H2K: A Heartbeat-based Key Generation Framework for ECG and PPG Signals

Junqing Zhang, Yushi Zheng, Weitao Xu, *Member, IEEE,* and Yingying Chen, *Fellow, IEEE*

Abstract—Wireless body area network is a key enabler for connected healthcare but recent cyberattacks have compromised its security and trustworthiness. This paper investigates heartbeat-based key generation to secure body area networks. The interpulse intervals (IPIs) between any two adjacent peaks of heartbeat signals are random and state-of-the-art literature has demonstrated that IPI is a good random source to be extracted as cryptographic keys. Heartbeat signals can be measured by electrocardiography (ECG) and photoplethysmography (PPG) sensors. A general heartbeat-based key generation framework applicable to both ECG and PPG signals is proposed. A robust peak detection algorithm is designed to capture noisy peaks and a simple yet efficient IPI alignment algorithm to align the common IPIs. A key establishment protocol is used to convert analog IPIs to digital binaries and reconcile them between legitimate devices. We evaluate the performance for both ECG signals from an online public database, MIT PhysioBank, and PPG signals collected from our testbed. The results demonstrate that our algorithm is robust and heartbeat-based key generation can be completed for both ECG and PPG signals. We finally create a PPG-based prototype and a demonstration video to show the practicality of our framework.

Index Terms—Body area networks, key generation, biometrics, electrocardiography (ECG), photoplethysmography (PPG), interpulse interval

1 Introduction

Wireless body area network (WBAN) has become an important part of the Internet of Things (IoT) as connected healthcare becomes prevalent [1], [2]. There are abundant commercial off-the-shelf (COTS) wearable devices such as Fitbits and smart watches as well as medical implantable devices, e.g., pacemakers. Wireless communications are preferred for these devices because they are free from constraints of cables. Wireless connections are essential for implantable devices, e.g., a pacemaker can be wirelessly configured and medical data in the pacemaker can be transmitted by using a device programmer.

The security and trustworthiness of the WBAN have become a big issue [3]. Numerous implantable and wearable devices transmit vital and/or private signals, such as health information or control signals. Wireless communications are broadcast and freely accessible to anyone within the communication range but the security countermeasures for healthcare devices are rather limited. This concern has been evidenced by e.g. a demonstration that ten types of implantable cardioverter defibrillators had no or limited security primitives and could be accessed by a common wireless platform [4]. Medical data stored inside them could be breached by malicious entities. It would be catastrophic if these weak devices had been attacked by hackers.

Cryptographic schemes are commonly used to protect wireless transmissions, consisted of symmetric encryption and key distribution. Symmetric encryption, such as Advanced Encryption Standard (AES), is used to protect connections between two legitimate devices, named Alice and Bob. There have been hardware and software efficient implementations of AES hence it is suitable for embedded devices. For example, an AES coprocessor is included in the ZigBee chip, TI cc2531[1]. On the other hand, secure and lightweight key distribution is challenging. Symmetric key is required for encryption and decryption at Alice and Bob, which is usually completed by public key cryptography (PKC) primitives [5]. However, PKC tends to be computationally expensive and thus it may not be applicable to many medical devices. For instance, implantable devices are limited in size and powered by a battery, which will not have sufficient computational and energy resources.

The above facts motivate researchers to design secure and lightweight alternatives for distributing cryptographic keys for wearable and medical devices. Heartbeat signals are found as a good candidate. To the best knowledge of the authors, the concept of exploiting physiological signals for secure communications can date back to 2003 [6]. Poon *et al.* extracted cryptographic keys from the interpulse intervals (IPIs) of heartbeats [7], which are the time intervals between any two adjacent peaks of heartbeats. Heartbeat signals have the following characteristics that make IPIs ideal to be explored as cryptographic keys [7]:

- *Reciprocity*. The heartbeat rate information measured

[1]https://www.ti.com/lit/ds/symlink/cc2531.pdf

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. The work of J. Zhang was supported in part by the UK EPSRC New Investigator Award and in part by the National Key Research and Development Program of China under grant ID 2020YFE0200600. The review of this paper was coordinated by xxx. (Corresponding author: Junqing Zhang.)

- *J. Zhang and Y. Zheng are with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, U.K. (emails: Junqing.Zhang@liverpool.ac.uk; Y.Zheng23@student.liverpool.ac.uk)*
- *W. Xu is with Department of Computer Science, City University of Hong Kong, Hong Kong SAR China. (email: weitaoxu@cityu.edu.hk)*
- *Y. Chen is with Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, US. (email: yingche@scarletmail.rutgers.edu)*

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier xxx

from different parts of the same person will be the same as they are all rooted in the contraction and relaxation of the cardiac muscle.
- *Uniqueness*. The patterns of heartbeats between different individuals are not identical [8].
- *Randomness*. The IPIs are found to be random [9].

In addition, Xu *et al.* has demonstrated that this technique is lightweight and suitable for devices that have the capability of measuring heartbeats [10]. A summary of the state-of-the-art literature is given in Table 1.

IPI can be extracted from electrocardiogram (ECG) and photoplethysmography (PPG) signals. Most of the work uses ECG signals from public databases, e.g., MIT PhysioBank databases [14], to evaluate their protocols [10], [12], [15]–[18]. There are also some research efforts using real ECG sensors [8], [11], [19], [20]. However, most of the ECG sensors require direct contact to the skin and will be difficult for daily use. PPG sensors can also measure heartbeats. However, employing PPG signals to extract IPI for key generation has not been reported yet[2]. PPG sensors are widely used in wearable devices such as Fitbit, which are very convenient to use. This paper extends IPI-based key generation for PPG sensors.

The IPI extraction is the most important step as the IPI is the random source. The extraction is quite challenging because heartbeat-related signals are non-stationary and noisy. Most literature did not reveal sufficient technical details on the IPI extraction. Wavelet analysis is usually used to process ECG signals to eliminate the amplitude baseline drift and noise [16], [17]. After the wavelet processing, it is still difficult to extract the peaks as they are swamped by numerous void peaks. A hard threshold detection based on the peak amplitude cannot extract IPIs accurately because the amplitudes of ECG peaks vary from time to time on the same person and are also affected by setups of different sensors. A local peak detection algorithm is used in [13]. However, it is difficult to specify the length of the local detection window as the heartbeat rates vary among different people. A normal sinus rhythm of an adult is 60 to 100 heartbeats per minute. A robust and accurate IPI extraction algorithm is thus urgently required. This paper addresses the challenge by using wavelet transform and clustering algorithm.

The generated key can be used for encryption and decryption, which involves at least two devices having the same key. However, most work only considers the key generation performance with one sensor [16], [17]; whether two devices will generate the same key remains unknown. Different from [16], [17], this paper further exploits the key generation for encryption/decryption between two devices, which involves them extracting IPIs separately and reconciling on the common key. The investigation of IPI alignment and key disagreement rate (KDR) analysis is rather limited [12], [13], [22], [23]. KDR quantifies the similarities of keys of two devices. IPI misalignment occurs when the IPI detection algorithm may miss the correct peaks or detect the wrong peaks. Seepers *et al.* designed a heartbeat misdetection algorithm [12], which divided the IPIs into different blocks and removed the entire block once an anomaly IPI was detected. Choi *et al.* took a step further by proposing a self-recovery procedure when peak detection failed or a fake peak was detected [22]. However, it was based on the assumption that peak misdetection only happened once over the entire process, which may not be realistic. Kim *et al.* also designed a peak misdetection recovery algorithm [23] but their algorithm needed a random number generator which may not be available. The misalignment will result in different measured IPIs at legitimate devices and thereof failure of key generation. A robust IPI alignment algorithm is still missing but urgently required. Even after the IPI alignment is implemented, there will still be differences between the measured IPIs due to the sampling noise, which will result in key disagreement between legitimate devices. There is very limited research on the KDR [13] and more investigation is required. The research on IPI alignment and KDR will be essential for generating identical keys at both parties. An efficient IPI alignment is designed in this paper by indexing each IPI and aligning them based on the common index.

This paper addresses the aforementioned challenges by designing a practical heartbeat-based key generation framework to secure wearable and implantable devices. Specifically, we considered a practical setup that two devices only exchange data for assisting IPI alignment and key agreement using wireless communications. The timestamps and IPI information is processed at each device to avoid leaking keying information. We evaluated the framework using both ECG and PPG signals to demonstrate the framework is generic and applicable. We used ECG data from the MIT PhysioBank database and simulated two virtual devices running the designed key generation framework. We built a testbed to collect PPG signals and created a prototype demonstration to show the framework is working successfully. Our contributions are listed as follows.

