Future Generation Computer Systems 77 (2017) 136–148

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Encryption by Heart (EbH)—Using ECG for time-invariant symmetric key generation

L. González-Manzano [a,*], José M. de Fuentes [a], P. Peris-Lopez [a,b], C. Camara [a]

[a] Computer Security Lab (COSEC), Carlos III University of Madrid. Avenida de la Universidad, 30, 28911 Leganes, Spain
[b] Aalto University, Konemiehentie 2, 02150 Espoo, Finland

HIGHLIGHTS

- We explore the use of ElectroCardioGram (ECG) data to symmetrically encrypt data.
- No previous approach has explored how ECG can produce symmetric encryption keys.
- EbH creates on-the-y, user-specific, time-invariant keys using current ECG values.
- 95.97% of unique keys, with up to 300 bits and 93.51 of min-entropy are produced.
- Experiments are carried out over a dataset of 199 subjects along 24 hours.

ARTICLE INFO

Article history:
Received 19 December 2016
Received in revised form 6 June 2017
Accepted 6 July 2017
Available online 26 July 2017

Keywords:
ECG
Symmetric encryption
Time-invariant keys

ABSTRACT

Wearable devices are a part of Internet-of-Things (IoT) that may offer valuable data of their porting user. This paper explores the use of ElectroCardioGram (ECG) records to encrypt user data. Previous attempts have shown that ECG can be taken as a basis for key generation. However, these approaches do not consider *time-invariant keys*. This feature enables using these so-created keys for symmetrically encrypting data (e.g. smartphone pictures), enabling their decryption using the key derived from the current ECG readings. This paper addresses this challenge by proposing EbH, a mechanism for persistent key generation based on ECG. EbH produces seeds from which encryption keys are generated. Experimental results over 24 h for 199 users show that EbH, under certain settings, can produce permanent seeds (thus time-invariant keys) computed on-the-fly and different for each user—up to 95.97% of users produce unique keys. In addition, EbH can be tuned to produce seeds of different length (up to 300 bits) and with variable min-entropy (up to 93.51). All this supports the workability of EbH in a real setting.

1. Introduction

Nowadays, wearable devices are a growing trend [1]. Smart bracelets that control sleepiness[1] or measure the heart pace,[2] smartshirts that measure and control the body sweat [2] and so on, are examples of the emergence and development of these devices. One characteristic of wearables is their short range [3]. They measure human characteristics being close or attached to the body, thus benefiting security because it is more difficult for an attacker to intercept collected data.

The widespread adoption of wearable devices and Implantable Medical Devices (IMDs), along with their wireless connectivity, make biosignals to be available for different purposes such as enabling smart e-health gateways [4]. However, a critical remark is that biological signals must be properly secured. As they convey personal information about the subject health status, it is crucial that only authorized entities can access that data. To this end, previous works have proposed different approaches to encrypt [5,6] or authenticate [7,8] biological signals.

Beyond the protection of the biosignals themselves, the focus of this paper is on their use to protect user personal information different from health data. Portable devices, and smartphones in particular, are an emerging trend worldwide. Among their uses, they serve to locally store private pictures or documents. Beyond their local storage, most of user data is stored in cloud services (e.g. Dropbox, Google Drive, etc.), thus easing their access everytime, everywhere. This is a huge benefit which cannot be confronted with security. Users are willing to easily access their data

* Corresponding author.
E-mail addresses: lgmanzan@inf.uc3m.es (L. González-Manzano), jfuentes@inf.uc3m.es (J.M. de Fuentes), pperis@inf.uc3m.es, pedro.peris-lopez@aalto.fi (P. Peris-Lopez), macamara@inf.uc3m.es (C. Camara).

[1] http://www.wareable.com/withings/best-sleep-trackers-and-monitors, last access Dec. 2016.

[2] http://www.bestfitnesstrackerreviews.com/fitness-trackers-for-heart-health.html, last access Dec. 2016.

http://dx.doi.org/10.1016/j.future.2017.07.018
0167-739X/

but without compromising its security [9,10] which is particularly relevant in the event of theft or loss. Nobody, except for the owner, should be able to access private data.

In order to protect data, encryption mechanisms are typically applied. There are two main categories—asymmetric and symmetric encryption [11]. The difference between them is in the applied keys. Thus, whereas asymmetric procedures use different keys for encryption and decryption, symmetric encryption needs the same for both processes.

This paper is focused on symmetric encryption, as it is faster and efficient in terms of the consumed resources, while offering suitable levels of security. A critical issue for its practical application is *key management*. According to the US National Institute of Standards and Technology, "key management provides the foundation for the secure generation, storage, distribution, and destruction of keys" [12]. In the considered setting, one interesting approach in this regard is deriving keys from user information. This removes the burden of key storage and destruction, as keys can be produced upon demand. Moreover, the proposed scenario does not need any kind of distribution. Therefore, secure key generation remains as the only open issue. To this extent, this paper focuses on a particular biosignal called ElectroCardioGram (ECG). Previous approaches have already pointed out that ECG signals can be used to generate encryption keys that change over time [13]. Other authors have explored how different IMDs or body sensors may derive a shared key leveraging on their independently perceived ECG signal [14,15].

Concerning secure key generation for symmetric encryption, there are three essential points. First, it is critical to derive a *time-invariant* key, that is, a key that remains the same over time. Consider that users are wearing a smart bracelet (or a portable medical device like a Holter) which collects their ECG signal. In this way, the secret key is derived from the collected signal, which can be seamlessly used for encrypting or decrypting sensitive data. Therefore, users will only be allowed to access their own data if the (current) decryption key is the same as the (initial) encryption one. On the other hand, the remaining critical aspects is that keys must be different across users and non-predictable. Otherwise, any unauthorized user could illegally gain access to private information. Both time-invariance and uniqueness ensure that users are allowed to access their data, at any time and in a private way. To the best of authors' knowledge, no previous approach based on ECG provides these issues. Therefore, this paper aims to address the following research questions:

- RQ1 Is it possible to create encryption/ decryption keys using the user's ECG signal?
- RQ2 Is each created key different between users?
- RQ3 Does each key remain invariant along the time?
- RQ4 Is each key difficult to reproduce?

To address these questions, this paper presents two main contributions. The first one is EbH, a mechanism to create time-invariant symmetric encryption keys derived from ECG. Users, at time T_1, generate a key using their ECG signal and encrypt their data. Then, for decryption purposes, at any time T_2, users generate the same key by using their ECG signal—recorded at the precise time when it is needed. Remarkably, EbH does not need any kind of key storage. Keys are created on-the-fly, through the collected ECG signal, and without the need of storing any key. To improve the security of the scheme, encryption keys are not produced directly by EbH, but they are derived from EbH output. This paradigm is referred to as key derivation and is a commonly accepted good security practice [16]. Therefore, EbH produces *seeds* that serve to derive encryption keys. The actual key derivation function is out of the scope of this proposal and, indeed, EbH could work with any highly non-linear function. The security of EbH is studied in terms of average number of attempts to decrypt (i.e. actual degree of time-invariance), min-entropy (i.e. unpredictability of seeds), probability of no decryption (i.e. degree of robustness) and difference among users (i.e. user-uniqueness). On the other hand, the second contribution of this paper is the implementation of EbH, which is made freely available. In this way, we aim to foster further research in this area.

Table 1
Notation.

Element	Description
$\mathcal{U}_i$	User i
$R_{\mathcal{U}_i}$	an entire Electrocardiogram (ECG) of $\mathcal{U}_i$
$ECG_{\mathcal{U}_i}^{T_*}$	ECG sample of$\mathcal{U}_i$ at time starting at T_*
$ECG_{\mathcal{U}_i}^{T_*}(k)$	Feature kth of ECG sample
L_w	Minimal partition (window) of ECG data (seconds)
L_a	Length of an *ECG sample* (seconds)
L_o	Observation period of ECG data (seconds)
$ECG_{Ref_{\mathcal{U}_i}}$	ECG reference data for $\mathcal{U}_i$
$\mathcal{S}_{\mathcal{U}_i}^{T_*}$	Seed for key generation for $\mathcal{U}_i$ at time T_*
$s_{\mathcal{U}_i}^{T_*}(k)$	Feature kth of seed $\mathcal{S}_{\mathcal{U}_i}^{T_*}$
$ECG_{Mod_{\mathcal{U}_i}}$	ECG model of $\mathcal{U}_i$
$ECG_{Mod_{\mathcal{U}_i}}(k)$	Feature kth of ECG model of $\mathcal{U}_i$
DT	Discard Threshold
TM	Tolerance Margin
T_i	Time at instant i
N_{feat}	Number of features
P_{ND}	Probability of no decryption
H_∞	Min-entropy [17]

The remaining of this paper is as follows. Section 2 describes the underlying model. Section 3 describes EbH, the proposed mechanism. Section 4 shows the proposal evaluation. Section 5 introduces the related work. Finally, Section 6 concludes the paper and points out future research directions.

2. Model

This section introduces the main concepts needed as well as the goals that have to be achieved by the proposed mechanism. Table 1 summarizes the notation used throughout the paper.

2.1. Definitions

The proposed mechanism receives some data as input and produces encryption keys as output. Concerning the input, let $R_{\mathcal{U}_i}$ be an entire Electrocardiogram (ECG) record of a user $\mathcal{U}_i$. In order for $R_{\mathcal{U}_i}$ be the input for EbH, two main processes have to be carried out. First, $R_{\mathcal{U}_i}$ is divided into windows of L_a seconds, thus leading to *ECG samples*. These samples are noted as $ECG_{\mathcal{U}_i}^{T_*}$ when they refer to user $\mathcal{U}_i$ starting at time T_*. The second process is to extract the member features of these samples, which will be critical for EbH. Thus, each sample is formed by a set of N_{feat} features, denoted as $ECG_{\mathcal{U}_i}^{T_*}(k)$, being k an ordinal value. Details of both processes are given in Section 3.

Before the key generation itself, it is assumed that the system knows the user $\mathcal{U}_i$ for some time t. Thus, this knowledge is referred to as *ECG reference data* and it is formed by a set of non-overlapping ECG samples. Thus, this data is denoted as $ECG_{Ref_{\mathcal{U}_i}} = \{ECG_{\mathcal{U}_i}^{T_0}, ECG_{\mathcal{U}_i}^{T_1}, \ldots, ECG_{\mathcal{U}_i}^{T_t}\}$. With respect to the output, the mechanism must produce seeds $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ that can be applied to generate encryption keys. The actual key derivation function is out of the scope of this proposal. Each seed is formed by a set of N_{feat} elements, $\mathcal{S}_{\mathcal{U}_i}^{T_*} = \{s_{\mathcal{U}_i}^{T_*}(1), s_{\mathcal{U}_i}^{T_*}(2), \ldots, s_{\mathcal{U}_i}^{T_*}(N_{feat})\}$, where T_* represents the moment in which the seed starts to be computed.

