Breathe-to-Pair (B2P): Respiration-Based Pairing Protocol for Wearable Devices

Jafar Pourbemany, Ye Zhu
pourbemany@ieee.org,y.zhu61@csuohio.edu
Department of Electrical Engineering & Computer Science
Cleveland State University
Cleveland, Ohio, USA

Riccardo Bettati
bettati@tamu.edu
Department of Computer Science and Engineering
Texas A&M University
College Station, Texas, USA

ABSTRACT

We propose Breathe-to-Pair (B2P), a protocol for pairing and shared-key generation for wearable devices that leverages the wearer's respiration activity to ensure that the devices are part of the same body-area network. We assume that the devices exploit different types of sensors to extract and process the respiration signal. We illustrate B2P for the case of two devices that use respiratory inductance plethysmography (RIP) and accelerometer sensors, respectively. Allowing for different types of sensors in pairing allows us to include wearable devices that use a variety of different sensors. In practice, this form of sensor variety creates a number of challenges that limit the ability of the shared-key establishment algorithm to generate matching keys. The two main obstacles are the lack of synchronization across the devices and the need for correct noise-induced mismatches between the generated key bit-strings.

B2P addresses the synchronization challenge by utilizing Change Point Detection (CPD) to detect abrupt changes in the respiration signal and consider their occurrences as synchronizing points. Any potential mismatches are handled by optimal quantization and encoding of the respiration signal in order to maximize the mismatch correction rate and minimize the message overheads. Extensive evaluation on a dataset collected from 30 volunteers demonstrates that our protocol can generate a secure 256-bit key every 2.85 seconds (around one breathing cycle). Particular attention is given to secure B2P against device impersonation attacks.

CCS CONCEPTS

• **Security and privacy → Security protocols**; **Mobile and wireless security**; **Security protocols**; *Key management*.

KEYWORDS

wearable devices, pairing, key generation, respiration, security, WBAN

ACM Reference Format:
Jafar Pourbemany, Ye Zhu and Riccardo Bettati. 2022. Breathe-to-Pair (B2P): Respiration-Based Pairing Protocol for Wearable Devices. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3507657.3528545

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WiSec '22, May 16–19, 2022, San Antonio, TX, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528545

1 INTRODUCTION

Wearable devices have become remarkably popular in many fields, such as health care, health management, the workplace, education, and scientific research. Nowadays, there are hundreds of different products in the market, including smartwatches, smart wristbands, smart glasses, smart jewelry, smart straps, smart clothes, smart belts, smart shoes, smart gloves, skin patches, and even implanted medical devices (IMDs). These devices are utilized in a variety of applications [13], such as monitoring of vital signs like heart rate, respiratory rate, body temperature, blood pressure, blood oxygen, and blood glucose.

Wearable devices can sense, collect, analyze, and store a wide range of information about human activities and behaviors to support a variety of different applications. Device-to-device (D2D) communication, expected to become prevalent in wearable applications [9], enables spectral-efficient data exchange between wearable devices due to spatial proximity. This data is often revealing and therefore sensitive. Therefore, communication between wearables should be protected by encryption, which gives rise to the need for a shared secure key.

Traditional pairing techniques either need user intervention or implicitly rely on device proximity. A common approach that is based on user intervention is Bluetooth pairing, which requires selecting a target device from a list of available devices, possibly further secured by a PIN. PIN-based methods need interaction with some sort of input device or display, which may be either inconvenient or impossible in many wearables [31]. Furthermore, PIN-based pairing is vulnerable to observation attacks such as shoulder surfing [31]. Public key cryptography (PKC), on the other hand, cannot be used to create a secure key on wearable devices, first because it requires a public key infrastructure (PKI), and second because asymmetric encryption is too expensive for resource-limited wearable devices. Various techniques have been proposed to mitigate these limitations; these take advantage of common features in the different wearable devices participating in the pairing.