- A complete IPI-based key generation protocol is proposed, which includes heartbeat measurement, IPI extraction, and key establishment.
- A robust and generic IPI extraction scheme is designed. We use wavelet transform to denoise the captured heartbeat signals and a clustering algorithm to obtain the IPIs. We design a simple yet efficient IPI alignment algorithm to enable Alice and Bob agree on the same IPIs. The scheme is applicable to both ECG and PPG signals.
- The performance of the proposed solution has been extensively evaluated using ECG data from the widely used MIT PhysioBank databases. Our algorithm is stable as it works successfully with ECG signals lasting more than 20 hours. We evaluated the KDR and key generation rate (KGR) to show the effectiveness of our protocol.
- The performance is also evaluated using PPG signals measured from our customized testbed. It is composed of PPG sensors, AD conversion (Arduino) and a PC for signal processing, which provides full access to the raw waveform of PPG signals. We also create a PPG-based prototype that consists of PPG sensors, Arduino and Raspberry Pi as the processing

[2]The work in [21] used the frequency domain feature of PPG signals for key generation. They did not investigate the IPI of PPG signals.

TABLE 1
A Summary of Literature

Paper	Random Extraction	Contribution	Data
Rostami *et al.* [11]	IPI (four LSB)	A secure pairing protocol for implantable devices ARM implementation	ECG sensors
Seepers *et al.* [12]	IPI (Bits 5 to 7)	A full fuzzy commitment-based protocol; Peak misalignment detection	MIT PhysioBank
Lin *et al.* [13]	IPI (Bits 4 to 6)	A full key generation protocol; Evaluating sampling frequency	Piezo vibration sensors
Chizari *et al.* [9]	IPI trend	Extensive randomness evaluation on a huge dataset (with 900,000,000 IPIs)	MIT PhysioBank

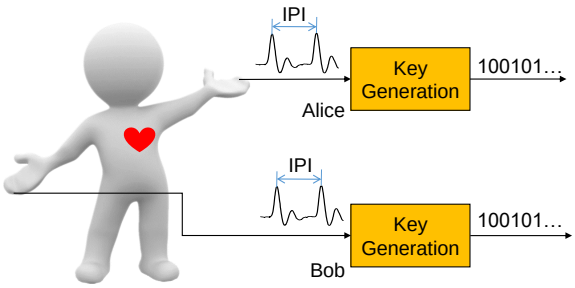


Fig. 1. System diagram. Two sensors are attached to the human body and measure the heartbeat signals. The IPIs of the heartbeat signals are extracted as cryptographic keys.

platform. A demo video has been produced to show a successful working system.

The rest of the paper is organized as follows. Section 2 briefly introduces the system overview. Section 3 presents the IPI extraction algorithm and Section 4 designs the key establishment protocol. Section 5 evaluates the performance of the proposed solution using the ECG signals from the MIT PhysioBank databases. In Section 6 we present a PPG-based testbed designs, experimental validation as well as a prototype. The related work is introduced in Section 8. Section 9 concludes the paper.

2 System Overview

The system diagram is portrayed in Fig. 1, where Alice and Bob represent two legitimate sensors attached to the human body and measure heartbeats. The key generation protocol is shown in Fig. 2, composed of three stages, namely heartbeat measurement, IPI extraction, and key establishment.

Any sensor that can capture heartbeat signals will work. This paper considers common ECG and PPG sensors. ECG sensors detect the electrical activities of the heart. ECG measurements usually require electrodes in direct contact with the skin, but there are also wearable ECG sensors, e.g., embedded in Apple Watch. PPG sensors consist of a light emitter and a detector. An LED will illuminate the tissue, which will be reflected by the blood. As the blood volume changes during a cardiac cycle, the detector will detect the variation and thereof the heartbeat [24]. PPG sensors are widely used in consumer electronics such as Fitbits. Snippets of ECG and PPG signals are shown in Fig. 3(a) and Figs 3(b), respectively.

Once the heartbeat signals are captured by sensors, the peaks of the ECG or PPG signals will be first located

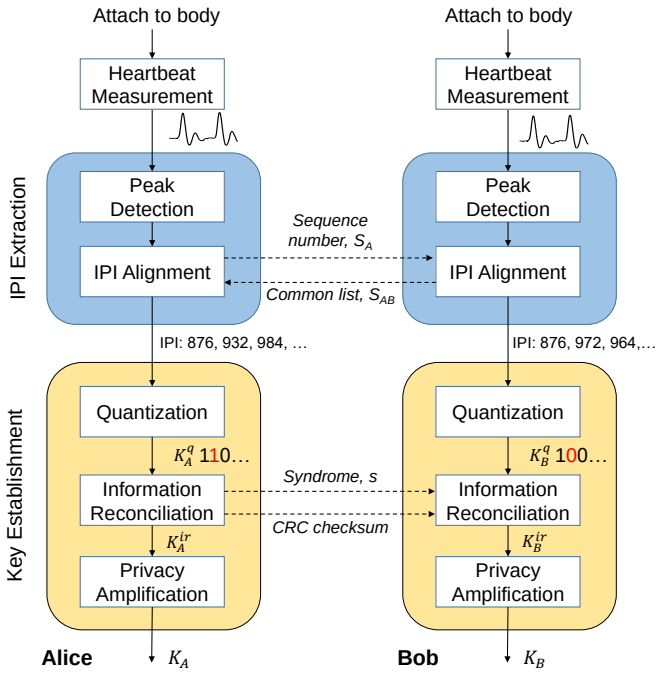


Fig. 2. Heartbeat-based key generation framework. IPI values are in the unit of millisecond. The dashed lines represent wireless transmissions.

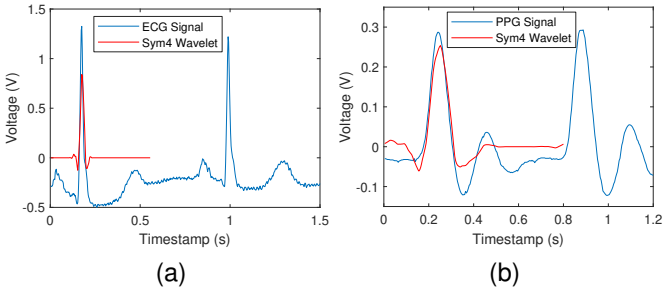


Fig. 3. (a) ECG signal from MIT PhysioBank database and Sym4 wavelet scale 3. (b) PPG signal measured by a pulse sensor and Sym4 wavelet scale 5.

by the peak detection algorithm. Alice and Bob will then individually calculate the IPI. There will be misdetected or missed peaks, IPI alignment is adopted to enable Alice and Bob to agree on peaks with common indexes. Peak detection and IPI alignment will be introduced in Section 3.

IPIs are analog values and should be converted to digital binary sequences, which is completed by the key establishment protocol, consisting of quantization, information

reconciliation, and privacy amplification. The protocol will be explained in Section 4.

Information exchange is required for the above process. WBAN devices have wireless communications modules, such as Bluetooth, ZigBee and WiFi. Bluetooth uses adaptive frequency hopping, which is suitable in high interference environments [25]. As this paper does not consider wireless interference, WiFi is used as an example.

3 IPI Extraction

IPI extraction is the most important step as it extracts entropy from heartbeat signals. However, many literature treated this step very trivially, which simply indicated that wavelet analysis was used but a detailed explanation and results are missing. In addition, detecting peaks and aligning them is also essential. It is necessary to study IPI extraction in a more comprehensive manner and reveal more technical details, which shall be beneficial for the community.

Some examples of ECG and PPG signals are given in Fig. 4(a) and Fig. 5, respectively. Same as most literature, this paper used ECG signals from the MIT PhysioBank database. Regarding PPG signals, we designed a pulse sensor-based testbed, which will be introduced in Section 6. The heartbeat signals captured by the sensor u are denoted as $x_u(t)$.

3.1 Peak Detection

IPI extraction relies on detecting the pulse peaks and calculating the intervals between any adjacent peaks. As shown in Fig. 3, both ECG and PPG signals have pulse peaks and their IPI can be given as

$$\Delta p_u(i) = t(i+1) - t(i), \tag{1}$$

where $t(i)$ is the timestamp of the i^{th} peak. Regarding ECG signals, the IPI is also referred as R-R interval.

An intuitive idea is to detect the peaks based on their amplitudes, which is not robust. As can be observed in Fig. 4(a) and Fig. 5, the peaks of the ECG and PPG signals varied significantly in amplitudes. In addition, the peak amplitudes will also vary based on the measuring devices and the heartbeat signals, hence a hard threshold will not work. A robust and adaptive peak detection algorithm is thus strongly required. We designed a three-step algorithm, consisted of wavelet analysis, peak elimination and peak refinement. The pseudo code is given in Algorithm 1.