2.2. Goals

The proposed approach has to meet the following three goals, inspired in the features that are needed for any biometric system to become an identifier [18] and also pointed out as research questions. First, seeds $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ must be *distinctive* per user. This property is essential to ensure that no other user produces the same seed. In practice, this ensures that data encrypted by user $\mathcal{U}_i$ cannot be decrypted by any other user $\mathcal{U}_j$. Mathematically, being *NS* the population size,

Goal 1 (*Uniqueness (Research Questions RQ1, RQ2)*)**.**

$$\text{For each } p \neq q \quad \mathcal{S}_{\mathcal{U}_p}^{T_r} \neq \mathcal{S}_{\mathcal{U}_q}^{T_s} \quad \forall r, s \quad \text{and} \quad \{\mathcal{U}_p, \mathcal{U}_q\} \in \{\mathcal{U}_i\}_{i=1}^{NS}. \tag{1}$$

On the other hand, seeds $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ have to be *time-invariant*, thus remaining constant over time. This feature enables that $\mathcal{U}_i$ can encrypt and decrypt her personal information at any point in time. This is the basis on top of which all symmetric encryption algorithms work—both encryption and decryption is carried out using exactly the same key [11,19]. Being T_p and T_q two arbitrary instants, this goal can be formalized as follows:

Goal 2 (*Time-Invariance (Research Questions RQ1, RQ3)*)**.**

$$\text{For all } T_p \neq T_q \quad \mathcal{S}_{\mathcal{U}_i}^{T_p} = \mathcal{S}_{\mathcal{U}_i}^{T_q} \quad \forall p, q \quad \text{and} \quad \mathcal{U}_i \in \{\mathcal{U}_i\}_{i=1}^{NS}. \tag{2}$$

Finally, the mechanism must be *invulnerable*, that is, seeds $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ have to be difficult to reproduce and guess by any third party. This prevents any attacker from being able to compute $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ thus gaining access to protected information. Mathematically, the probability that an adversary (Adv) gains access to the private information of a legitimate user ($\mathcal{U}_i$), can be expressed as follows:

Goal 3 (*Invulnerability (Research questions RQ1, RQ4)*)**.**

$$P_{Adv}(\mathcal{S}_{Adv}^{T_p} = \mathcal{S}_{\mathcal{U}_i^*}^{T_q}) = \delta \quad \forall p, q \quad \text{and} \quad \mathcal{U}_i \in \{\mathcal{U}_i\}_{i=1}^{NS} \tag{3}$$

where δ represents a negligible value.

3. ECG-based key generation mechanism

This section introduces EbH, the proposed mechanism as the first contribution of this paper. Section 3.1 gives an overview of EbH. Afterwards, Section 3.2 describes the core of EbH.

3.1. Overview

The overview of EbH is presented in Fig. 1. EbH requires the user wearing a smart device (e.g., a sport bracelet or a medical-external heart monitor) that provides ECG values. This device is wirelessly connected to a hub (typically a smartphone or even an ad-hoc transmitter/receiver), which contains the data to be protected.

Before protecting the user information, the ECG user model ($ECG_{Mod_{\mathcal{U}_i}}$) is built. This process is executed once at the setup phase and may be repeated after some years due to variations in the ECG signal [20]. For this purpose, the smart device provides a set of ECG values (ECG reference data $ECG_{Ref_{\mathcal{U}_i}}$, recall Section 2.1) which are processed by the hub device.

At any time T_1 when the user wants to encrypt certain data, the hub gathers a set of ECG samples $ECG_{\mathcal{U}_i}^{(obs(T_1))} = \{ECG_{\mathcal{U}_i}^{T_{1-L_o}}, ECG_{\mathcal{U}_i}^{T_{1-(L_o+1)}}, \ldots, ECG_{\mathcal{U}_i}^{T_1}\}$, where L_o is the size of the observation period. The need for this period is motivated below.

Table 2
Example $ECG_{Mod_{\mathcal{U}_i}}$ creation from $ECG_{Ref_{\mathcal{U}_i}}$.

	0.7	0.2	1.2	0.4	−0.1
$ECG_{Ref_{\mathcal{U}_i}}$	0.5	0.7	−0.2	1.1	−1.2
	1.2	−1	0.4	−0.5	0.8
$ECG_{Mod_{U_i}}$	**0.80**	**−0.03**	**0.47**	**0.33**	**−0.17**

The seed $\mathcal{S}_{\mathcal{U}_i}^{T_1}$ is derived by the hub from both $ECG_{Mod_{\mathcal{U}_i}}$ and its similarity to $ECG_{\mathcal{U}_i}^{(obs(T_1))}$. In a nutshell, $\mathcal{S}_{\mathcal{U}_i}^{T_1}$ is composed by a set of features, each one being the result of the similarity between the corresponding features of $ECG_{Mod_{\mathcal{U}_i}}$ and $ECG_{\mathcal{U}_i}^{(obs(T_*))}$. For the sake of robustness, this comparison is only performed in those features of $ECG_{Mod_{\mathcal{U}_i}}$ that convey enough cardiac information. For the cases in which it does not happen, the corresponding feature $s_{\mathcal{U}_i}^{T_1}(k)$ of $\mathcal{S}_{\mathcal{U}_i}^{T_1}$ is given a neutral value.

After obtaining the seed $\mathcal{S}_{\mathcal{U}_i}^{T_1}$, it is taken as input to a random process (e.g. a Pseudo-Random Number Generator (PRNG) [11,21]) to generate the encryption key $K_{\mathcal{U}_i}$. However, the design of such a key derivation function is outside the scope of EbH.

Once the user, at any time T_2, wants to decrypt previously encrypted data, the key generation process is the same as the one described. The main difference is that current ECG values ($ECG_{\mathcal{U}_i}^{(obs(T_2))}$) are taken for this calculation. As it is a symmetric decryption, it is only successful if the decryption key is the same as the encryption one.

3.2. Mechanism description

The process of key derivation involves two main steps, namely user model creation and seed computation. Each one is described in a separate section.

3.2.1. ECG user model creation

In this step, the ECG user reference data is considered. In particular, each ECG sample is transformed so that each sample contains a set of features.

Feature extraction of ECG signals can be grouped into two main categories. Fiducial-based approaches use characteristics points (e.g., amplitude, relative amplitude and time duration between peaks PQRS) of an ECG signal to derive the features [22,23]. On the other hand, non-fiducial methods use a transform domain like Fourier, Hadamard or Wavelet to extract the features [24,25].

After feature extraction, the model $ECG_{Mod_{\mathcal{U}_i}}$ for user $\mathcal{U}_i$ is built by computing the average of each feature in $ECG_{Ref_{\mathcal{U}_i}}$, as shown in Table 2. For simplicity, Table 2 shows a reference model formed by three ECG samples, each one with five features (i.e. $N_{feat} = 5$).

3.3. Seed computation

In order to compute the seed $\mathcal{S}_{\mathcal{U}_i}^{T_*}$, there are two main data elements to be considered. On the one hand, the user model $ECG_{Mod_{\mathcal{U}_i}}$ described in Section 3.2.1. On the other hand, a set of current ECG samples $ECG_{\mathcal{U}_i}^{T_*}$ for an observation period starting at time T_*. Over this set of samples, feature extraction is carried out in the same way as it was done with the user model.

The rationale behind using an observation period (which may span across one or more ECG samples) is that the resulting seed $\mathcal{S}_{\mathcal{U}_i}^{T_*}$ is computed considering the similarity between $ECG_{Mod_{\mathcal{U}_i}}$ and $ECG_{\mathcal{U}_i}^{(obs(T_*))}$. Thus, it is expected that the bigger the observation period is, the more similar ECG values are as compared to the user model.

In particular, seeds are formed by the same amount of features (N_{feat}) that appear in the considered ECG samples. The value of each feature $s_{\mathcal{U}_i}^{T_*}(k)$ of the seed is determined by the similarity between

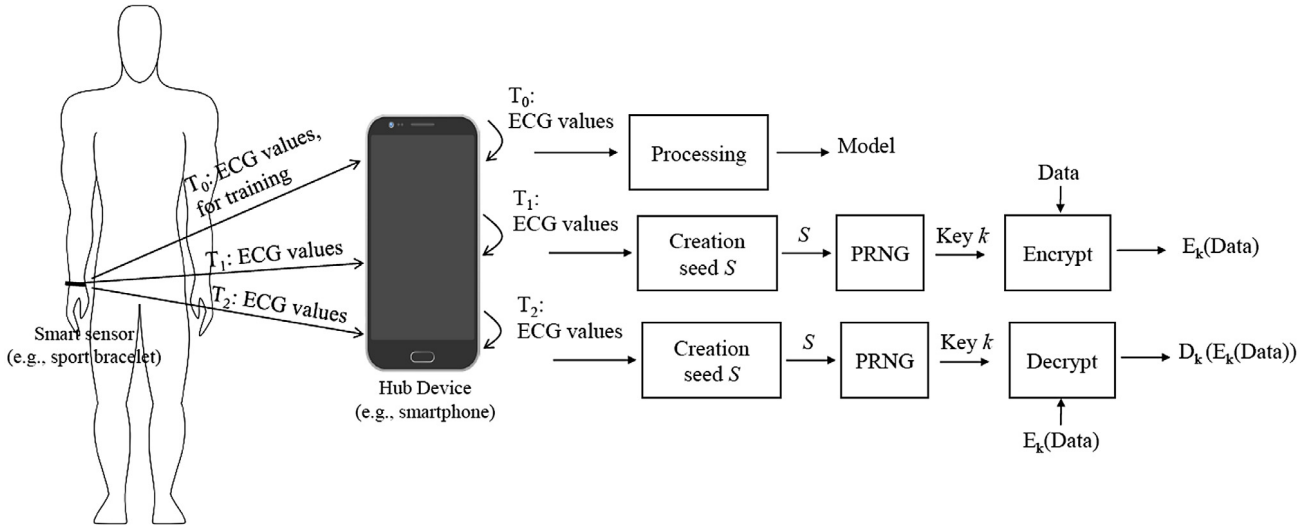

Fig. 1. Overview of EbH using a smartphone as hub and a smart bracelet as ECG data source.

$ECG_{Mod_{\mathcal{U}_i}}(k)$ and $ECG_{\mathcal{U}_i}^{(obs(T_*))}(k)$, following the procedure described below.