Proximity-based pairing, such as NFC or RFID, requires a tightly controlled transmission range in order to avoid relay attacks. Pairing based on observation of a biometric activity is a natural choice because wearable devices are attached to the same body. Both behavioral biometrics (e.g., gait) and physiological biometrics (e.g., heart rate and muscle activity) can be used for wearable device pairing and authentication. In recent years, various biometric-based pairing techniques have been provided that exploit a variety of sensors

such as accelerometers, gyroscopes, magnetometers, ECG sensors, PPG sensors, EMG sensors, and piezo sensors, among others.

Contributions: To the best of our knowledge, this paper is the first study to address the following problems in the pairing of wearable devices using biometric signals: We describe the **first use** of breathing signals to generate shared keys and so securely pair wearable devices. This study is also the **first attempt** to pair devices with **different** sensing mechanisms. The proposed method must address *(1)* different sampling rates on the different devices, *(2)* different sensing mechanism on the devices, *(3)* different dynamics of the signals collected from the devices, and *(4)* lack of synchronization between signals collected by the different devices.

We designed and implemented a key generation protocol, called Breathe-to-Pair (B2P), which uses the observation of the respiration activity to generate the same keys for two wearable devices that happen to use two different types of sensors to capture the breathing. As an example, we study the case two devices, of which the first uses Respiratory Inductance Plethysmography (RIP) and the second an accelerometer to measure respiration. Note that while we focus on the breathing activity to generate the signals used for device pairing, other activities can be used as well, such as heartbeat, gait and motion in general, or others. All of these activities can be captured by devices that measure different signals, which leads to many of the problems that we listed above.

The rest of the paper is organized as follows. Section 2 provides a background of the respiration mechanism and the techniques to observe respiration signals. Section 3 discusses the trust model and the adversary model. In Section 4, we motivate and describe the different stages of the B2P protocol. In Section 5, we explain the experiment setup and performance metrics. Then, we present the best results and investigate the impact of different factors on the B2P performance and the resistance to attacks. The discussion and related work are provided in Section 7 and Section 8. Finally, Section 9 concludes the paper.

2 BACKGROUND

2.1 Respiration Mechanism

A respiration cycle consists of inhalation and exhalation in which the lungs will be inflated or deflated via three muscles: diaphragm, sternocleidomastoid (SCM), rib muscles (intercostal). During inhalation, the external intercostal contracts to bring out the rib cage, the SCM contracts to pull the rib cage upwards, and the diaphragm contracts and goes down to expand the thoracic cavity and increase the air volume in the lungs. During exhalation, the diaphragm relaxes and goes up, and the internal intercostal and abdominal muscles contract to make the lungs deflate back. The movement of the rib cage is shown in Figure 1a. The normal breathing rate for a healthy adult human is 12-18 breathing cycles per minute at rest. This rate varies based on age.

2.2 Respiration Sensing Mechanisms

Continuous measurement of the respiratory rate (RR) is used in many healthcare and sports applications. The respiratory rate is often measured using airflow detection, but it can also be observed using chest and abdominal movement, and other signals. Several respiratory measurement techniques have been developed based on

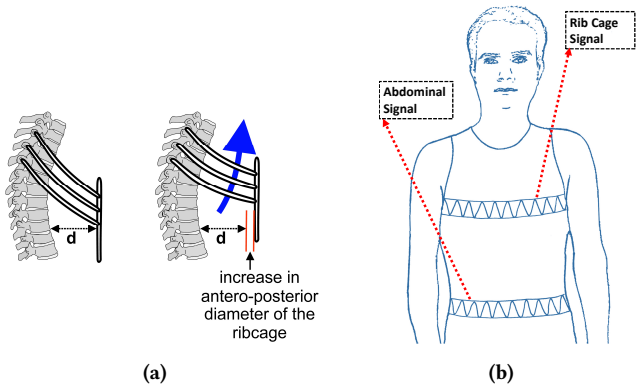


Figure 1: (a) Movement of the Rib Cage in Breathing [4], (b) RIP Bands at Chest and Abdomen