3.1.1 Wavelet Analysis

Wavelet analysis has been proved efficient to remove the noise in ECG signals [16]. As shown in Fig. 3(a) and Fig. 3(b), the sym4 wavelet resembles for both peaks of the ECG and PPG signals. Different from other papers, we demonstrated the same sym4 wavelet can be used for both ECG and PPG signals, which makes our algorithm generic.

Multiple level wavelet decomposition can be applied to the heartbeat signals, $x_u(t)$. Drift of signal amplitudes are low frequency parts. We can deliberately remove these parts and the reconstructed signal is denoted as $x_u^w(t)$, which is illustrated in Fig. 4(b). The effect for the PPG signals is similar hence the results are not shown here for brevity. From now on we use ECG signals to explain our algorithm.

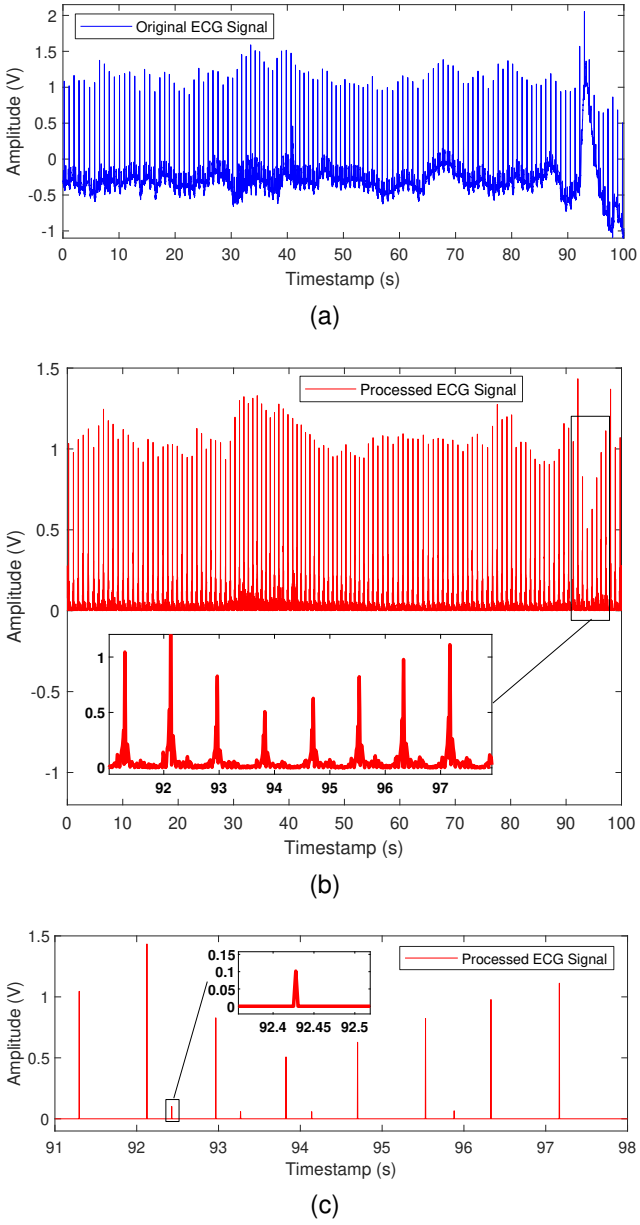


Fig. 4. (a) A snapshot ECG signal from MIT PhysioBank database. Dataset: mitdb/101. (b) ECG signal processed by wavelet analysis. (c) ECG signal processed after peak elimination.

3.1.2 Peak Elimination

As shown in Fig. 4(b), there are many other peaks in ECG signals, which should be eliminated to detect the correct peaks. We designed a window peak elimination algorithm. The signal, $x_u^w(t)$, is partitioned into multiple segments, each with 300 ms. The window length is selected based on the fact that the heartbeat interval of any people alive will be larger than 300 ms, otherwise, the person will be in danger to life. This can ensure that only one (valid or void) peak in each segment but will not accidentally remove any valid peaks. We only kept the peaks and set the values of the rest of the points inside the segment as zero. The detected peak index is $t'(k)$.

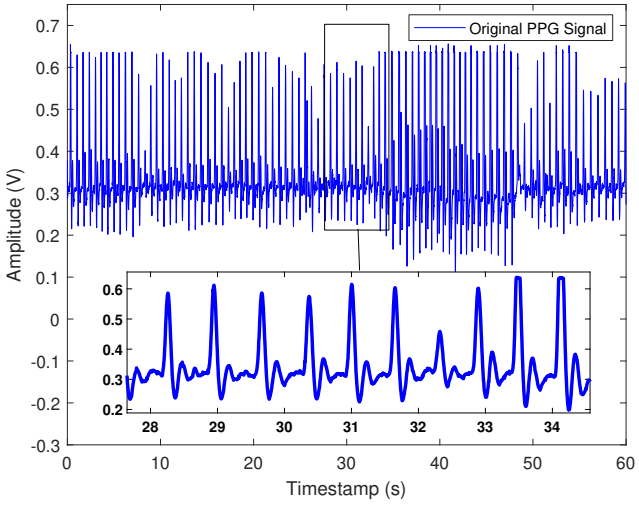


Fig. 5. A snapshop of the collected PPG signal using our testbed.

Algorithm 1 Peak Detection

INPUT: $x_u(t)$ % Heartbeat signals of sensor u
OUTPUT: $\{t(i), \Delta p_u(i)\}$ % Detected IPI of sensor u
% Wavelet Analysis
Apply wavelet decomposition to $x_u(t)$.
Remove the low frequency components.
Apply wavelet reconstruction and get $x_u^w(t)$.
% Peak Elimination
Partition $x_u^w(t)$ into segments with length of 300 ms.
Detect the peak in each segment and set the rest as 0.
Obtain the peak index $\{t'(k)\}$.
% Peak Refinement
Calculate intervals between adjacent detected peaks, $\Delta t_u(k) = t'(k+1) - t'(k)$
Cluster Δt_u into two groups (correct and incorrect).
Detect incorrect intervals and calibrate the corresponding peaks.
Update the detected $\{t(i), \Delta p_u(i)\}$.

3.1.3 Peak Refinement

There will be void peaks detected, as exemplified in Fig. 4(c). While the peak amplitudes may vary significantly, the range of IPIs is rather stable, which inspires us to further remove void peaks based on peak intervals.

As shown in Algorithm 1, we first calculate the intervals between any two adjacent detected peaks, including void peaks. The calculated intervals consist of the correct IPIs and the incorrect ones, which can be clustered into two groups. We can then find the timestamps that result in the incorrect intervals and calibrate the peaks of these timestamps by setting their values as zero. Each device finally refines the detected IPIs and obtain $\{t(i), \Delta p_u(i)\}$.

3.2 IPI Alignment

After the above steps, there could still exist misdecteted or missed peaks, making Alice and Bob hard to reach an agreement. However, key establishment requires common peaks at both ends for generating identical keys. An intuitive way is to directly compare the timestamps of the peaks at both devices and only keep the common ones. Unfortunately, it is not applicable because the timestamps (IPI) are the random source and should not be transmitted publicly.

Algorithm 2 IPI Indexing

INPUT: $\{t(i), \Delta p_u(i)\}$
INPUT: T_{ref} % reference IPI
OUTPUT: $\{t(i), S_u(i), \Delta p_u(i)\}$
$S_u(1) = \left\lfloor \frac{t(1)}{T_{ref}} \right\rceil + 1$;
for $i \leftarrow 2$ **to** N **do**
  $S_u(i) = \left\lfloor \frac{t(i)}{T_{ref}} \right\rceil + 1$;
  if $S_u(i) == S_u(i-1)$ **then**
    $S_u(i) = 0$
  end if
end for
Remove void peaks (whose sequence numbers are zero)

A simple yet efficient IPI alignment algorithm is designed by indexing each IPI a sequence number and aligning between Alice and Bob based on the common sequence numbers. This is inspired by the packet sequence number used in wireless protocols. Heartbeats are continuous, hence the peaks and the IPIs can be indexed based on their corresponding timestamps, as explained in Algorithm 2. Alice and Bob will use the same reference IPI, T_{ref}, e.g., the mean value of Alice's IPIs. In this case, Alice will first calculate T_{ref} and then transmit it to Bob wirelessly. Because both Alice and Bob measure heartbeat signals simultaneously, their timestamps are almost the same. For any i^{th} peak of the sensor u, its sequence number can be calculated as

$$S_u(i) = \left\lfloor \frac{t(i)}{T_{ref}} \right\rceil + 1, \tag{2}$$

where $\lfloor \cdot \rceil$ denotes the operating of rounding to the nearest integer. If it is the same as the sequence number of its previous IPI, it indicates that this IPI is invalid and its $S_u(i)$ is set as zero.