Similarity analysis is driven by two parameters, namely Discard Threshold (DT) and Tolerance Margin (TM). DT is used to discard features whose value is considered significantly small and thus unstable to be involved in the seed generation process. For this purpose, only $ECG_{Mod_{\mathcal{U}_i}}$ is taken into account, according to Eq. (4). Thus, feature $s_{\mathcal{U}_i}^{T_*}(k)$ is given a neutral value if the absolute value of $ECG_{Mod_{\mathcal{U}_i}}(k)$ is below DT. Note that the absolute value is applied to make EbH compatible with feature extraction strategies that may give negative values. Thus, Eq. (4) is applied to all N_{feat} features of $\mathcal{S}_{\mathcal{U}_i}^{T_*}$.

$$s_{\mathcal{U}_i}^{T_*}(k) = \begin{cases} \text{neutral} & \text{if } |ECG_{Mod_{\mathcal{U}_i}}(k)| < DT \\ \text{not neutral (see Eq. (5))} & \text{otherwise.} \end{cases} \quad (4)$$

For the cases in which a given feature $s_{\mathcal{U}_i}^{T_*}(k)$ has not been valued as neutral, parameter TM comes into play. TM indicates the tolerance margin to consider that a given feature $ECG_{Mod_{\mathcal{U}_i}}(k)$ and $ECG_{\mathcal{U}_i}^{(obs(T_*))}(k)$ are close enough. Eq. (5) describes the three possible cases to assign values to each non-neutral feature $s_{\mathcal{U}_i}^{T_*}(k)$.

$$s_{\mathcal{U}_i}^{T_*}(k) = \begin{cases} -1 & \text{if } ECG_{Mod_{\mathcal{U}_i}}(k) < 0 \text{ and} \\ & ECG_{\mathcal{U}_i}^{(obs(T_*))}(k) \in [ECG_{Mod_{\mathcal{U}_i}}(k) - TM, ECG_{Mod_{\mathcal{U}_i}}(k) + TM] \\ 1 & \text{if } ECG_{Mod_{\mathcal{U}_i}}(k) > 0 \text{ and} \\ & ECG_{\mathcal{U}_i}^{(obs(T_*))}(k) \in [ECG_{Mod_{\mathcal{U}_i}}(k) - TM, ECG_{Mod_{\mathcal{U}_i}}(k) + TM] \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Both TM and DT can be easily identified in Fig. 2. The histogram represents features $ECG_{Mod_{\mathcal{U}_i}}(k)$ along with TM (white bars). Furthermore, gray bars show the values of $ECG_{\mathcal{U}_i}^{(obs(T_*))}(k)$ (assuming $N_{feat} = 5$). Horizontal lines show the effect of DT—feature 2 is neutralized. At each feature (i.e. pair of bars) and considering Eqs. (4) and (5), the value for $s_{\mathcal{U}_i}^{T_*}(k)$ is pointed out. Thus, at the light of Fig. 2, the generated seed is $\mathcal{S}_{\mathcal{U}_i}^{T_*} = \{0, \times, 1, 0, -1\}$, where $\times$ represents a neutral value.

4. Evaluation

This section introduces the assessment carried out over the proposed mechanism. For this purpose, Sections 4.1 and 4.2 describe the experimental dataset and how it has been prepared to serve as an input for EbH. Feature extraction is described in Section 4.3.

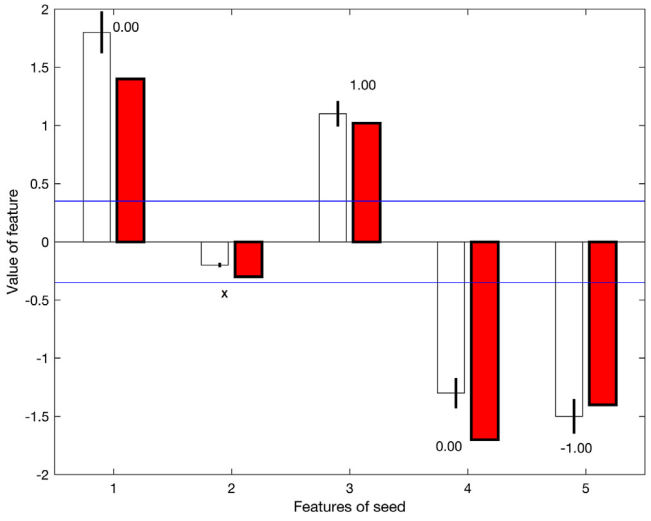

Fig. 2. Seed generation example for $N_{feat} = 5$. White bars represent $ECG_{Mod_{\mathcal{U}_i}}(k)$, along with their DT (vertical line in the edge). Colored bars represent $ECG_{\mathcal{U}_i}^{(obs(T_*))}(k)$. Horizontal lines show TM. Numbers on each bar are the corresponding values for $s_{\mathcal{U}_i}^{T_*}(k)$, where $\times$ represents the neutral value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Afterwards, Section 4.4 defines the assessment criteria, whereas Section 4.5 shows and discusses the experimental results. Our prototype implementation is freely available at http://github.com/jmdefuentes/EbH.

4.1. Dataset description

Experiments have been carried out using dataset E-HOL-03-0202-003, provided by the Telemetric and ECG Warehouse (THEW) of University of Rochester.[3] The rationale of using this dataset is threefold. On the one hand, it includes long-term recordings (around 24 h) of subjects. It is particularly appropriate in this scenario to analyze the time-invariance feature of keys. On the other hand, the subjects do not have any significant cardiac disease—that is, the population is homogeneous without any bias between

[3] http://thew-project.org/index.htm, last access Dec. 2016.

individuals. Finally, the dataset is composed of 199 subjects, which is very convenient to study the user uniqueness of derived keys—three subjects have been discarded from the original dataset due to an insufficient size of the file.

4.2. Data pre-processing

The input of our system consists on ECG values that are continuously (i.e., sampling frequency is set at 200 Hz) obtained from a set of individuals. Some manipulations of the ECG signal are needed for the generation of the seed $\mathcal{S}_{\mathcal{U}_i}^{T_*}$. In Fig. 3, the pre-processing algorithm is summarized.

First of all the ECG signal must be cleaned before any other process. For that, the existing sources (i.e., respiration and power-line) of noise are eliminated (Fig. 3, step 1). More precisely, first the DC component (average value) is eliminated and then a pass-band filter is employed. For this filter, the lower-cut frequency and the upper-cut-off frequency are set to 0.67 Hz and 45 Hz, respectively. The lower stop-band aims to eliminate the noise produced by the subject respiration. On the other hand, the upper-stop band frequency pursues the elimination of the power-line noise and also the preservation of as much information as possible in the pass-band.

Once the signal is clean, ECG samples $ECG_{\mathcal{U}_i}^{T_*}$ are obtained. For this purpose, two steps are carried out. First, the ECG record is split into windows of 2 s ($L_w = 2$; Fig. 3, step 2). This decision is inspired on previous and well-known ECG-based identification proposals [26]. Therefore, each portion corresponds to 2–3 heart beats since a healthy individual beats 60–100 times per minute. Afterwards, a set of windows conforms an ECG sample, which is set to a given amount of minutes (parameter L_a in our experimentation). In other words, each ECG sample $ECG_{\mathcal{U}_i}^{T_*}$ represents a period of time of subject $\mathcal{U}_i$, starting at time T_*.

4.3. Feature extraction

Once data has been pre-processed, the first step is to extract features from ECG samples (recall Section 3.2.1). Among all the feature extraction algorithms, the Walsh–Hadamard Transform (WHT) is chosen [27] (Fig. 3, step 3). Thus, each coefficient of this transform becomes a feature $ECG_{\mathcal{U}_i}^{T_*}(k)$. The choice of WHT is motivated by two reasons. First, it is very efficient from the computational point of view since it can be implemented by a matrix multiplication. Second, it works with signal compression, keeping the majority of the information at the lower frequency coefficients [28,29]. Fig. 4 illustrates this issue by showing 3-beats of an ECG signal at the time-domain and at the WHT domain. It can be seen that coefficients beyond 250 are less representative.

In our experimentation we started considering the lower 256 coefficients ($N_{feat} = 256$) as a trade-off between the system efficiency and information preservation. Furthermore, a procedure to reduce N_{feat} has been applied. In particular, selection of attributes has been carried out using correlation-based feature subset selection for the attribute evaluator and best-first search strategy for the search method [30]. Experimentally, the best results were achieved with an amount of $N_{feat} = 194$ features.

The particular procedure for feature extraction is described below. For each window of 2 s ($L_w = 2$ s) within an ECG sample $ECG_{\mathcal{U}_i}^{T_*}$ of 180 s ($L_a = 180$ s—recall Section 4.2), WHT is computed, thus obtaining a set of N_{feat} WHT coefficients. Afterwards, each feature of the sample is computed as the average of the corresponding coefficients (Fig. 3, step 4). Thus, the kth feature $ECG_{\mathcal{U}_i}^{T_*}(k)$ of the sample is the average of the kth coefficient of all its windows. This procedure is repeated for all features.

4.4. Assessment criteria

The evaluation of EbH involves different assessment criteria, one for each pursued goal (recall Section 2.2). Each criterion is presented in this Section and will be applied in Section 4.5. In the following, all criteria will be referred to seeds since EbH makes use of a key derivation procedure. Thus, ensuring the achievement of goals in respect to seeds is equivalent to doing the same over the resulting keys.

Regarding uniqueness (Goal 1), the coincidence of two users producing the same seed ($\mathcal{S}_{\mathcal{U}_p}^{T_r} = \mathcal{S}_{\mathcal{U}_q}^{T_s}$) is measured, leading to *unequivocal* and *non-unequivocal* implementations.

With respect to time-invariance (Goal 2), this will be measured by two criteria: (1) the average number of attempts to decrypt (referred to as *AD*) and (2) the probability of no decryption (denoted as P_{ND}). Regarding *AD*, time-invariance is achieved if $AD = 1$. Recall that in EbH data is encrypted first using the first seed, i.e. $\mathcal{S}_{\mathcal{U}_i}^{T_0}$. Decryption is carried out at any posterior time T_d using the corresponding seed $\mathcal{S}_{\mathcal{U}_i}^{T_d}$. Thus, if $AD = 1$, it means that the user can decrypt with the said seed with no delay. Otherwise, if $AD = 2$, it means that, on average, seed $\mathcal{S}_{\mathcal{U}_i}^{T_d}$ is not valid to decrypt, but the following one (i.e. $\mathcal{S}_{\mathcal{U}_i}^{T_{d+1}}$) is valid. Therefore, the user must wait for some period (in this example, L_a is 3 min, recall Section 4.2) before decrypting.

The worst case of decryption happens if the first seed $\mathcal{S}_{\mathcal{U}_i}^{T_0}$ is different from the remaining ones produced by user $\mathcal{U}_i$. In this case, $AD = \infty$. However, the probability of this happening has to be considered. Let us consider a case with three seeds, two of them equal. If any of the equal ones are produced first, it is straightforward to see that $AD = (1 + 2)/2 = 1.5$ attempts. However, if the different one is first, then $AD = \infty$. In order to consider this issue, *AD* is formalized as follows.