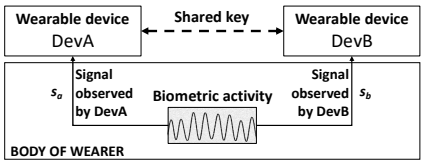


Figure 2: Model of the Biometric-based Key Generation

detecting chest and abdominal movement. Respiratory inductance plethysmography (RIP) is a common method to measure pulmonary ventilation. RIP consists of two belts of sinusoid wire coils positioned around the thorax and abdomen, as shown in Figure 1b. The changes in the anteroposterior diameter of the abdomen and the rib cage during respiration change the coils' self-inductance and oscillation frequency. The latter is converted to a digital signal, the amplitude of which is proportional to the respiratory volume [6]. A number of recent studies [11, 22] use accelerometers to detect the respiration signal based on the movement of the chest and abdominal wall. We note that a main challenge of this method is the removal of motion artifacts.

Many other techniques exist to measure respiration parameters. These techniques use a wide variety of sensors, such as anemometers, thermistors, thermocouples, pyroelectric detectors, fiber optic sensors, capacitive sensors, resistive sensors, microphones, nanocrystal and nanoparticles sensors, and many others [32].

Given the wide variety of sensor mechanisms to observe respiration, it is unlikely in practice that two wearable devices use the same signal to observe respiration. This leads to the difficulties described above when using respiration for pairing. In this paper, we develop a pairing protocol that allows the pairing of devices that collect different signals to observe the same activity, and we demonstrate how it enables effective pairing of devices that use different sensor technologies (RIP and accelerometers in the specific case described in this paper) to measure respiration. We expect that the protocol designed for the example of RIP and accelerometer can be directly applied to other combinations of sensing techniques as well, such as, *e.g.*, photoelectric reflection-type pulse wave sensors *vs.* phonocardiography for measuring heart rate.

3 SYSTEM MODELS

3.1 System Model

The goal of pairing two wearable devices *DevA* and *DevB* is two-fold: *(i)* Ensure that both *DevA* and *DevB* are "worn" by the same individual. *(ii)* Establish a shared key using information that is shared only by the two devices. An approach that addresses both requirements is to use a biometric activity, call it S, that can be accessed by the two devices but not by devices that are not connected to the body of the wearer (see Figure 2). In this paper, we choose *respiration* as the observed biometric activity. The challenge with using biometric activity is that the sensors at a device do not observe S directly, but rather infer S from physical measurements of features that are accessible to the device (in the example of respiration, these features are ventilation for RIP and chest movement for accelerometer sensors). For device *DevX*, the observed signal, call it s_x, for activity S can be affected as follows:

1) The signal s_x can be noisy. Noise can be due to measurement errors, or it can be caused by external disturbances. For example, an accelerometer may capture motions that are not caused by breathing. 2) The signal s_x may be delayed. This may be due to time discretization or other data collection delays. 3) The signal s_x may have an inconsistent specification, such as a different sampling rate or different amplitude range.

For all these reasons, the two signals s_a and s_b, observed by *DevA* and *DevB*, are not equal. They may even be difficult to correlate at a granularity fine enough to allow for the reliable and low-latency establishment of shared keys.

We make the following assumptions for the pairing: *(i)* The wearable devices are attached to the same body, and the pairing is based on the breathing activity of the subject. *(ii)* The two devices are free to choose their technique to collect respiration signals. In our case, one device uses a RIP sensor and the other an accelerometer. *(iii)* The two devices are free to sample at different rates, and may not have support for synchronization. *(iv)* The two devices do not share any prior information before the pairing.

3.2 Threat Model

We assume a powerful adversary who knows the details of the B2P protocol. Due to the broadcast nature of the wireless communication between wearable devices, the adversary is capable of eavesdropping on the communication between legitimate wearable devices. Through eavesdropping, the adversary can access the content of the exchanged packets between the two devices before their successful pairing, as these packets are not yet encrypted.