The alignment will require exchanging the sequence number between Alice and Bob. Alice will transmit sequence numbers of her IPIs, S_A, to Bob, who will locally compare with his sequence numbers, S_B. Bob will work out a common list by calculating the intersection between S_B and S_A, i.e., $S_{AB} = S_B \cap S_A$, and send the list to Alice. Both Alice and Bob will only keep the IPIs matching to the common list, $\{S_{AB}\}$. It is worth noting that the sequence numbers are publicly transmitted from Alice to Bob but it does not reveal the information of the IPIs.

After the above steps are completed, Alice and Bob will obtain a list of common IPIs, $\{t(i), \Delta p_A(i)\}$ and $\{t(i), \Delta p_B(i)\}$, respectively. It should be noted that while Alice and Bob have the IPIs with the common index, their IPI values, namely $\Delta p_A(i)$ and $\Delta p_B(i)$, will not be identical, due to the measurement errors and sampling noise.

4 Key Establishment Protocol

After Alice and Bob extract the aligned IPIs, they will need to convert the analog IPI values into digital binary sequences, which are required by the cryptographic applications. Key generation from wireless channels has been investigated comprehensively in the last decade [26], which

Algorithm 3 IPI trend-based quantization algorithm

INPUT: $\Delta p_u(i)$ % IPI interval of sensor u
OUTPUT: $K_u^q(i)$ % Key sequence of sensor u
for $i \leftarrow 1$ **to** $N-1$ **do**
  if $\Delta p_u(i+1) > \Delta p_u(i)$ **then**
    $K_u^q(i) = 1$
  else
    $K_u^q(i) = 0$
  end if
end for

consists of channel probing, quantization, information reconciliation, and privacy amplification. This paper exploits the randomness source from heartbeats, hence IPI extraction is designed to replace channel probing. The rest three steps are borrowed from wireless key generation.

4.1 Quantization

There have been several quantization algorithms to extract randomness from IPIs. Much work agreed that the four least significant bits (LSBs), i.e., bits 5678 of IPI have high entropy [10], [11]. However, Ortiz-Martin *et al.* found it may not be correct via an in-depth entropy test [27]. They found that the combination of bits 2638 is better than the four LSBs. Chizari *et al.* reckoned IPI values are not fully suitable as random sources based on extensive tests on huge datasets with almost 900,000,000 IPIs [9]. To the best knowledge of the authors, this is the most comprehensive evaluation. They further demonstrated that the IPI trend is a better candidate, which is therefore used in this paper.

The IPI trend-based quantization algorithm is given in Algorithm 3. The trend-based quantization has also been investigated in wireless key generation, which is named as differential-based quantizer in [28], [29]. This quantizer is very easy to implement as it only requires comparing the adjacent IPI values, which does not involve any complicated operations [29]. After this stage, Alice and Bob obtains binary sequence, K_A^q and K_B^q, respectively, which are usually not identical due to the difference between Δp_A and Δp_B.

4.2 Information Reconciliation and Privacy Amplification

We apply information reconciliation to correct the key mismatch between Alice and Bob. Secure sketch is used [30], explained in Algorithm 4. Without loss of generality, Alice acts as the initiator of this process. Alice and Bob will use the same error correction code (ECC) set, e.g., a BCH codeset $\mathcal{C}(n, k, t_c)$ which has a correction capacity of t_c/n.

Alice first randomly selects a codeword, c, from $\mathcal{C}$. She will then mask her key, K_A^q, with c using a simple exclusive-OR (XOR) operation and will send the syndrome, s, to Bob. Assuming Bob receives s correctly with the help of channel coding, he can reveal c_B by XORing s with his key, K_B^q. When the following condition satisfies,

$$\frac{d_H(c_B, c)}{l_K} < \frac{t_c}{n}, \tag{3}$$

where $d_H(\cdot,\cdot)$ denotes the calculation of hamming distance and l_K is the key length, Bob will obtain $c' = c$. He can finally reconcile his key by XORing c' and s. Alice and Bob will get the same keys, i.e., $K_A^{ir} = K_B^{ir}$.

Algorithm 4 Information reconciliation - secure sketch

INPUT: K_A^q, K_B^q % Quantized keys of Alice and Bob
INPUT: $\mathcal{C}$ % ECC set shared by Alice and Bob
OUTPUT: K_A^{ir}, K_B^{ir} % Reconciled key
Alice randomly selects a codeword c from an ECC set $\mathcal{C}$
Alice calculates $s = \text{XOR}(K_A^q, c)$
Alice transmits s to Bob through a public channel
$K_A^{ir} = K_A^q$
Bob receives s
Bob calculates $c_B = \text{XOR}(K_B^q, s)$
Bob decodes c_B to get c'
Bob calculates $K_B^{ir} = \text{XOR}(c', s)$

In practice, the key agreement can be confirmed by cyclic redundancy check (CRC). Alice can calculate the checksum of her key, $CRC(K_A^{ir})$, and transmit it to Bob. Bob can also calculate the checksum of his key, $CRC(K_B^{ir})$, and compare with the received checksum to deduce the key agreement. In the event that the checksums of Alice and Bob do not match, key generation fails and Alice and Bob have to restart from the IPI extraction.

There are information exchanges, namely the syndrome and CRC checksum, over the public wireless channel during information reconciliation, which can be overheard by eavesdroppers. Therefore, privacy amplification is adopted to remove the information leakage, which can be achieved by using a hash function $\mathcal{H}(\cdot)$, given as

$$K_u = \mathcal{H}(K_u^{ir}). \tag{4}$$

After the privacy amplification, both sensors shall obtain the same keys. The key generation process is completed.

4.3 Evaluation Metrics

4.3.1 Key Disagreement Rate (KDR)

KDR represents the percentage of the different key bits between sensor u and sensor v, given as

$$KDR_{uv} = \frac{\sum_i^{l_K} |K_u^q(i) - K_v^q(i)|}{l_K}. \tag{5}$$

KDR is only used offline to evaluate the key mismatch during each key generation round. The calculation requires keys at both devices, i.e., K_u^q and K_v^q, but we are not allowed to transmit the keys in a practical system.

4.3.2 Uniqueness

Uniqueness evaluates how the heartbeats present different patterns and the keys deviate from each other, when two sensors are attached to two persons. As will be explained in Section 6.1, this can be tested by attaching two PPG sensors to the fingertips of two volunteers.

We can still use the KDR to evaluate the key uniqueness when heartbeat signals are captured from two persons.

4.3.3 Randomness

As the generated key serves for cryptographic algorithms, e.g., AES, the key should be random. Otherwise, the system will be vulnerable to brute force attacks.

National Institute of Standards and Technology (NIST) has provided a randomness test suite to evaluate the true random number generator and pseudo random number generator [31]. Each test in the NIST test suite will return a p-value. When it is larger than a threshold, e.g., 0.01, the sequence is deemed to pass the particular test. Otherwise, the sequence is not random. This suite has also been widely used in the heartbeat-based key generation work [9]–[11], [27] as well as key generation from wireless channels [26].

4.3.4 Key Generation Rate (KGR)

KGR describes how fast the system can generate the keys, which is defined as

$$KGR = \frac{l_K}{T_K}, \tag{6}$$

where T_K is the total time for acquiring the heartbeats.

5 Performance Evaluation on ECG Signals

5.1 MIT PhysioBank Dataset

MIT PhysioBank is a free and public database which hosts extensive physiological and clinical data and related software [14]. Many heartbeat-based key generation work relies on this database [10], [12], [15]–[18]. We also used this extensive resource to evaluate our protocol on ECG signals. In particular, we used the following databases:

- MIT-BIH Arrhythmia Database (mitdb) [32]
- MIT-BIH Normal Sinus Rhythm Database (nsrdb) [14]
- MIT-BIH Long-Term ECG Database (ltdb) [14]
- European ST-T Database [33]
- Long Term ST Database (ltstdb) [34]

These databases cover both healthy subjects (nsrdb) and people with heart diseases (mitdb). In addition, as shown in Table 2, many datasets have quite long ECG signals, e.g., datasets ltdb, ltstdb and nsrdb provide more than 20 hours ECG signal records. This allows us to evaluate our protocol against different health conditions and make the analysis statistically meaningful.