Seeds $\{\mathcal{S}_{\mathcal{U}_i}^{T_1}, \mathcal{S}_{\mathcal{U}_i}^{T_2}, \ldots, \mathcal{S}_{\mathcal{U}_i}^{T_n}\}$ produced by user $\mathcal{U}_i$ are divided in groups according to their values. Thus, let us rewrite this set as $\mathcal{S}_{\mathcal{U}_i} = \{\{\mathcal{S}_{\mathcal{U}_i}^{T_*} = \alpha\}, \{\mathcal{S}_{\mathcal{U}_i}^{T_*} = \beta\} \ldots \{\mathcal{S}_{\mathcal{U}_i}^{T_*} = \omega\}\}$. Let m be the number of seeds, N_{ss} the number of subsets, $|G_i|$ the amount of seeds in subset ith and $|G \neq 1|$ the amount of subsets with more than one seed. Thus, *AD* is defined as shown in Eq. (6).

$$AD = \frac{\sum_{i=1}^{N_{ss}} |G_i| \cdot \frac{|G_i|-1}{m-1}}{|G \neq 1|}. \tag{6}$$

On the other hand, P_{ND} measures the likelihood of not being able to decrypt data. Thus, P_{ND} is calculated by dividing the amount of subsets with only one seed ($|G \neq 1|$) by the total number of seeds (m), see Eq. (7). It is easy to notice that each subset with only one seed leads to undecryptable data.

$$P_{ND} = \frac{N_{ss} - |G \neq 1|}{m}. \tag{7}$$

Concerning the invulnerability goal (Goal 3), two magnitudes are studied. First, the seed length, which determines the size of the search space by the adversary. Associated with this issue, the brute-force effort to carry out such a search must be considered. Second, the seed randomness, which is measured by min-entropy [17]. This magnitude is considered as the most conservative way to measure this issue. Recall that encryption keys are produced after a key derivation function seeded with $\mathcal{S}_{\mathcal{U}_i}^{T_*}$. Thus, seed unpredictability (i.e. entropy) should be calculated to discuss the appropriateness of their use.

Min-entropy is defined considering a set of elements $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ which have an associated probability distribution $P = \{p_1, p_2, \ldots, p_n\}$. It is calculated according to Eq. (8).

$$H_\infty = -log(max(p_i)). \tag{8}$$

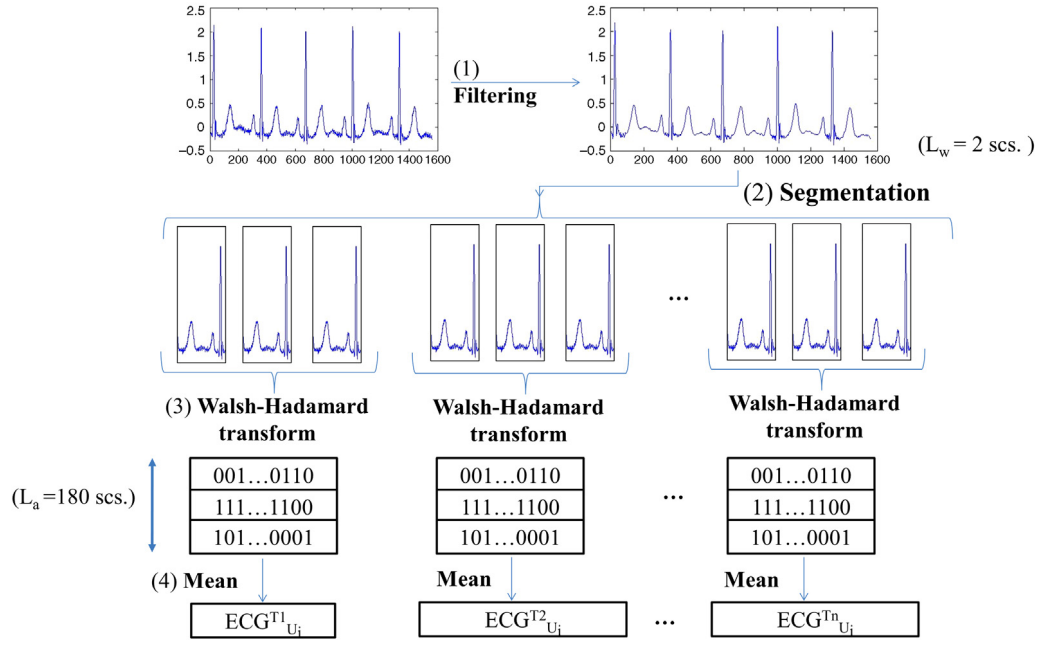

Fig. 3. Data pre-processing procedure of ECG records.

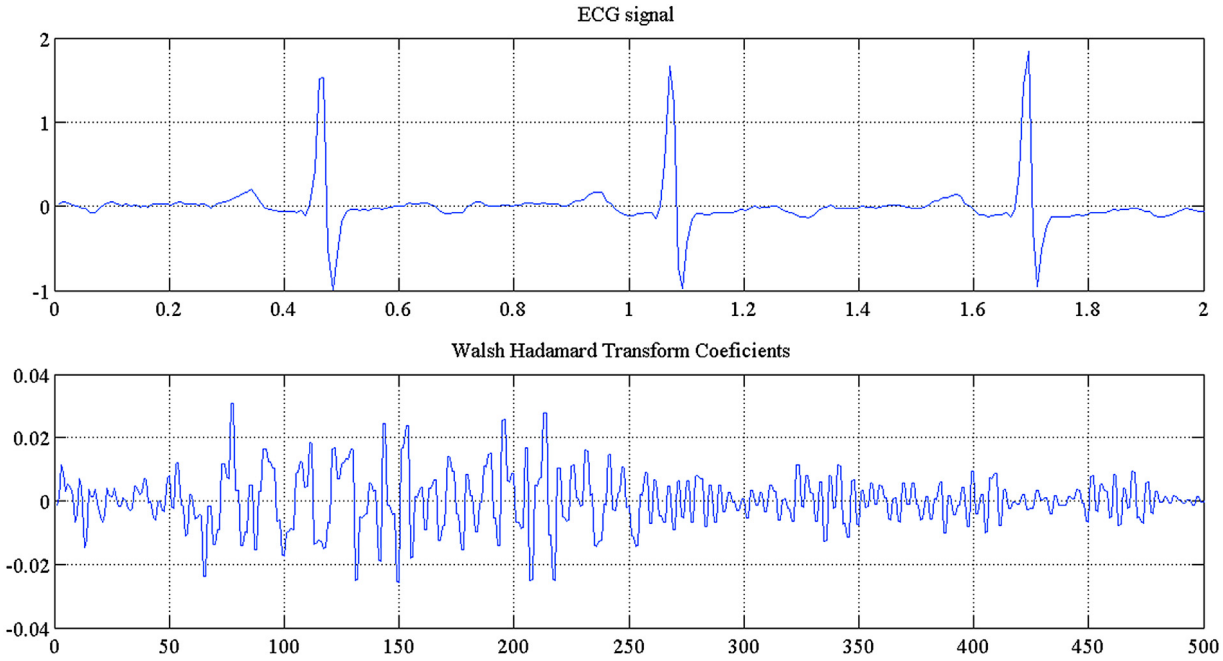

Fig. 4. ECG signal and Walsh–Hadamard spectrum.

Thus, in EbH, we calculate the total min-entropy as the sum of the min-entropies for all N_{feat} features. Thus, min-entropy is calculated per each feature $s^{T_*}_{U_i}(k)$, and a total for all features is computed. Eq. (9) formalizes the applied min-entropy, where $p^{(i)}_{max}$ is the maximum probability of each value (i.e. 1, 0, −1 or neutral, recall Section 3.3) in feature ith.

$$H_\infty(EbH) = \sum_{i=1}^{N_{feat}} -\log(\max(p^{(i)}_{\max})). \tag{9}$$

4.5. Experimental assessment

This section introduces the results of the experimental assessment. For the sake of clarity, the content is organized according to EbH intended goals (Section 2.2). Therefore, time-invariance and uniqueness are studied in Section 4.5.1 whereas invulnerability is analyzed in Section 4.5.2. Remarks about the practicality of the approach are shown in Section 4.5.3. Moreover, other side results of the experimentation are presented in Section 4.5.4. Tables A.5, A.6, A.7, and A.8 in the Appendix show the numerical results in detail.

In these experiments, there are five parameters to consider. First, the amount of ECG samples that constitute the ECG reference data ($ECG_{Ref_{\mathcal{U}_i}}$) needed to built the ECG user model ($ECG_{Mod_{\mathcal{U}_i}}$). Secondly, parameter L_a determines how much time each sample $ECG^{T*}_{\mathcal{U}_i}$ represents. This is important as it is the smallest data unit handled by EbH and, without prejudice of generality, L_a is a multiple of L_w. Third, L_o defines the amount of time the user is observed prior to deriving each seed $\mathcal{S}^{T*}_{\mathcal{U}_i}$. The two last parameters are the discard threshold DT and the threshold margin TM introduced in Section 3.3.

In the following, we report the results for different values for each parameter. Regarding the amount of samples in the model, we have considered 60% of the whole set of samples inspired on the value commonly used in machine learning [31] (referred to as training samples). For completeness we have also analyzed the polarized case when the model is only built with 20% of samples. With respect to L_a, we have considered 3 min. L_o has been set to $3 \cdot L_a$ and $10 \cdot L_a$ (i.e. 9 and 30 min, respectively). DT is valued 3% and 0.5%, whereas TM ranges from 1% to 20%.

All these values have been determined after extensive experimentation—they have been selected as they lead to an illustrative discussion. In Section 4.5.3, guidelines are provided to parameterize the system in practice. One important remark is that, for a real-world scenario, the size of the training set (i.e., 20% or 60% in our experiments) represents the time needed for the system to start working. Thus, in our experiments and considering the size of our dataset, EbH needs to be trained for 4.8 h (resp. 14.4 h). Remarkably, this action should be carried out once during the whole system lifetime. We leverage the rest of the dataset (i.e., 19.2 h or 9.6 h, respectively) to assess that EbH keeps meeting the intended goals for a long time after being trained.

4.5.1. Time-invariance and uniqueness

These two goals are achieved when each user generates the same seed over time and when it is different from that of other users. Fig. 5 shows the amount of seeds for several settings. Among all parameters, the size of the training set has a critical impact on the amount of seeds. The system outperforms well when this parameter is set to 60%, and in particular, a maximum of 535 seeds (i.e. 2.69 seeds/user) are generated. On the contrary, a worsening of the system is observed when the training set is reduced to 20% and this value raises to 2009 (i.e. 10.1 seeds/user).