Although the adversary cannot attach a wearable device to the body to pair with a legitimate device, the adversary may attempt to pair with a legitimate device with the following impersonation attacks: (1) The adversary may attempt to pair with a legitimate device based on a third party's respiration signals. (2) The adversary may be able to remotely observe the victim's breathing activities, typically using a camera, and then process the collected data either manually, or using automated tools [21]. She can then approximate the breathing activity based on the data. With the knowledge of the pairing protocol, the access to the content of packets exchanged between legitimate wearable devices, and access to a third party's signals or the biometric activity (respiration in our case) through a camera, the adversary is in a position to launch either type of impersonation attacks.

4 PROTOCOL DESIGN

For a pairing protocol that is based on a biometric process to support a variety of devices, it must address the following challenges, which we have described above: (1) the features collected by the different devices give rise to signals with different sample frequencies, amplitudes, and noise characteristics (measurement noise vs. disturbances). (2) The signals collected by the different devices are likely not sufficiently synchronized to be used as shared data needed for the key exchange. (3) In order to minimize its own quantization error, each device optimizes its quantization levels. When devices have different quantization levels, they interpret the signal differently, which leads to errors in the pairing process. (4) Finally, even if all challenges above are addressed, errors may remain across the collected signals, which again leads to errors in the pairing.

In the following, we describe how we address each of these challenges. At the end of this section, we describe how this naturally gives rise to the design, architecture, and finally to an implementation, of the B2P pairing protocol.

4.1 Differing Signals Characteristics

We handle the problem of signals being collected at different frequencies, having different amplitudes, and different noise characteristics as follows:

Different Sample Frequencies: One approach to handle the case two signals that are collected at different rates is to *down-sample* the higher-rate signal. For downsampling, we can use skipping (keep sample closest to a sampling point and skip the rest), or averaging (interpolate new sample from the closest samples).

Different or Unequal Signal Amplitudes: We can easily handle this by *normalization*, such that the signal lies in the range of $[-1, 1]$.

Different Noise Characteristics: All biometric sensors suffer from measurement errors, which can be largely eliminated by filtering high frequency components. In addition, the collected data contains noise due to external disturbances. In our example, accelerometers pick up motion that is unrelated to breathing. Similarly, RIP sensors may get disturbed by a poor fit of the vest.

Since the human respiration rate is within a range from 6 to 30 per minute [26], a low-pass filter with the cut-off frequency of 0.5 Hz or a band-pass filter with cut-off frequencies of 0.1-0.5 Hz can be used to filter out both measurement noise and disturbances. In order to decrease the difference between the two devices even further, we apply a Savitzky-Golay filter to eliminate the effect of possible noise in the range of the band-pass filter's cut-off frequencies and smoothen the respiration signal without changing its features.

4.2 Lack of Synchronization

Keeping the signals synchronized is important to pairing performance. The synchronization is especially critical for pairing with signals collected with different sensing mechanisms. In such cases, the mode transitions in the biometric activity (inhalation vs. exhalation in the case of breathing) may be detected at different points in time by the different sensing mechanisms. A naïve method for synchronization is to use *obvious features* such as peaks and valleys

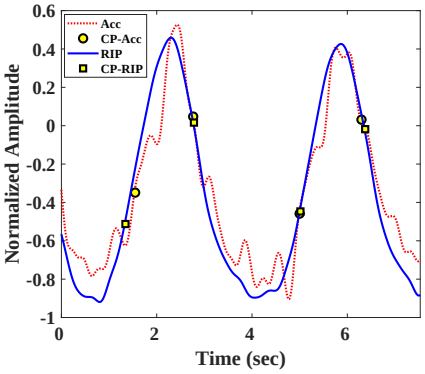


Figure 3: A Typical Example of CP Detection (Both signals are normalized and filtered already.)

in the signal. In practice, this is quite unreliable, mostly because of noise caused by disturbances. We found that a better way to synchronize is to use change points of selected statistical features in the signal. A Change Point (CP) is a point at which a statistical features of a signal, such as mean, standard deviation (STD), root mean square (RMS), changes abruptly. In this study, we compare the effectiveness of change-point detection for three main statistical features: mean, STD, and RMS.