We used the Matlab functions provided by the PhysioBank[3] to read the signals from the dataset. Our data analysis algorithms are implemented in Matlab.

5.2 Results

5.2.1 IPI Extraction and KDR

The IPIs of Alice and Bob are given in Fig. 6, using the mitdb/101 dataset as an example. Except some anomaly detected IPIs, these two curves in general match each other very well. Hence, they can serve the common random source to generate keys for Alice and Bob.

The performance of our proposed IPI extraction algorithm is given in Table 2. Many MIT PhysioBank databases provide annotation for the R peaks of the ECG signals. For example, MIT-BIH Arrhythmia Database is annotated by two or more cardiologists independently [32]. These annotations are considered correct and used as benchmarks to evaluate our algorithm. As shown in Table 2, our algorithm can detect a high proportion of correct peaks, compared with the annotated ones, ranging from 81.12% to 96.12%. In addition, we also compared the timestamps of our detected peaks of the sensor A with those of the annotation to evaluate the accuracy of our IPI detection algorithm. We deliberately selected the worst case and presented the results in Fig. 7; our algorithm can achieve a high accuracy.

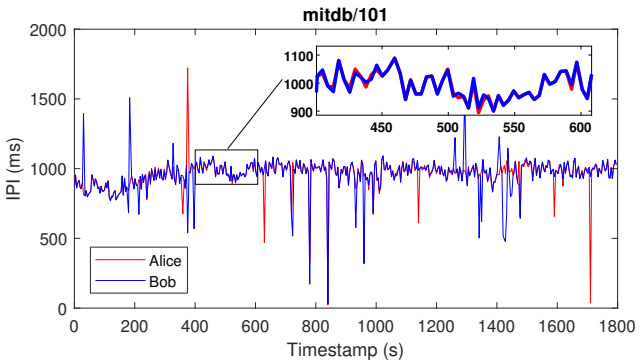


Fig. 6. The IPIs of Alice and Bob. Dataset: ECG signal from mitdb/101

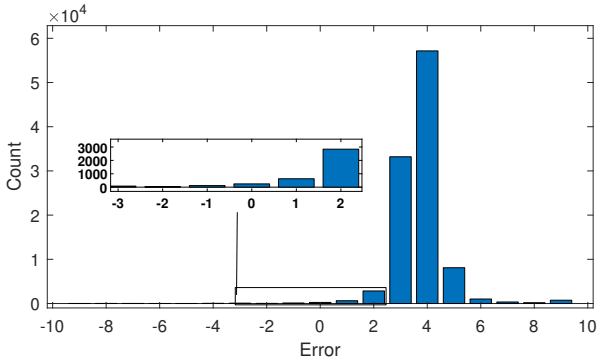


Fig. 7. The distribution of peak detection errors. The error is calculated as the timestamps differences (in millisecond) between our algorithm and the annotation. Dataset: ECG signal from ltdb/14046.

We have also calculated the distribution of the IPIs of the ECG signals, as exemplified in Figs. 8(a)-(e). We calculated the mean and variance of the measured IPIs and plotted curves following the normal distribution with their individual estimated mean value and variance. As can be observed, the measured IPIs do not follow a normal distribution.

The benchmark KDR is calculated as follows. We first directly compare the timestamps of all IPIs of Alice and Bob and only keep the ones close to each other. In practice, it is not possible as either device is not able to know the timestamp of other devices. We then quantize these refined IPIs and calculate their KDR as the benchmark. As shown in Table 2, the KDR of our scheme is very close to the benchmark KDR, which indicates the effectiveness of our IPI alignment algorithm. In addition, all the KDRs are smaller than 20%, which are well within the correction capacity of ECCs. For example, BCH can correct up to 25%

[3]https://archive.physionet.org/physiotools/matlab/wfdb-app-matlab/html/rdann.html

TABLE 2
IPI Extraction and Alignment Performance.

Dataset	Duration (minutes)	f_s (Hz)	N_{ann}, # of Annotated Peaks	N_A, # of Detected Peaks at A	N_B, # of Detected Peaks at B	N_{AB}, # of Common Peaks at A and B	N_{AB}/N_{ann}, Peak Detected Ratio	Benchmark KDR	KDR
edb/e0103	120	250	7336	7316	7320	7051	96.12%	5.80%	5.90%
edb/e0113	120	250	9173	8846	8945	8111	88.42%	7.90%	8.10%
ltdb/14046	1410	128	115278	106241	108144	93514	81.12%	20.40%	21.10%
ltdb/15814	1333	128	103388	103453	100517	94510	91.41%	20.60%	20.70%
ltstdb/s20011	1373	250	100053	100731	100809	92142	92.09%	10.70%	9.80%
ltstdb/s20231	1406	250	103100	99735	99461	91136	88.40%	13.70%	14.20%
mitdb/101	31	360	1874	1885	1889	1798	95.94%	11.10%	12.80%
mitdb/123	31	360	1519	1552	1601	1449	95.39%	11.80%	12.80%
nsrdb/16272	1500	128	97146	97281	99159	87744	90.32%	14.50%	16.00%
nsrdb/19093	1394	128	83670	88127	88627	78458	93.77%	11.50%	14.10%
PPG	5	225	NA	407	407	379	NA	5.50%	5.90%

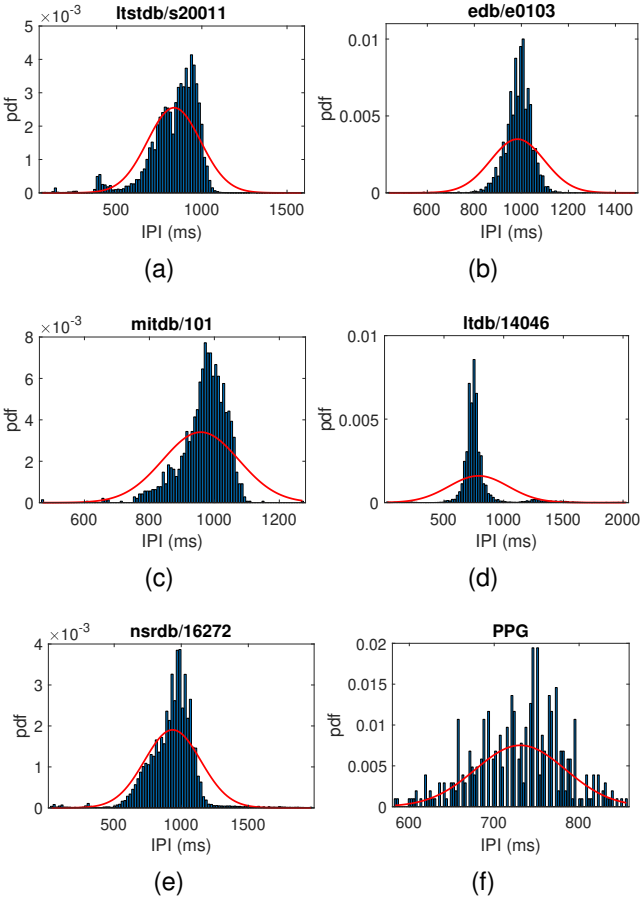


Fig. 8. IPI distribution. (a)-(e) describe ECG signals from different datasets. (f) describes PPG signals from the experiment 1. The red curves are the normal distribution with the mean and variance of the data.

mismatch [35]. Hence, the same keys can be generated at Alice and Bob.

5.2.2 KGR

The generated keys usually serve for cryptographic applications, which are not required to refresh the keys in real time. For example, AES has been commonly applied to protect many networks, including ZigBee, WiFi and LoRaWAN. The key lengths for AES can be 128-, 192-, or 256-bit. Our protocol can extract one bit from each heartbeat on average. When the rate is 60 heartbeats per minute, it takes up to five minutes to generate a set of 256-bit key, which still meets the requirements of these protocols.

There have also been research efforts to investigate one-time pad (OTP) encryption to achieve perfect secrecy, which encrypts each message bit with a random key bit [15]. As the keys cannot be reused, OTP encryption requires a fast KGR. Multiple features of ECG signals have been extracted to improve KGR, which can generate 16 bits per heartbeat cycle [17]. However, it is still too slow to transmit bulk data. For example, the data rate of ZigBee is 250 kps and the rate of WiFi is in the order of Mb/s. It will only be feasible to exchange confidential and short messages securely using the OTP, e.g., transmitting vital control signals to the implantable devices such as pacemakers.

5.2.3 Randomness

Chizari *et al.* has carried out a comprehensive evaluation of the randomness of the keys generated from ECG signals using the IPI trend-based quantization [9]. As this paper also used the same trend-based quantization method to generate keys from ECG signals, hence the randomness results are omitted for simplicity.