On the other hand, the tolerance margin TM parameter also has a significant impact on the amount of seeds. Thus, the bigger it is, the higher the amount of seeds. Again, this trend is also according to the expectations—bigger tolerances enable more valid values for each single feature, thus leading to different combinations of values for each feature.

The remaining settings, discard threshold DT and length of the observation period L_o, do not have any consistent effect on this matter.

Based on these results, there are some settings in which time-invariance is significantly achieved. In particular, when $L_o = 10 \cdot L_a$, $DT = 0.03$, $TM = 0.01$ and 60% of samples are taken to build $ECG_{Mod_{\mathcal{U}_i}}$, 208 seeds have been produced for the 199 users (Table A.8). This implies 1.04 keys per user on average. This result supports the time-invariance of the approach, as most of the time users will be producing the same seed. Indeed, this happens in up to 183 users when $ECG_{Mod_{\mathcal{U}_i}}$ is formed by 20% of samples (Table A.5) and 191 when it is formed by 60% of samples (Table A.6).

Apart from the mere amount of seeds, time-invariance is also measured by the attempts to decrypt AD and the probability of no decryption P_{ND}, also presented in Tables A.5 and A.6. Regarding AD, the best result (1.13 attempts) is achieved with the said parameter values. This value is significantly affected by TM —almost 5 attempts are needed to decrypt a piece of data when TM is raised to 0.05. It can be seen that this magnitude raises faster than the growth of TM itself. Concerning P_{ND}, it is noticeable that all settings lead to a maximum probability of no decryption of 5%. Interestingly, when $TM = 0.01$, there is no probability of this undesired situation happening.

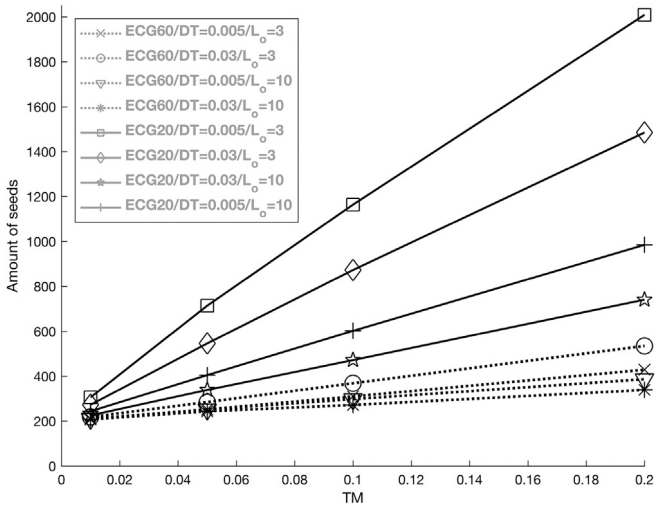

Fig. 5. Amount of seeds $\mathcal{S}^{T*}_{\mathcal{U}_i}$ depending on TM. Series correspond to different combinations of the amount of samples in $ECG_{Mod_{\mathcal{U}_i}}$ (ECG60 and ECG20 refer to 60% and 20% training set respectively), the value of L_o (in multiples of L_a) and DT.

With respect to distinctiveness, it must be noted that it is achieved if all of them are different. As shown in Tables A.7 and A.8, our results show that most configurations lead to unique seeds per user (see values without $*$ in the said tables). However, a small amount of settings (only when $DT = 0.005$ for $ECG_{Mod_{\mathcal{U}_i}}$ with 60% of samples) lead to seeds that are shared by at least two users. It must be noted that the impact is limited thanks to the size of the user set. Indeed, in order for a malicious user to benefit from this fact, she should guess which are the sharers in order to gain access to their personal information.

4.5.2. Invulnerability

The invulnerability of the generated seeds is given by their size and randomness. Each issue is studied separately.

Size analysis. Fig. 6 shows the size of seeds for the considered settings. It must be noted that the size is measured in amount of symbols, each one having values $\{1, 0, -1, neutral\}$, as explained in Section 3.3. Thus, if each symbol were coded by 2 bits, the actual seed size would be doubled. For the sake of generality, we leave the actual bit coding out of the discussion.

As it may be seen, seeds range from 115 symbols to 175. Interestingly, having $DT = 0.03$ also causes that under that percentage the seed is around 118 symbols. This trend is in line with expectations—if DT is bigger, more ECG features are not considered for the seed derivation process, thus causing shorter values. The size of the training set has also an impact on the seeds size. In particular, the size grows when the training set is bigger. However, the effect of this variable is limited, as it only causes a minor variation. According to our experiments, having a bigger training set implies that the ECG reference model is less impacted by incidental minimum values that cause that some features fall below the DT threshold. On the other hand, neither TM or L_o cause any significant effect on this issue.

The seed size also determines the hardness of a potential brute-force search. To quantify this issue, recall that the seed $\mathcal{S}^{T*}_{\mathcal{U}_i}$ is a vector of length $1 \times N_{feat}$ positions in which p of them can be activated with 3 different values $\{0, 1, -1\}$. Then, the number of

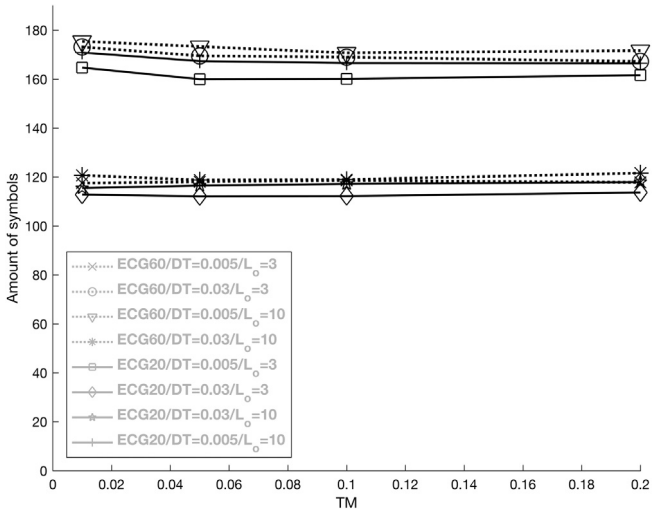

Fig. 6. Size of $\mathcal{S}^{T_*}_{\mathcal{U}_i}$ (in number of symbols) depending on *TM*. Series correspond to different combinations of the amount of samples in $ECG_{Mod_{\mathcal{U}_i}}$ (ECG60 and ECG20 refer to 60% and 20% training set respectively), the value of L_o (in multiples of L_a) and *DT*.

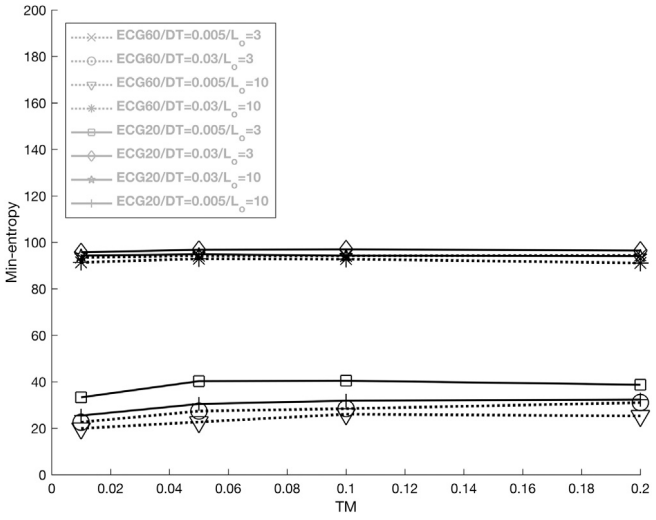

Fig. 7. Min_Entropy per seed $\mathcal{S}^{T_*}_{\mathcal{U}_i}$ depending on *TM*. Series correspond to different combinations of the amount of samples in $ECG_{Mod_{\mathcal{U}_i}}$ (ECG60 and ECG20 refer to 60% and 20% training set respectively), the value of L_o (in multiples of L_a) and *DT*.

Table 3
NA (clock cycles) and corresponding years.

		p			
		100		150	
		NA	Years	*NA*	Years
N_{feat}	194	$1.4 \cdot 10^{58}$	$4.9 \cdot 10^{41}$	$8.9 \cdot 10^{43}$	$3.1 \cdot 10^{27}$
	256	$1.2 \cdot 10^{73}$	$4.2 \cdot 10^{56}$	$1.3 \cdot 10^{74}$	$4.5 \cdot 10^{57}$

attempts (*NA*) needed (in the worst case) to find the seed is given by Eq. (10). *NA* corresponds to all combinations of N_{feat} values taken p at a time multiplied by the amount of permutations of 3 values in each of the p positions.

$$NA = \frac{N_{feat}!}{p!(N_{feat} - p)!} \cdot 3^p. \tag{10}$$

To contextualize the above equation we provide some numerical values regarding the cost for an attacker. We assume that each seed can be generated in just one instruction. To estimate the amount of time an attacker would need, let consider the use of an Intel Core i7 processor which runs 92 billion instructions per second.[4] Table 3 presents *NA* (clock cycles) and its associated amount of years for $N_{feat} = \{194$ attribute selection, 256 all features$\}$ and $p = \{100, 150\}$. We consider these values for p as they are in line with the obtained seed sizes.

It is noteworthy the huge number of attempts and years required to get the necessary seed applying brute force, for instance for $N_{feat} = 194$ and $p = 150$, up to $3.1 \cdot 10^{27}$ years would be needed. Besides, an attacker should also consider the fact that p is not static and then, much more attempts should be performed with different p. Consequently, this study guarantees the infeasibility of applying a brute force attack against our proposed method EbH. Taking into account Eq. (10), the security level of our proposal EbH is lower bounded by $\delta = \frac{1}{2^{\log_2(NA)}}$.

Seed unpredictability. With respect to unpredictability, Fig. 7 shows the evolution of the min-entropy of generated seeds under the considered settings.

All parameters in these experiments have some effect on the min-entropy, being *DT* the one with higher impact. Surprisingly, having higher values of *DT* cause achieving better min-entropy. According to our experiments, the reason behind is that *DT* removes some features that are consistently having the same value in the resulting seeds, thus becoming predictable.

Another interesting finding is that the min-entropy benefits from having smaller training sets. This is due to the fact that a smaller training set leads to an ECG reference model that is not always that consistent with posterior observations. This causes more variability among seeds, thus achieving higher min-entropy. In a similar way, having larger L_o cause lower entropy—this implies having more data from the user, thus making ECG features to be more predictable. Last but not least, the tolerance margin has also a limited impact on the min-entropy. In particular, higher tolerance values typically cause a small increase on min-entropy. Nevertheless, this increase is not representative and it does not hold for all experiments.