Due to the motion of the wearer, accelerometer data is noisier than RIP data. Hence, some sudden change may lead to a CP in as segment of the accelerometer signal that does not contain a CP in the RIP signal. We pair up common CPs using a two-step process: We first find the respiration cycles based on signal peaks and valleys. Then we find the biggest CPs within each peak-valley cycle. A typical example of CP detection results is shown in Figure 3: The normalized signals collected from both the accelerometer and the RIP sensor are loosely synchronized using the peaks and valleys of the signals. We observe that the CPs of the two signals are close to each other. We describe in Section A.1 how we tolerate small differences between the CPs detected with different signals.

4.3 Quantization

Devices that sample different signals naturally have different levels at which they quantize the sampled values. For pairing such devices with different types of sensors, we need to maximize the number of bits that can be agreed upon by two sensors of different types on each sample, which is also the ultimate goal for pairing.

A typical example of respiration signals collected by a RIP sensor and an accelerometer is shown in Figure 4a. The difference between the two respiration signals is also shown in Figure 4a. Figure 4b shows that the difference can be modeled as a Gaussian distribution. This leads to the following theorem:

Theorem 1. *With the differences between two respiration signals modeled as a Gaussian distribution with the mean μ and standard deviation σ, and the signal values modeled with a uniform distribution in the range $[-\frac{d}{2}, \frac{d}{2}]$, the optimal number of bits per sample, denoted as b, that can maximize the bit agreement on a shared K-bit secure key by the two devices collecting the two respiration signals satisfies the following equation:*

$$b\left(-\frac{\lambda_2}{\lambda_3}+\frac{2\,e^{-\frac{{\lambda_1}^2}{2\sigma^2}}}{2^{2b}\,d\,\sqrt{2\pi\sigma^2}}\right)\left(-\frac{\lambda_2}{\lambda_3}\right)^{\frac{2K}{b}-1}+\ln\left(-\frac{\lambda_2}{\lambda_3}\right)=0\,,$$

where $\lambda_1=\mu-\frac{2}{2^b}$, $\lambda_2=\mathrm{erf}\left(\frac{\lambda_1}{\sqrt{2\sigma^2}}\right)-\mathrm{erf}\left(\frac{\mu}{\sqrt{2\sigma^2}}\right)$, *and* $\lambda_3=2^b\,d$.

In Theorem 1, the function $\mathrm{erf}(.)$ denotes the Gauss error function which is defined as $\mathrm{erf}(z)=\frac{2}{\sqrt{\pi}}\int_0^z e^{-x^2}dx$. The theorem is on b, the number of bits per sample. Usually, the number of quantization levels, denoted as q, can be determined as $q=2^b$.

The proof of Theorem 1 is straightforward. With the modeling on the differences, we can define the objective function as the product of b and the probability of containing the differences between the signals within the same quantization interval, for all the bits in the key K. This optimization can be solved with derivatives [5].

Our user experiments with 30 participants also show that the normalized signal values collected from the sensors can be approximated as a uniform distribution as shown in Figure 5b. According to Theorem 1, the optimal number of quantization levels is four, and correspondingly the number of bits per sample is two.

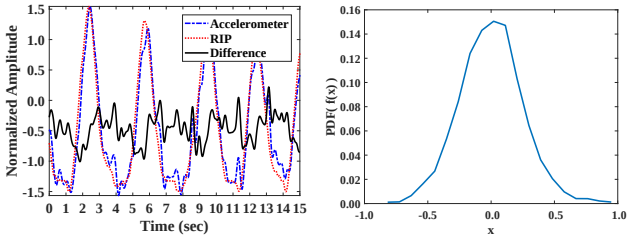
Lloyd-Max Quantization: Given the optimal parameter on the number of quantization levels, we further relax the constraint on the uniform quantizer by allowing quantization intervals of different lengths. The Lloyd-Max quantizer [18] is known for minimizing the quantization error by optimizing the quantization intervals. It finds the best values for quantization intervals and their corresponding quantization levels by considering the boundaries of the quantization intervals as the midpoints of the corresponding quantized values, $a_i=\frac{1}{2}(L_i+L_{i+1}), i=1,...,M-1$, and the quantized values as the centroid of the quantization intervals,