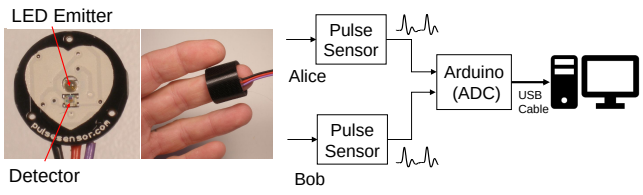
Interested readers please refer to [9]–[11], [27] for the randomness evaluation results when other quantization methods are used to generate keys from ECG signals.

6 Experimental Evaluation on PPG Signals

6.1 Experimental Platform

We created a testbed to collect PPG signals. As shown in Fig. 9(a) and Fig. 9(b), a PPG sensor-based experimental system consists of pulse sensors to collect heartbeat signals, Arduino for analog-to-digital conversion, and PC (Matlab) for signal processing.

We did two experiments:



(a) (b)

Fig. 9. (a) Pulse sensor worn to a finger. (b) The schematic diagram of the experimental setup.

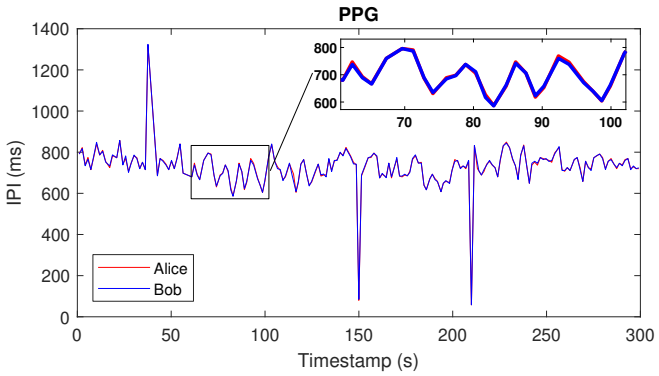


Fig. 10. The IPIs of Alice and Bob. Dataset: PPG signals collected in experiment 1.

- Experiment 1: Two PPG sensors, Alice and Bob, were attached to the fingertips of two hands of the volunteer. This experiment represented two legitimate sensors wishing to generate a common key.
- Experiment 2: Two PPG sensors, Alice and Eve, were attached to the fingertips of two volunteers. This experiment evaluated the uniqueness of the key.

Each experiment ran for five minutes. The sampling frequency for both experiments was configured as 225 Hz. Both sensors captured heartbeat signals simultaneously. The experiments and data collection have been approved by the Research Ethics Committees of the University of Liverpool, Liverpool, UK (reference: 5856). The same Matlab codes for processing ECG signals were used.

6.2 Results

6.2.1 IPI Extraction and KDR

Regarding the experiment 1, the IPIs of Alice and Bob measured from PPG signals are given in Fig. 10, which indicates a good agreement. The IPI extraction and alignment performance of the PPG signals are given in Table 2 (last row). The distribution of the IPIs measured by PPG signals is exemplified in Fig. 8(f), which exhibit a good variation.

The KDR between Alice and Bob is 5.9%, which is within the correction capacity of ECCs, hence the key disagreement can be corrected and key establishment can be completed.

6.2.2 Uniqueness

Regarding the experiment 2, the IPIs of Alice and Eve measured from the PPG signals are given in Fig. 11. Their IPIs varied in totally different manners. The KDR between

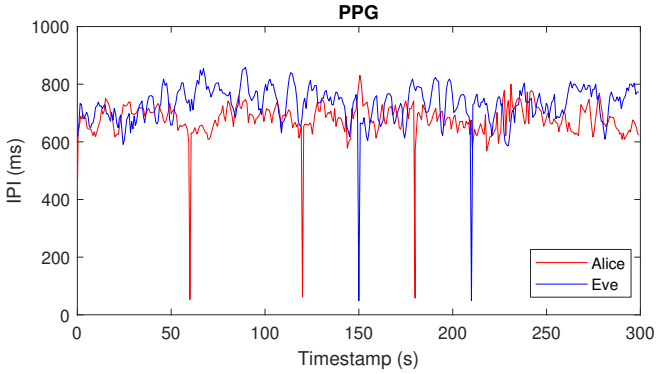


Fig. 11. The IPIs of Alice and Eve. Dataset: PPG signals collected in experiment 2.

TABLE 3
NIST randomness test results. Keys generated from PPG signals at the quantization and privacy amplification steps.

Test	Quantization	Privacy Amplification
Monobit frequency	0.187	0.532
Block frequency	0.245	0.437
Cum. Sums (fwd)	0.300	0.469
Cum. Sums (rev)	0.211	0.946
Run	0.797	0.515
Longest one block	0.073	0.973
Serial 1	0.318	0.825
Serial 2	0.647	0.938
Appro. Entropy	0.279	0.772
DFT	0.093	0.136

the keys at Alice and Eve is 51.6%, which is no better than a random guess. This demonstrated the uniqueness of the keys because Eve generates uncorrelated keys and she cannot deduce the keys of Alice.

6.2.3 Randomness

We evaluated the randomness of the keys of Alice generated from PPG signals using a Python implementation of the NIST test suite[4]. We tested the keys both at the quantization and privacy amplification steps. When a key at the quantization step is non-random, it might still be able to pass the randomness test after the privacy amplification [36]. In this case, the key is not secure as it will be vulnerable to dictionary attacks [36]. Hence, we need to ensure the keys at both stages are random.

The results are given in Table 3. All the p-values are larger than 0.01, hence the generated key sequences pass the NIST randomness tests. Therefore, PPG signals are suitable to be extracted as cryptographic keys.

6.3 Prototype and Demonstration

We also created a prototype system, whose schematic diagram and photo are given in Fig. 12(a) and Fig. 12(b), respectively. Raspberry Pi is used for signal processing and

[4]https://github.com/stevenang/randomness_testsuite

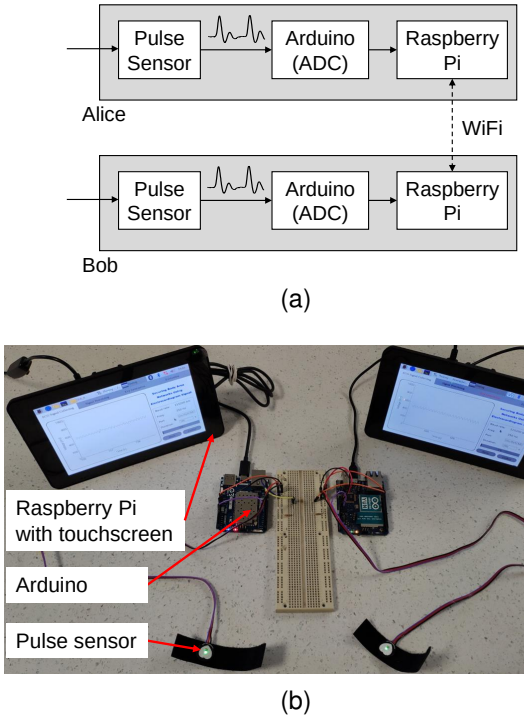


Fig. 12. (a) The schematic diagram of the prototype system. (b) The photo of the prototype system.

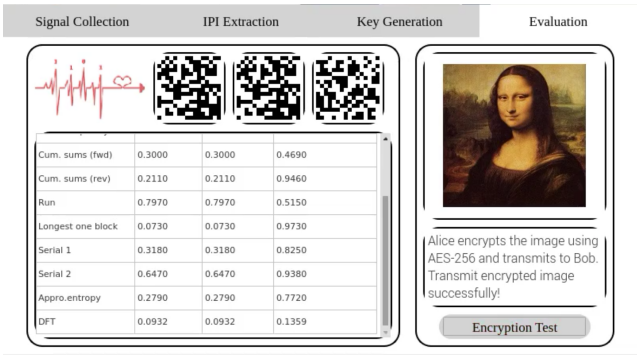


Fig. 13. The graphic user interface (GUI) of the heartbeat-based key generation prototyping.

display (with the help of a touchscreen). All the algorithms are implemented using Python. A guideline of the hardware connections and codes are provided by the manufacturer[5].

Different from the experimental testbed, there is no physical connection between the two units. Alice and Bob are running the key generation protocol individually. They can exchange the required information by WiFi, e.g., the sequence number S_A, the common sequence list S_{AB}, the syndrome s, and the CRC checksum of Alice $CRC(K_A^{ir})$.