4.5.3. Practical remarks

In order for EbH to be practical, keys have to be produced in a short period of time. Otherwise, the user would be forced to wait for protecting her data or for accessing to them. Moreover, EbH storage needs have to be affordable for state-of-the-art devices. Both issues are studied first. On the other hand, when EbH is going to be applied in practice, some guidelines must be given to tune the system, beyond the per-goal analysis shown in Section 4.5. These guidelines are provided at the end of this Section.

Time and storage considerations. There are two issues that have to be taken into account. On the one hand, the user has to be observed for a period determined by parameter L_o. However, recalling the definition of $ECG^{(obs(T_i))}_{\mathcal{U}_i}$ given in Section 3.1, at any time T_i that the user wants to generate a key (either for encryption or decryption), the ECG data comes from the immediately precedent L_o seconds. The only exception happens if the user tries to encrypt information in the very first L_o seconds of operation of EbH, right after the creation of $ECG_{Mod_{\mathcal{U}_i}}$. Disregarding this singular case, the user does not need to wait L_o seconds until the key is created. From the practical viewpoint, this is feasible since the smart device and the hub are assumed to be carried together.

[4] http://download.intel.com/newsroom/kits/40thanniversary/pdfs/Intel_40th_infographic_sm.pdf, last access Dec. 2016.

The second issue that may have practical impact for immediacy is the time needed for data preprocessing. Recalling Section 4.2, there are several steps to be carried out—filtering, segmentation and feature extraction (WHT and mean calculation). Eq. (11) gives the expression for this time, by adding the contributions of each of the aforementioned steps.

$$T_{preproc} = T_{filter} + T_{segment} + T_{WHT} + T_{calcmean}. \quad (11)$$

For any state-of-the-art device, $T_{segment}$ and $T_{calcmean}$ can be considered negligible since they are basic file and mathematical operations. However, the remaining steps are relevant in terms of computation. For our experiments, both operations have been carried out using Matlab in a 16 Gb RAM 2.8 GHz Intel Core i7 computer. Although these settings slightly exceed the current capabilities of an average smartphone, we believe that this technology may be available in the short term—new smartphones are already equipped with powerful 64-bit quad-core processors and a good amount of RAM.[5] Under these settings, according to Eq. (11), in the worst case, e.g. $L_o = 1800$ s, $T_{preproc} = 81.1$ ms for every key. This time is not only affordable, but also could be done in parallel in the L_o seconds of gathering ECG data.

Once data is preprocessed, the own key computation has to be carried out. Therefore, the actual time taken to derive a key is formed by the preprocessing time and the key computation one (Eq. (12)). According to our experiments, $T_{compKey} = 0.23$ ms, thus resulting in an overall time $T_{key} = 81.33$ ms

$$T_{key} = T_{preproc} + T_{compKey}. \quad (12)$$

Last but not least, in order for EbH to be feasible it is necessary to verify if the storage space is reasonable for current devices. Considering the previous explanation on how ECG values are collected, L_o determines the upper limit of the storage needed in the hub to compute keys. Eq. (13) gives the expression for the required storage, where $sizeof(feat)$ gives the size in bytes of an ECG features vector. Note that the division L_o/L_a gives the actual amount of ECG samples needed to collect in the period L_o, but one additional ECG sample is needed to store $ECG_{Mod_{\mathcal{U}_i}}$.

$$Storage_{hub}(bytes) = \left(\frac{L_o}{L_a} + 1\right) \times N_{feat} \times sizeof(feat). \quad (13)$$

Considering our experiments, in the worst case (i.e., $L_o = 1800$ s.), the storage needed is $Storage_{hub} = (1800/180 + 1) \times 194 \times 4 \simeq 8.6$ kb. Clearly, this space is affordable for any regular-class smartphone being used as hub.

Guidelines for tuning EbH. In general terms, EbH benefits from the size of the training set. Our results show that most indicators improve significantly when the system is trained for 14.4 h (i.e., 60% of the dataset) as compared to when the training lasts for 4.8 h (i.e.,20% of the dataset). The only exception is min-entropy, which only improves for the bigger value of DT (DT = 0.05).

Apart from this general advice, in practice there are two dimensions that are usually confronted—security and usability [32]. In the following, we discuss how the system parameters should be chosen to prioritize each of these dimensions.

If security is the main concern, seeds must be as unique, long and unpredictable as possible. As it has been previously shown, there is no single setting that improves these three aspects at a time. However, a suitable balance among them can be found for bigger values of the training set and the period of observation L_o, combined with middle values of TM and DT. Under these settings, EbH produces a reasonable amount of unique seeds, of intermediate size and entropy.

[5] https://www.qualcomm.com/news/onq/2017/05/23/powered-snapdragon-835-htc-u11-takes-touch-next-level.

On the contrary, if ease of use is at stake, the user must face the lowest amount of issues when trying to access personal data. In this case, combining the smallest value of TM and the biggest one of DT leads to the best results in both the amount of attempts to decrypt and the probability of no decryption. Indeed, the user only needs 1.13 attempts to decrypt whereas the said probability is dramatically low (0.3%). Interestingly, the size of the observation period can be small (9 min in our experiments) while keeping both properties largely unaltered. The balance between both issues remain since the amount of attempts to decrypt would raise to 1.22 while the probability of no decryption would lower to 0.1%.

4.5.4. Additional considerations

For completeness, two considerations are included regarding our experiments. First, the dataset was acquired using the SpaceLab-Burdick digital Holter recorder (SpaceLab-Burdick, Inc., Deerfield, WI). Thus, ECG signals were recorded using three pseudo-orthogonal lead configuration (X, Y, Z). The results shown in this paper correspond to one of the leads. Once the lead is arbitrarily chosen, this must be used for all the encryption and decryption operations.

On the other hand, we have also carried out experiments without feature selection ($N_{feat} = 256$ instead of $N_{feat} = 194$). Our findings show that the results are affected to some extent by feature selection. When all features are considered, the amount of seeds produced by each user is very similar. Each seed is significantly larger than the one obtained with the same configuration (e.g. 95.83 vs. 51.33 symbols), also with better minimum entropy (82.51 vs. 33.5). The increase in the seed randomness is linear to the increase in the amount of features. Thus, the amount of features is almost tripled ($256/93 = 2.75$), in a very similar factor to the increase of entropy ($82.51/33.5 = 2.46$). However, this increase does not scale well when the optimal minimum entropy is taken into account. For 93 (resp. 256) features, the optimal minimum entropy is 55.99 (resp. 154.13). Therefore, in the selected features dataset the achieved entropy represented 59.83% of its optimum value. On the contrary, in the dataset with all features it only accounts for 53.53%. Considering all these issues, it can be concluded that using all features leads to greater and less predictable seeds, but the improvement is not linear with the increase in the amount of features.

4.5.5. Overall discussion

Results obtained for EbH show that the proposed approach is promising for seed generation. However, this general finding can be refined into specific remarks.

The system needs a representative number of ECG samples to build $ECG_{Mod_{\mathcal{U}_i}}$. A percentage of 60%, which is a commonly employed value, seems an appropriate value for the workability of the system. In practice, this is not a practical shortcoming, as the proposed setting (i.e. user porting a bracelet and a smartphone) does not impose any time-related restriction. Indeed, for our experiments we consider 60% of one day, i.e. around 16 h. Thus, in practice every user could be working with EbH after the first day of training.

Most settings lead to distinctive keys among users. This is quite beneficial, as every user has data that can only be decrypted by herself. Even in those situations in which two user seeds coincide, the population is big enough (i.e. 196 users) to render this kind of attack impractical. Recall that the attacker would need to steal the hub (i.e. the smartphone) and connect it to her own bracelet for this attack to be successful.

Time-invariance is reasonably achieved under some settings. The optimal outcome is to have all users to produce a unique seed. EbH does not achieve this goal, but it achieves a close one. Results have shown that a vast majority of users (up to 95.97%) may

Table 4
Comparison table of reviewed works.

	Time-invariant keys	Different keys among users	Keys difficult to reproduce	Biometric trait
EbH	√	√	√	ECG
[34]	×	√	√	Face
[35]	×	√	×	Face
[36]	×	√	×	Face
[37]	×	×	√	Iris
[38]	×	×	√	Iris, fingerprint
[39]	×	×	√	Fingerprint
[42]	×	×	√	Palm of the hand
[43]	×	√	√	Voice
[44]	×	×	√	Digital audio watermarking
[48]	×	√	√	Signatures[a]
[49]	×	×	√	Handwriting[a]
[50]	×	×	√	Fingerprint vein[a]
[51]	×	√	√	–
[45]	×	×	√	ECG
[46]	×	√	√	ECG
[47]	×	√	√	ECG, PPG

–Not specified.
[a] Just for the evaluation process but the approach is general.

produce unique keys. Moreover, the amount of attempts to decrypt is significantly low. Another critical remark is that the probability of no decryption (which is paramount to ensure acceptance among users) is in most cases non-representative (as it is lower than 3%).

In sum, the feasibility of the proposed mechanism is supported by the previous facts as long as the dataset size is representative in both amount of users (199 subjects) and timespan (24 h). It is worth noting that the used dataset belongs to one of biggest and publicly available databases. However, this system could be improved considering the following pair of issues. On the one hand, a bigger database with more long-term recordings, e.g., years, could be used. In this way, we could analyze the changes produced in a very long time period for encryption/decryption keys, and this then would help to identify when these keys should be updated. Then, having very long ECG recordings, data stream mining techniques could be applied as on similar problems [33]. On the other hand, a database with tagged activities would be desirable. It will help us to study when users have more chances to successfully encrypt and decrypt. For instance, it is expected that if you encrypt something just after being running, its decryption would be quite challenging.

5. Related work

Multiple biometric traits have been used for key generation. Face images have been extensively applied. Teoh et al. [34] generate keys from facehashes, that is hashes created from facial images. The same goal is addressed by [35]. Applying 3-D face images, Chen et al. [36] present an entropy-based method to create deterministic bit sequences. The creation of keys not just by face images but by the iris ones is quite well-known. Rathgeb et al. [37] propose the binary codification of the iris, creating iris codes where context-based information helps to detect reliable bits. In [38] authors go a step further applying the iris and a second biometric trait, the fingerprint. Extracted features are fused to create a 256-bit key constructed under the problem of large numbers factorization. By contrast [39] only considers fingerprints and applies distortion. Generated keys are used when previous ones are lost or stolen, thus being these latter ones cancellable.

Other proposals generate keys from assorted biometric traits such as creating hashes from the palm of the hand [40], using handwritten signatures [41], moving a pair of devices simultaneously trying to get the same key [42], analyzing voice signals [43] or even studying a method to generate a key for digital audio watermarking [44].