$$L_i=\frac{\int_{a_{i-1}}^{a_i} x f_X(x)dx}{\int_{a_{i-1}}^{a_i} f_X(x)dx}, i=0,...,M-1,$$

where $f_X(x)$ is the probability density function (PDF) of Signal X. In this paper, we assume that the algorithm knows the PDF of signals in advance. The Lloyd-Max quantizer essentially allocates more quantization intervals to those ranges of values with a larger probability of occurrence. We hypothesize that the optimization by Lloyd-Max quantizer can also maximize the number of bits that can be agreed on by two sensors, as more quantization intervals, essentially more bits, are allocated to the ranges with high probabilities of occurrence. Therefore, using the Lloyd-Max quantization, all the samples can be used during the key generation phase, and we can use the change points for synchronization purposes.

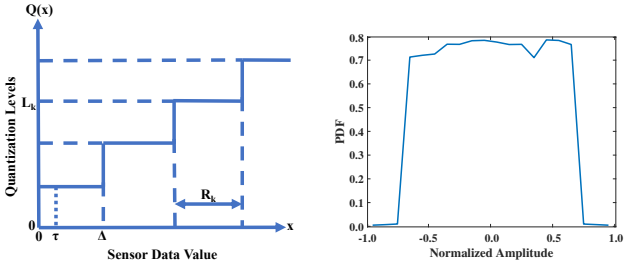
4.4 Error Correction

Both down-sampling and normalization, as well as quantization, can remove differences in biometric signals collected with different sensing mechanisms. Any remaining differences can cause mismatches between bits generated by the quantizers of both devices. So, it is necessary to use some form of error correction to remove bit mismatches. Using a plain error correction code (ECC) not only is vulnerable to data leakage but also adds overhead to the protocol and can decrease entropy. We apply a modified BCH error correction code to address these issues and correct the mismatches.



(a) (b)

Figure 4: (a) Respiration Signals Measured by RIP and Accelerometer Sensor (b) Probability Density Function of Difference Between the Two Signals



(a) (b)

Figure 5: (a) Quantization Levels *vs.* Sensor Data Value, (b) Breathing Signal's Distribution

BCH-based Error Correction: Assume a bit-string of length K. A Bose Chaudhuri Hocquenghem (BCH) code can encode and decode this string using a codeword of length $N = 2^m - 1$ (m is a positive integer and $m \geq 3$) such that it can correct up to t errors, where $t < 2^{m-1}$, and the number of check bits is $N - K \leq mt$ [10]. Unfortunately, the overhead of check bits is too high for resource-constrained devices.

To eliminate this overhead, we modify the BCH method such that a BCH code of length N can carry a message of length N. Our solution (illustrated in Figure 6) works as follows: For the pairing based on the ith CP, *DevA* generates a random bit-string R_i, of length K, and uses a BCH-encoder to encode it into codeword C_i, of length N. Then *DevA* selects Q_{Ai} which is N bits of quantized bits around the ith CP, and generates block B_i by performing an exclusive or (XOR) operation on C_i and Q_{Ai}. It then sends B_i to *DevB* over an unencrypted channel. When *DevB* receives B_i, it selects its Q_{Bi} and performs XOR on B_i and Q_{Bi} to extract C'_i. Then it uses the BCH-decoder to extract R'_i. As long as the number of mismatch bits between Q_{Ai} and Q_{Bi} is less than t, we have $R'_i = R_i$. Afterward, *DevB* uses the BCH-encoder to encode R'_i to C''_i and does XOR on C''_i and B_i to derive Q'_{Ai}. Finally, *DevB* generates a hash denoted as M_i based on Q'_{Ai} and send the hash to *DevA*.

