036, 2010.

[200] G. H. Zhang, C. C. Y. Poon, and Y. T. Zhang. Analysis of using interpulse intervals to generate 128-bit biometric random binary sequences for securing wireless body sensor networks. *IEEE Transactions on Information Technology in Biomedicine*, 16(1):176–182, 2012.

[201] Y. Zhang, Y. Sun, P. Phillips, G. Liu, X. Zhou, and S. Wang. A

multilayer perceptron based smart pathological brain detection system by fractional fourier entropy. *Journal of Medical Systems*, 40(7):173, 2016.

[202] Z. Zhang, H. Wang, A. V. Vasilakos, and H. Fang. Ecg-cryptography and authentication in body area networks. *IEEE Transactions on Information Technology in Biomedicine*, 16(6):1070–1078, 2012.

[203] G. Zhao, G. Liu, H. Li, and M. Pietikainen. 3d gait recognition using multiple cameras. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 529–534. IEEE, 2006.

[204] G. Zheng, G. Fang, M. A. Orgun, and R. Shankaran. A comparison of key distribution schemes using fuzzy commitment and fuzzy vault within wireless body area networks. In *PIMRC*, pages 2120–2125, 2015.

[205] G. Zheng, G. Fang, R. Shankaran, M. Orgun, J. Zhou, L. Qiao, and K. Saleem. Multiple ecg fiducial points based random binary sequence generation for securing wireless body area networks. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2016.

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

[207] A. E. F. Zuniga, K. T. Win, and W. Susilo. Biometrics for electronic health records. *Journal of medical systems*, 34(5):975–983, 2010.

[208] A. Zúquete, B. Quintela, and J. P. da Silva Cunha. Biometric authentication using brain responses to visual stimuli. In *BIOSIGNALS*, pages 103–112, 2010.