A demonstration video has been created and submitted, which exemplified the practicability of our system. As shown in Fig. 13, we partitioned our system into four parts, namely signal collection, IPI extraction, key generation and evaluation. Besides the algorithms introduced in the previous sections, we implemented an encryption test in the evaluation part to demonstrate a successful cryptographic integration. Alice uses her key to encrypt an image using AES and transmits it to Bob. Once receiving it successfully, Bob will be able to decrypt the image using his key. Please refer to our demonstration video for full information.

7 Discussion

There will be multiple wearable devices in the WBAN exchanging private and/or confidential data, which requires encryption and decryption for transmissions, e.g., using AES. Sharing common keys for them is indeed challenging as they are usually embedded devices with limited computational resources. Key generation from heartbeat is a promising solution. When the devices are attached to the same person, they will be able to extract the same IPIs, and hence generate same keys, which can be observed from Fig. 6 (for ECG signals) and Fig. 10 (for PPG signals). The generated common key can be used for symmetric encryption and decryption.

The generated key is dynamic and random. This is because IPI varies and any two IPIs will never be identical, as shown in Fig. 6 (for ECG signals) and Fig. 10 (for PPG signals). Therefore, the key can be refreshed regularly, approximately every five minutes for a set of 256-bit key, as discussed in Section 5.2.2. Even if a key is stolen, a new key can be generated quickly to replace the revealed key.

[5]https://github.com/WorldFamousElectronics/Raspberry_Pi/

8 Related Work

8.1 Heartbeat-based Authentication

Besides key generation, heartbeat signals have also been used for user authentication, based on the uniqueness of the human cardiac systems. Arteaga-Falconi *et al.* used two ECG electrodes to collect heartbeat signals for authentication [37]. Zhao *et al.* designed a sophisticated system by using wrist-worn PPG sensors [38]. The system may be subject to human movement and the authors developed a motion artifacts detection and mitigation strategy. Different from the previous work, Wang *et al.* used a built-in accelerometer on the smartphones to capture heartbeat signals [39]. The user simply presses the phone to the chest and can be identified by using a few heartbeats. In order to improve the authentication accuracy, Wu *et al.* used both the motion sensor and PPG sensors [40]. All these research efforts have demonstrated the feasibility of heartbeat-based authentication.

Different from authentication, heartbeat-based key generation will require further consideration to agree on common keys based on inaccurate measurements of the heartbeat signals between two or more sensors, which is the research focus of this paper.

8.2 Key Generation from Wireless Channels

Besides heartbeat-based key generation, key generation from wireless channels is also widely investigated for WBAN, which exploits the unpredictable features of the wireless channel as the key [26]. This usually occurs between two wireless devices mounted on two persons [41], [42], and the protocol extracts the common randomness from the wireless channel between these two devices.

TABLE 4
COTS PPG and ECG Chips, Development Kits and Arduino Boards.

Type	Chip	Evaluation Kit	Arduino Board
ECG	TI ADS1292[6]	TI ADS1292ECG kit[7]	ProtoCentral kit[8]
	AD AD8232[9]	AD8232 Evaluation Board[10]	SparkFun SEN-12650[11], DFROBOT SEN0213 [12]
	MAX30003[13]	MAX30003WING[14]	ProtoCentral kit[15]
PPG	MAX30101[16], MAX86140[17]	Maxim board 7141[18]	SparkFun SEN-15219[19]
	NA	NA	Pulse sensors[20], DFROBOT SEN0203[21]

This technique requires wireless transmissions between two devices to glean channel information, which will introduce additional energy consumption and is not friendly to many wearable and implantable devices. Ali *et al.* proposed to leverage the existing data communications to measure the channel [41]. However, it is restricted to the users' mobility pattern and frequency of data exchanges. As analyzed in [18], it is affected by the surrounding environments, e.g., there may be severe interference from other medical equipment in the hospital.

To this end, it is desirable to extract keys from IPIs of heartbeats when devices have the capability of measuring the cardiac signals. On the other hand, there has been extensive research on key generation from wireless channels [26]. While the different randomness sources lead to varied algorithms to harvest entropy, other steps, namely quantization, information reconciliation, and privacy amplification, can be borrowed for heartbeat-based key generation, as discussed in Section 4.

8.3 ECG and PPG Sensors

A detailed description and comparison of ECG and PPG sensors can be found in [24], [43]. Table 4 presents a non-exclusive list of the COTS ECG and PPG chips and sensors. However, many devices usually do not provide application programming interfaces (APIs) to access the raw data. On the other hand, there are a number of open source boards that provide Arduino libraries. These development sensor boards and their software libraries will allow researchers to develop prototypes and collect real data. There have been research attempts using pulse sensor [24], [40], TI ADS1298 [11] and AD ADAS1000 [20] to establish the heartbeat-based key generation testbeds.

[6]http://www.ti.com/product/ADS1292
[7]http://www.ti.com/tool/ADS1292ECG-FE
[8]https://www.protocentral.com/analog-adc-boards/783-ads1292r-ecgrespiration-breakout-board.html
[9]https://www.analog.com/en/products/ad8232.html
[10]https://www.analog.com/media/en/technical-documentation/user-guides/AD8232-EVALZ_UG-514.pdf
[11]https://www.sparkfun.com/products/12650
[12]https://www.dfrobot.com/product-1510.html
[13]https://www.maximintegrated.com/en/products/analog/data-converters/analog-front-end-ics/MAX30003.html
[14]https://datasheets.maximintegrated.com/en/ds/MAX30003WING.pdf
[15]https://www.protocentral.com/open-source-health/1149-protocentral-max30003-single-lead-ecg-breakout-board.html
[16]https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30101.html
[17]https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX86140.html
[18]https://www.maximintegrated.com/en/design/reference-design-center/system-board/7141.html
[19]https://www.sparkfun.com/products/15219
[20]https://pulsesensor.com/
[21]https://www.dfrobot.com/product-1540.html

9 Conclusion

In this paper, we designed a robust, generic, and practical heartbeat-based key generation framework that is applicable for both ECG and PPG signals. It consists of a peak detection algorithm to extract pulse peaks and an IPI alignment algorithm to enable the same indexed IPIs between legitimate devices. Key establishment protocol involves quantization, information reconciliation and privacy amplification, which enable legitimate devices to convert analog IPIs to digital key bits and reconcile on a common sequence. We carried out extensive evaluation using several ECG datasets from the public MIT PhysioBank databases. We also designed a testbed to collect PPG signals and evaluated our protocol. Results indicated our framework can accurately extract IPIs and key generation can be completed using both ECG and PPG signals. A PPG-based prototype was also created and a demonstration video was submitted to show a successful working system. Our future work will be studying the effect of sensor attachment on the IPI measurements when e.g., the human is walking.

References

[1] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1658–1686, 2014.

[2] G. Zheng, R. Shankaran, M. A. Orgun, L. Qiao, and K. Saleem, "Ideas and challenges for securing wireless implantable medical devices: A review," *IEEE Sensors J.*, vol. 17, no. 3, pp. 562–576, 2016.

[3] M. Zhang, A. Raghunathan, and N. K. Jha, "Trustworthiness of medical devices and body area networks," *Proc. IEEE*, vol. 102, no. 8, pp. 1174–1188, 2014.

[4] E. Marin, D. Singelée, F. D. Garcia, T. Chothia, R. Willems, and B. Preneel, "On the (in) security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proc. Annual Conf. Computer Security Applications*, 2016, pp. 226–236.

[5] K.-A. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 577–601, 2015.

[6] S. Cherukuri, K. K. Venkatasubramanian, and S. K. Gupta, "Biosec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Proc. Int. Conf. Parallel Processing Workshops*, 2003, pp. 432–439.

[7] C. C. 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 Commun. Mag.*, vol. 44, no. 4, pp. 73–81, 2006.

[8] C. Camara, P. Peris-Lopez, H. Martín, M. Aldalaien *et al.*, "ECG-RNG: A random number generator based on ECG signals and suitable for securing wireless sensor networks," *Sensors*, vol. 18, no. 9, p. 2747, 2018.

[9] H. Chizari and E. C. Lupu, "Extracting randomness from the trend of IPI for cryptographic operators in implantable medical devices," *IEEE Trans. Depend. Sec. Comput.*, 2019.

[10] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li, "IMDGuard: Securing implantable medical devices with the external wearable guardian," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1862–1870.

[11] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-heart (H2H) authentication for implanted medical devices," in *Proc. ACM SIGSAC Conf. Computer & Communications Security*, Berlin Germany, Nov. 2013, pp. 1099–1112.

[12] R. M. Seepers, J. H. Weber, Z. Erkin, I. Sourdis, and C. Strydis, "Secure key-exchange protocol for implants using heartbeats," in *Proc. ACM Int. Conf. Computing Frontiers*, 2016, pp. 119–126.