Upon receiving M_i, if *DevA* was successful in verifying hash value M_i based on Q_{Ai}, this means that $Q'_{Ai} = Q_{Ai}$. Therefore, *DevA* uses Q_{Ai} as the common secure key and sends an acknowledgment message to *DevB*. If *DevA* fails to verify hash value M_i based on Q_{Ai}, it will continue with the next pairing attempt. This procedure will be continued until a common key is found or a timeout occurs. For further amplification, the SHA-256 hashing algorithm is applied to the generated key. Wearable devices can run the B2P protocol regularly to generate new secure keys.

We note that the adversary can receive Block B_i by eavesdropping the communication between the two devices. She is not able, however, to derive Q_{Ai} from B_i because she has no knowledge about R_i and the quantized bits in *DevA*. The adversary may guess R_i with a brute force attack and confirm the correct R_i based on B_i and M_i obtained by eavesdropping. So the security strength is essentially K bits for an N-bit key. A desired security strength can be achieved by selecting the appropriate parameters N and K.

4.5 Design and Implementation of the B2P Protocol

The above description naturally gives rise to an architecture of the B2P protocol, which is described in Figure 7. The protocol consists of four main modules. The first module, called *Signal Preprocessing*, is designed to handle the differences in the data collected by different sensing mechanisms and the filtering of the noise outside the respiration frequency range. It implements the necessary down-sampling of the data, the normalization of the amplitudes and the filtering. The *Quantization* module is designed to tolerate the differences (primarily different quantization levels) caused or applied by the differences in sensing mechanisms. The *Synchronization* module uses change-point detection (CPD) methods to detect changes that can be used to synchronize signals collected by the different sensing mechanisms. Finally, the *Error Correction and Confirmation* module corrects the bit mismatches caused by differences that can not be removed by the Quantization module. It uses a variation of the BCH code to do that. This module also contains the mechanism of the shared-key generation, which we described in Section 4.4. The output of the protocol implementation is the shared key.

4.6 Mitigation of Impersonation Attacks

Since breathing activities can be easily observed by an adversary, the use of breathing signals for pairing is vulnerable to impersonation attacks, where a malicious device attempts to pair with a *bona-fide* device. Devices with different sensing mechanisms, are particularly vulnerable, as the filtering needed to improve the agreement on key bits for such devices removes those high-frequency components that are difficult to observe by the attacker. We propose a method to mitigate these attacks by *segmenting* the signals.

The segmenting method divides an output bit string from the quantizer, with respect to a change point, into n_{seg} segments. Instead of using bits from successive segments for pairing, the method may skip neighboring segments and use only bits from selected segments for pairing. The rationale behind this method is as follows: The smoothing effect brought by the filtering can make the next few samples of a sensor signal predictable if the adversary can estimate the value of one sample with reasonable accuracy. This predictability is greatly reduced if the samples are relatively distant from the sample that the adversary may estimate accurately. The best segment choice should contain segments that are distant from each other. In our implementation, when $n_{seg} = 10$ and five segments are needed for pairing, we choose segments with index 2, 5, 6, 7, 10. This selection has proven to be effective in our experiments.