The use of cardiovascular signals and ECG in particular, for key generation is the main issue to consider herein. This research line has received limited research attention. An encryption algorithm based on chaotic functions is proposed in [45]. In this scheme a secret key is generated from ECG signals using the Lyapunov exponent's spectrum to extract signal features. The key is directly constructed from the signal but for decryption purposes a secure channel is required to send the secret key. In the context of body sensor networks (BSNs), Zhang et al. [46,47] propose a fast biometric approach to generate 128-bit keys from ECG signals for ensuring confidentiality and authentication in BSNs communications. Keys are generated from Inter-Pulse Intervals (IPIs) extracted from cardiovascular signals (i.e. ECG or PPG), such that each bit is generated comparing two IPIs. The main difference with our approach is that Zhang et al. proposal do not aim to generate time-invariant keys, which makes their approach to be unsuitable for the considered scenario.

But not all proposals focus on a particular biometric trait, some of them propose a general approach. Sheng et al. [48] apply a semi-supervised clustering algorithm to identity consistent and discriminative biometric signals which are later used to generate keys. Ballard et al. [49] propose the use of biometric samples and a key as input of their system to get a cryptographic key. The main drawback is that users need to include a password together with the biometric samples. M. Khalil-Hani et al. [50] propose the use of a fuzzy vault scheme in which a biometric trait together with a secret key is used for encryption purposes. Again, the main weakness in comparison with the proposed approach is that a secret key is involved in the process. In addition, some other proposals, though not directly focused on creating keys through biometric traits, use them to achieve a final goal. In this regard, [51] presents a scheme in which biometric traits are used to generate keys applied for authentication purposes.

Table 4 shows a comparison analysis with previous proposals. We consider whether keys are equal along time (time-invariant), different amount users (uniqueness) and difficult to reproduce (invulnerable). The type of managed biometric trait is also pointed out. Symbol √ means that a feature is considered and symbol × means the opposite. From the table is noticed that the generation of keys through biometric traits has been extensively explored. Nonetheless, as far as we are concerned, just the proposed approach achieves a time-invariant key generation ensuring the difference among users and being resilient to guessing attacks, that is, keys are difficult to reproduce by attackers. Indeed, the fact of creating keys invariant along time is a clear benefit in system in which a continuous variable, e.g. an ECG signal, is at stake.

Table A.5
EbH experimental results with 20% of samples for $ECG_{Mod_{\mathcal{U}_i}}$. AD, P_{ND} and amount of users producing fully time-invariant seeds.

		AD				P_{ND}				Time-invariant seeds			
		TM											
	DT	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**
$L_o = 3 \cdot L_a$	**0.005**	11.26	62.66	85.01	89.05	0.006	0.026	0.049	0.092	134	39	10	1
	0.01	10.97	61.29	84.35	89.19	0.005	0.024	0.045	0.084	135	41	11	1
	0.03	8.08	56.51	78.35	90.26	0.004	0.018	0.034	0.065	145	48	19	2
	0.05	5.05	44.81	71.11	85.26	0.003	0.013	0.025	0.049	158	65	28	9
$L_o = 10 \cdot L_a$	**0.005**	4.38	34.17	61.07	75.25	0.008	0.037	0.071	0.139	161	80	37	14
	0.01	4.03	30.99	57.11	74.46	0.008	0.033	0.064	0.128	163	86	43	16
	0.03	2.11	20.08	44.02	68.12	0.005	0.025	0.048	0.095	177	109	63	27
	0.05	1.57	13.86	35.30	59.96	0.003	0.019	0.037	0.074	183	125	78	38

Table A.6
EbH experimental results with 60% of samples for $ECG_{Mod_{\mathcal{U}_i}}$. AD, P_{ND} and amount of users producing fully time-invariant seeds.

		AD				P_{ND}				Time-invariant seeds			
		TM											
	DT	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**
$L_o = 3 \cdot L_a$	**0.005**	1.66	9.88	24.83	52.03	0.002	0.009	0.018	0.035	182	138	99	52
	0.01	1.44	9.03	24.00	49.45	0.002	0.008	0.016	0.031	185	141	101	56
	0.03	1.22	5.60	16.38	39.45	0.001	0.005	0.012	0.024	189	155	119	72
	0.05	1.22	4.24	12.73	30.35	0.001	0.004	0.009	0.019	189	162	129	88
$L_o = 10 \cdot L_a$	**0.005**	1.25	6.24	13.23	32.31	0.004	0.019	0.035	0.067	188	151	125	80
	0.01	1.25	6.02	11.65	30.46	0.004	0.018	0.032	0.062	188	152	130	84
	0.03	1.13	4.99	9.60	25.28	0.003	0.016	0.026	0.050	191	157	137	95
	0.05	1.13	3.91	7.00	17.49	0.003	0.013	0.020	0.039	191	163	147	113

Table A.7
EbH experimental results with 20% of samples for $ECG_{Mod_{\mathcal{U}_i}}$. Amount of seeds, seed size (in number of symbols) and min-entropy.

		Amount of seeds				Seed size (symbols)				$H_\infty(EbH)$			
		TM											
	DT	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**
$L_o = 3 \cdot L_a$	**0.005**	306	714	1164	2009	164.67	159.97	160.09	161.59	33.30	40.26	40.44	38.74
	0.01	296	663	1075	1858	150.37	145.19	145.31	146.52	51.68	60.08	60.27	58.81
	0.03	273	546	873	1485	112.84	112.13	112.22	113.67	95.74	96.82	97.03	96.45
	0.05	251	448	697	1173	90.79	95.96	97.61	98.89	96.54	98.80	99.16	98.70
$L_o = 10 \cdot L_a$	**0.005**	244	405	602	985	170.84	167.33	166.55	166.47	25.49	30.48	31.86	32.37
	0.01	241	385	564	920	155.26	152.97	152.23	152.31	44.74	48.60	50.08	50.44
	0.03	226	339	472	741	115.50	116.48	117.18	117.93	94.33	94.87	94.32	94.08
	0.05	217	307	406	618	89.73	94.40	96.99	99.01	95.89	98.47	98.55	98.34

Table A.8
EbH experimental results with 60% of samples for $ECG_{Mod_{\mathcal{U}_i}}$. Amount of seeds, seed size (in number of symbols) and min-entropy. Values marked with (*) means that seeds are repeated among users.

		Amount of seeds				Seed size (symbols)				$H_\infty(EbH)$			
		TM											
	DT	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**	**0.01**	**0.05**	**0.1**	**0.2**
$L_o = 3 \cdot L_a$	**0.005**	219*	285*	369*	535*	173.15	169.49	168.95	167.18	22.61	27.42	28.45	31.04
	0.01	216	272	349	495	157.34	155.69	155.34	153.62	41.79	44.44	45.36	48.09
	0.03	211	252	310	429	117.51	118.08	118.60	117.79	93.51	94.30	94.21	94.41
	0.05	210	241	289	381	93.09	96.19	97.86	99.66	98.83	99.73	99.99	100.09
$L_o = 10 \cdot L_a$	**0.005**	211*	252*	298*	387*	175.44	173.33	170.70	171.67	19.92	22.69	26.07	25.32
	0.01	211	250	290	373	160.63	158.10	156.13	157.40	37.55	41.14	43.99	42.76
	0.03	208	244	272	339	120.68	118.77	118.94	121.60	91.38	92.98	92.81	91.13
	0.05	207	235	257	308	95.87	96.00	97.85	101.57	99.05	99.01	99.24	99.87

6. Conclusion. Future work

The use of ElectroCardioGram (ECG) biosignals have already been applied into the cryptographic arena. Thus, previous efforts have explored its application to protect health information or to authenticate users.

This paper has addressed the potential of ECG to derive time-invariant encryption keys. Thanks to this property, it is possible for an external data storage (e.g. a smartphones) to symmetrically encrypt its data, allowing the user to decrypt it at any time just by deriving the key from its current ECG value. The proposed mechanism, EbH, has been shown to provide with keys featuring high degree of time-invariance over a 24 h period. Remarkably, these keys are different for every user (in a set of 199 individuals) and they have suitable levels of min-Entropy and length so as to resist guessing attacks.

Future research works will be focused on expanding this approach to other biosignals (such as PPG) and exploring other settings that may allow smaller training times. Furthermore, another aspect that deserves attention is how to handle data re-encryption.

In particular, it is convenient that EbH seeds may become different (but time-invariant for a period) for a given user. Such periodical key renewal needs to be researched. The extension of the evaluation considering large-scale datasets (i.e., several months/years of a significant amount of users) is also interesting to ensure the goal assessment in long periods. Finally, the application of data stream mining techniques is also relevant to enable EbH handling changes (concept drift study) in ECG signals throughout the subject's lifetime also considering performed activities.

Acknowledgments

Funding: This work was supported by the MINECO grants TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You) and TIN2016-79095-C2-2-R (SMOG-DEV); by the CAM grant S2013/ICE-3095 (CIBERDINE: Cybersecurity, Data, and Risks), which is co-funded by European Funds (FEDER); and by the Programa de Ayudas para la Movilidad of Carlos III University of Madrid, Spain (J. M. de Fuentes and L. Gonzalez-Manzano grants).

Data used for this research was provided by the Telemetric and ECG Warehouse (THEW) of University of Rochester, NY.

Authors would also like to thank the anonymous reviewers for their comments.