[13] Q. Lin, W. Xu, J. Liu, A. Khamis, W. Hu, M. Hassan, and A. Seneviratne, "H2B: Heartbeat-based secret key generation using piezo vibration sensors," in *Proc. 18th Int. Conf. Information Processing in Sensor Networks (IPSN)*, Montreal Quebec, Canada, Apr. 2019, pp. 265–276.

[14] A. L. Goldberger, L. A. 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*, vol. 101, no. 23, pp. e215–e220, 2000, accessed on 18 July, 2020. [Online]. Available: https://physionet.org/about/database/

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

[16] N. Karimian, Z. Guo, M. Tehranipoor, and D. Forte, "Highly reliable key generation from electrocardiogram (ECG)," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 6, pp. 1400–1411, 2017.

[17] G. Zheng, G. Fang, R. Shankaran, M. A. Orgun, J. Zhou, L. Qiao, and K. Saleem, "Multiple ECG fiducial points-based random binary sequence generation for securing wireless body area networks," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 655–663, 2017.

[18] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Shen, "Flexible and efficient authenticated key agreement scheme for BANs based on physiological features," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 845–856, 2018.

[19] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Y. Huang, "Body area network security: robust key establishment using human body channel," in *Proc. 3rd USENIX Conf. Health Security and Privacy*, Aug. 2012, pp. 1–10.

[20] S. Pirbhulal, H. Zhang, W. Wu, S. C. Mukhopadhyay, and Y.-T. Zhang, "Heartbeats based biometric random binary sequences generation to secure wireless body sensor networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 12, pp. 2751–2759, 2018.

[21] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Pska: Usable and secure key agreement scheme for body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 1, pp. 60–68, 2010.

[22] W. Choi, Y. Lee, D. Lee, H. Kim, J. H. Park, I. S. Kim, and D. H. Lee, "Energy-aware key exchange for securing implantable medical devices," *Security and Communication Networks*, vol. 2018, 2018.

[23] 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," *J. Supercomputing*, vol. 75, no. 6, pp. 3288–3301, 2019.

[24] J. Blasco and P. Peris-Lopez, "On the feasibility of low-cost wearable sensors for multi-modal biometric verification," *Sensors*, vol. 18, no. 9, p. 2782, 2018.

[25] L. Mucchi and A. Carpini, "Aggregate interference in ISM band: WBANs need cognitivity?" in *Proc. Int. Conf. Cognitive Radio Oriented Wireless Networks Communications*, 2014, pp. 247–253.

[26] J. Zhang, T. Q. Duong, A. Marshall, and R. Woods, "Key generation from wireless channels: A review," *IEEE Access*, vol. 4, pp. 614–626, Mar. 2016.

[27] L. Ortiz-Martin, P. Picazo-Sanchez, and P. Peris-Lopez, "Are the interpulse intervals of an ECG signal a good source of entropy? an in-depth entropy analysis based on NIST 800-90b recommendation," *Future Generation Computer Systems*, vol. 105, pp. 346–360, 2020.

[28] B. Zan, M. Gruteser, and F. Hu, "Key agreement algorithms for vehicular communication networks based on reciprocity and diversity theorems," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 4020–4027, 2013.

[29] J. Zhang, A. Marshall, and L. Hanzo, "Channel-envelope differencing eliminates secret key correlation: LoRa-based key generation in low power wide area networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 462–12 466, 2018.

[30] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[31] "A statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, accessed on 18 July, 2020. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final

[32] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[33] 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*, vol. 13, no. 9, pp. 1164–1172, 1992.

[34] F. Jager, A. Taddei, G. B. Moody, M. Emdin, G. Antolič, R. Dorn, A. Smrdel, C. Marchesi, and R. G. Mark, "Long-term ST database: a reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia," *Medical and Biological Engineering and Computing*, vol. 41, no. 2, pp. 172–182, 2003.

[35] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Efficient key generation by exploiting randomness from channel responses of individual OFDM subcarriers," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2578–2588, 2016.

[36] G. Li, A. Hu, J. Zhang, L. Peng, C. Sun, and D. Cao, "High-agreement uncorrelated secret key generation based on principal component analysis preprocessing," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 3022–3034, 2018.

[37] J. S. Arteaga-Falconi, H. Al Osman, and A. El Saddik, "ECG authentication for mobile devices," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 591–600, 2016.

[38] T. Zhao, Y. Wang, J. Liu, and Y. Chen, "Toward continuous user authentication using PPG in commodity wrist-worn wearables," in *Proc. Annual Int. Conf. Mobile Computing and Networking (MobiCom)*, Los Cabos, Mexico, Oct. 2019, pp. 1–3.

[39] L. Wang, K. Huang, K. Sun, W. Wang, C. Tian, L. Xie, and Q. Gu, "Unlock with your heart: Heartbeat-based authentication on commercial mobile phones," *Proc. ACM Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–22, 2018.

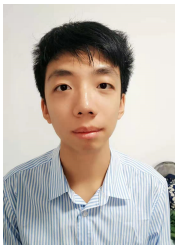
[40] G. Wu, J. Wang, Y. Zhang, and S. Jiang, "A continuous identity authentication scheme based on physiological and behavioral characteristics," *Sensors*, vol. 18, no. 1, p. 179, 2018.

[41] S. T. Ali, V. Sivaraman, and D. Ostry, "Eliminating reconciliation cost in secret key generation for body-worn health monitoring devices," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2763–2776, 2013.

[42] L. Shi, J. Yuan, S. Yu, and M. Li, "MASK-BAN: Movement-aided authenticated secret key extraction utilizing channel characteristics in body area networks," *IEEE Internet Things J.*, vol. 2, no. 1, pp. 52–62, 2015.

[43] N. Karimian, M. Tehranipoor, D. Woodard, and D. Forte, "Unlock your heart: Next generation biometric in resource-constrained healthcare systems and IoT," *IEEE Access*, vol. 7, pp. 49 135–49 149, 2019.

Junqing Zhang received the B.Eng and M.Eng degrees in Electrical Engineering from Tianjin University, China in 2009 and 2012, respectively, and the Ph.D degree in Electronics and Electrical Engineering from Queen's University Belfast, UK in 2016. From Feb. 2016 to Jan. 2018, he was a Postdoctoral Research Fellow with Queen's University Belfast. He is a Lecturer (Assistant Professor) with University of Liverpool, UK. His research interests include Internet of Things, wireless security, physical layer security, key generation, and radio frequency fingerprint identification.



Yushi Zheng received the B. Eng degree in Electrical and Electronic Engineering from University of Liverpool (UoL), UK in 2020. From Sep. 2020, he is working towards the MSc degree with University College London (UCL), UK.

Weitao Xu is an Assistant Professor at the Department of Computer Science at City University of Hong Kong. Before that, he was a Postdoctoral Research Associate at the School of Computer Science and Engineering at UNSW from June 2017 to August 2019. He obtained his PhD degree from the University of Queensland in 2017. He received his B.E. degree in Communication Engineering and M.E. degree in Communication and Information System both from the School of Information Science and Engineering, Shandong University, China, in 2010 and 2013, respectively. His research areas include mobile computing, sensor network and IoT.

Yingying (Jennifer) Chen is a Professor of Electrical and Computer Engineering and Peter Cherasia Endowed Faculty Scholar at Rutgers University. She is the Associate Director of Wireless Information Network Laboratory (WINLAB). She also leads the Data Analysis and Information Security (DAISY) Lab. She is an IEEE Fellow. Her research interests include mobile sensing and computing, cyber security and privacy, Internet of Things, and smart healthcare. Her background is a combination of Computer Science, Computer Engineering and Physics. She had extensive industry experiences at Nokia previously. She has published over 200 journal articles and conference papers. She is the recipient of multiple Best Paper Awards from EAI HealthyIoT 2019, IEEE CNS 2018, IEEE SECON 2017, ACM AsiaCCS 2016, IEEE CNS 2014 and ACM MobiCom 2011. She is also the recipient of NSF CAREER Award and Google Faculty Research Award. She received NJ Inventors Hall of Fame Innovator Award and is also the recipient of IEEE Region 1 Technological Innovation in Academic Award. Her research has been reported in numerous media outlets including MIT Technology Review, CNN, Fox News Channel, Wall Street Journal, National Public Radio and IEEE Spectrum. She has been serving/served on the editorial boards of IEEE Transactions on Mobile Computing (IEEE TMC), IEEE Transactions on Wireless Communications (IEEE TWireless), IEEE/ACM Transactions on Networking (IEEE/ACM ToN) and ACM Transactions on Privacy and Security.