Appendix

See Tables A.5–A.8

References

[1] H. Liu, H. Ning, Y. Yue, Y. Wan, L.T. Yang, Selective disclosure and yoking-proof based privacy-preserving authentication scheme for cloud assisted wearable devices, Future Gener. Comput. Syst. (2017).
[2] J. Lage, A.P. Catarino, H. Carvalho, A. Rocha, Smart shirt with embedded vital sign and moisture sensing, in: SPWID 2015: The First International Conference on Smart Portable, Wearable, Implantable and Disability-Oriented Devices and Systems, Iaria, 2015, pp. 25–30.
[3] A. Pyattaev, K. Johnsson, S. Andreev, Y. Koucheryavy, Communication challenges in high-density deployments of wearable wireless devices, IEEE Wirel. Commun. 22 (1) (2015) 12–18.
[4] A.M. Rahmani, T.N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach, Future Gener. Comput. Syst. (2017).
[5] A.A.S. Ali, X. Zhai, A. Amira, F. Bensaali, N. Ramzan, Heterogeneous implementation of ECG encryption and identification on the Zynq SoC, in: 24th Annual International Symposium on Field-Programmable Custom Computing Machines, (FCCM), IEEE, 2016 139–139.
[6] F. Sufi, S. Mahmoud, I. Khalil, A wavelet based secured ECG distribution technique for patient centric cpproach, in: 5th International Summer School and Symposium on Medical Devices and Biosensors, IEEE, 2008, pp. 301–304.
[7] J.S. Arteaga-Falconi, H.A. Osman, A.E. Saddik, ECG authentication for mobile devices, IEEE Trans. Instrum. Meas. 65 (3) (2016) 591–600.
[8] S.J. Kang, S.Y. Lee, H.I. Cho, H. Park, ECG authentication system design based on signal analysis in mobile and wearable devices, IEEE Signal Process. Lett. 23 (6) (2016) 805–808.
[9] R. Kang, L. Dabbish, N. Fruchter, S. Kiesler, ?my data just goes everywhere:? user mental models of the internet and implications for privacy and security, in: Eleventh Symposium on Usable Privacy and Security, SOUPS 2015, 2015, pp. 39–52.
[10] E. Chin, A.P. Felt, V. Sekar, D. Wagner, Measuring user confidence in smartphone security and privacy, in: Proceedings of the Eighth Symposium on Usable Privacy and Security, ACM, 2012, p. 1.
[11] B. Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C, first ed., John Wiley & Sons, Inc., 2015.
[12] E. Barker, W. Barker, W. Burr, M. Smid, NIST SP800-57 Recommendation for Key Management, Part 1: General, NIST, 2012.
[13] E. Okoh, A.I. Awad, Biometrics applications in e-health security: A preliminary survey, in: International Conference on Health Information Science, Springer, 2015, pp. 92–103.
[14] M. Rostami, A. Juels, F. Koushanfar, Heart-to-heart (H2H): Authentication for implanted medical devices, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS'13, ACM, 2013, pp. 1099–1112.
[15] R. Seepers, C. Strydis, I. Sourdis, C.D. Zeeuw, Enhancing heart-beat-based security for mhealth applications, IEEE J. Biomed. Health Inform. 21 (1) (2016) 254–262.
[16] C. Adams, G. Kramer, S. Mister, R. Zuccherato, On the security of key derivation functions, in: K. Zhang, Y. Zheng (Eds.), Information Security: 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27–29, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 134–145. http://dx.doi.org/10.1007/978-3-540-30144-8. URL http://dx.doi.org/10.1007/978-3-540-30144-8_12.
[17] E. Barker, J. Kelsey, Recommendation for the Entropy Sources Used for Random Bit Generation, NIST DRAFT Special Publication 800-90B, 2016.
[18] U. Uludag, S. Pankanti, S. Prabhakar, A.K. Jain, Biometric cryptosystems: issues and challenges, Proc. IEEE 92 (6) (2004) 948–960.
[19] J.H. Kong, L.-M. Ang, K.P. Seng, A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments, J. Netw. Comput. Appl. 49 (2015) 15–50.
[20] C. Camara, P. Peris-Lopez, J.E. Tapiador, Human identification using compressed ECG signals, J. Med. Syst. 39 (11) (2015) 148.
[21] E.B. Barker, J.M. Kelsey, SP 800-90A. Recommendation for random number generation using deterministic random bit generators, Tech. rep., 2012, National Institute of Standards & Technology.
[22] Y. Gahi, M. Lamrani, A. Zoglat, M. Guennoun, B. Kapralos, K. El-Khatib, Biometric identification system based on electrocardiogram data, in: Int. Conference on New Technologies, Mobility and Security, NTMS, 2008, pp. 1–5.
[23] R.D. Labati, R. Sassi, F. Scotti, ECG biometric recognition: Permanence analysis of QRS signals for 24 hours continuous authenticationsafie2011electrocardiogram, in: 2013 IEEE International Workshop on Information Forensics and Security, (WIFS), IEEE, 2013, pp. 31–36.
[24] F. Agrafioti, D. Hatzinakos, ECG based recognition using second order statistics, in: 6th Annual Conference on Communication Networks and Services Research, CNSR, 2008, pp. 82–87.
[25] M. Hejazi, S. Al-Haddad, Y.P. Singh, S.J. Hashim, A.F.A. Aziz, ECG biometric authentication based on non-fiducial approach using kernel methods, Digit. Signal Process. 52 (2016) 72–86.
[26] I. Odinaka, L. Po-Hsiang, A.D. Kaplan, J.A. O'Sullivan, E.J. Sirevaag, J.W. Rohrbaugh, ECG biometric recognition: A comparative analysis, IEEE Trans. Inf. Forensics Secur. 7 (6) (2012) 1812–1824.
[27] T. Beer, Walsh transforms, Amer. J. Phys. 49 (5) (1981) 466–472.
[28] W.S. Kuklinski, Fast Walsh transform data-compression algorithm: E.c.g. applications, Med. Biol. Eng. Comput. 21 (4) (1983) 465–472. http://dx.doi.org/10.1007/BF02442635.
[29] D. Venugopal, S. Mohan, S. Raja, An efficient block based lossless compression of medical images, Optik-Int. J. Light Electron Opt. 127 (2) (2016) 754–758. http://dx.doi.org/10.1016/j.ijleo.2015.10.154.
[30] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., in: Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
[31] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, third ed., Morgan Kaufmann Publishers Inc., 2011.
[32] L.F. Cranor, S. Garfinkel, Security and Usability: Designing Secure Systems that People can Use, O'Reilly Media, Inc., 2005.
[33] M.M. Gaber, J. Gama, S. Krishnaswamy, J.B. Gomes, F. Stahl, Data stream mining in ubiquitous environments: state-of-the-art and current directions, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 4 (2) (2014) 116–138.
[34] A.B. Teoh, D.C. Ngo, A. Goh, Personalised cryptographic key generation based on FaceHashing, Comput. Secur. 23 (7) (2004) 606–614.
[35] Y. Wang, K. Plataniotis, Fuzzy vault for face based cryptographic key generation, in: Biometrics Symposium, 2007, IEEE, 2007, pp. 1–6.
[36] B. Chen, V. Chandran, Biometric based cryptographic key generation from faces, in: 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, IEEE, 2007, pp. 394–401.
[37] C. Rathgeb, A. Uhl, Context-based biometric key generation for Iris, IET Comput. Vis. 5 (6) (2011) 389–397.
[38] A. Jagadeesan, T. Thillaikkarasi, K. Duraiswamy, Cryptographic key generation from multiple biometric modalities: Fusing minutiae with iris feature, Int. J. Comput. Appl. 2 (6) (2010) 16–26.
[39] S.V. Gaddam, M. Lal, Efficient cancelable biometric key generation scheme for cryptography, IJ Netw. Secur. 11 (2) (2010) 61–69.
[40] T. Connie, A. Teoh, M. Goh, D. Ngo, PalmHashing: a novel approach for cancelable biometrics, Inform. Process. Lett. 93 (1) (2005) 1–5.
[41] M. Freire-Santos, J. Fierrez-Aguilar, J. Ortega-Garcia, Cryptographic key generation using handwritten signature, in: Proc. SPIE, Vol. 6202, 2006, pp. 225-231.
[42] H. Yüzugüzel, J. Niemi, S. Kiranyaz, M. Gabbouj, T. Heinz, ShakeMe: Key generation from shared motion, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, (CIT/IUCC/DASC/PICOM), IEEE, 2015, pp. 2130–2133.
[43] F. Monrose, M.K. Reiter, Q. Li, S. Wetzel, Using voice to generate cryptographic keys, in: 2001: A Speaker Odyssey-the Speaker Recognition Workshop, 2001.

[44] M.K. Dutta, P. Gupta, V.K. Pathak, Biometric based unique key generation for authentic audio watermarking, in: International Conference on Pattern Recognition and Machine Intelligence, Springer, 2009, pp. 458–463.
[45] C.-K. Chen, C.-L. Lin, C.-T. Chiang, S.-L. Lin, Personalized information encryption using ECG signals with chaotic functions, Inform. Sci. 193 (2012) 125–140.
[46] G. Zhang, C.C. Poon, Y. Zhang, A fast key generation method based on dynamic biometrics to secure wireless body sensor networks for p-health, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, IEEE, 2010, pp. 2034–2036.
[47] C.C. Poon, Y.-T. Zhang, S.-D. Bao, A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health, IEEE Commun. Mag. 44 (4) (2006) 73–81.
[48] W. Sheng, S. Chen, G. Xiao, J. Mao, Y. Zheng, A biometric key generation method based on semisupervised data clustering, IEEE Trans. Syst. Man Cybern.: Syst. 45 (9) (2015) 1205–1217.
[49] L. Ballard, S. Kamara, F. Monrose, M.K. Reiter, Towards practical biometric key generation with randomized biometric templates, in: Proceedings of the 15th ACM Conference on Computer and Communications Security, ACM, 2008, pp. 235–244.
[50] M. Khalil-Hani, M.N. Marsono, R. Bakhteri, Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm, Future Gener. Comput. Syst. 29 (3) (2013) 800–810.
[51] S. Kumari, X. Li, F. Wu, A.K. Das, K.-K.R. Choo, J. Shen, Design of a provably secure biometrics-based multi-cloud-server authentication scheme, Future Gener. Comput. Syst. 68 (2017) 320–330.

L. González-Manzano is assistant professor working in the Computer Security Lab at the University Carlos III of Madrid, Spain. She is Computer Scientist Engineer and Ph.D. in Computer Science by the University Carlos III of Madrid. Her Ph.D. focuses on security and privacy in social networks. She is currently focused on Internet of Things and cloud computing security, as well as, on cybersecurity. Indeed, she has published several papers in national and international conferences and journals and she is also involved in national R+D projects.

José M. de Fuentes is visiting lecturer in the Computer Science and Engineering Department at University Carlos III of Madrid, Spain. He is Computer Scientist Engineer and Ph.D. in Computer Science by the University Carlos III of Madrid. His main research interests are cybersecurity as well as security and privacy in the internet of things and ad-hoc networks. He has published several articles in international conferences and journals. He is participating in several national R+D projects.

P. Peris-Lopez is Visiting Lecturer at the Department of Computer Science, Universidad Carlos III de Madrid, Spain. He holds a M.Sc. in Telecommunications Engineering and Ph.D. in Computer Science. His research interests are in the field of protocols design, primitives design, lightweight cryptography, cryptanalysis etc. Nowadays, his research is focused on Radio Frequency Identification Systems (RFID) and Implantable Medical Devices (IMD). In these fields, he has published a great number of papers in specialized journals and conference proceedings. For additional information see: www.lightweightcryptography.com.

C. Camara is Ph.D. student working in the Computer Security Lab at the University Carlos III of Madrid, Spain. She holds a M.Sc. in Computer Science and Technology, with specialization in Artificial Intelligence (Carlos III University of Madrid) and a M.Sc. in Biomedical Engineering (Technical University of Madrid). Her research interests are in the fields of applied cryptography and security for implantable medical devices and biomedical signal processing.