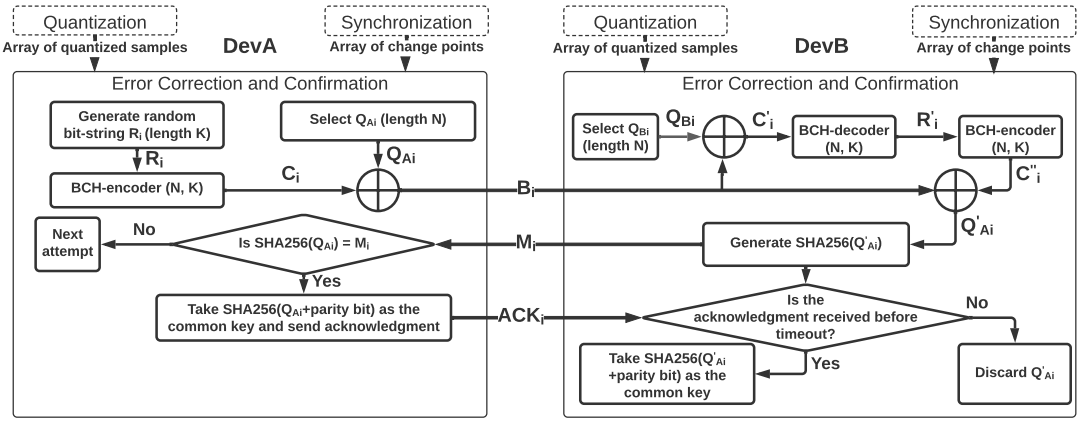


Figure 6: Steps for BCH Error Correction and Confirmation

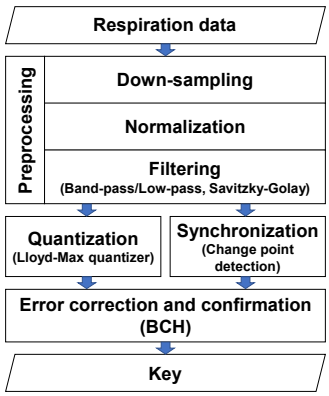


Figure 7: Protocol Overview

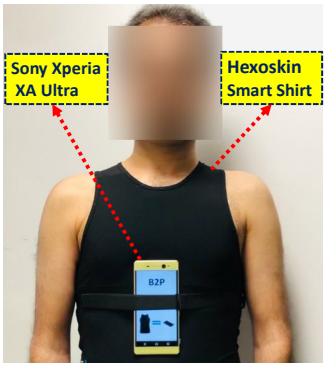


Figure 8: Devices Used in the Experiment

5 EVALUATION

We evaluated the performance of the B2P pairing protocol on a testbed using a Hexoskin smart shirt [2] with an RIP sensor and two smartphones with accelerometers. We note that a similar pairing performance can be obtained from both smartphones, independently of whether the two phones are used independently or bound together. This indicates that the normalization, filtering, and quantization steps can largely tolerate the differences between accelerometers of the phones; this in turn means that a separate sensor calibration step becomes unnecessary. The specifications of the sensors used in pairing are presented in Table 1, which shows that the RIP sensor and the accelerometers sense the respiration activities in very different ways. After receiving approval from the Institutional Review Board (IRB) in our university, we collected data from 30 participants, including 19 males and 11 females, with a mean age of 30.14 years (between 20 and 43). Figure 8 shows the experiment setup. The participants were asked to wear the smart shirt, attach the smartphone to their chest, and sit on a chair. The experiment duration was about 5 minutes per participant.

5.1 Performance Metrics

We evaluate the pairing performance using the following metrics:

Key Generation Rate: This metric is defined as the number of keys generated per second. Practical systems often use keys with a length of 128 or 256 bit. We, therefore, focus on these two key lengths in these evaluations. In general, a higher rate is desired.

Entropy of Generated Keys: This metric is defined as a measurement of "surprise" contained in the generated keys by measuring the entropy [15]. Given a randomly generated key W, its entropy can be computed as $H(W) = -P(w_0)\log_2 P(w_0) - P(w_1)\log_2 P(w_1)$, where $P(w_0)$ and $P(w_1)$ are the probability of zeros and ones in W, respectively. The ideal entropy for generated keys is 1.

False Negative Ratio: This metric indicates the probability that two legitimate devices attached to the same body can not pair successfully. FNR is computed as $FNR = \frac{FN}{EP}$, where FN is the number of incorrectly missed keys, and EP is the number of expected successful attempts.

False-Positive Ratio: We define the false-positive ratio as the percentage of pairing attempts that incorrectly generate common keys among all the expected unsuccessful attempts. The metric indicates the likelihood of successfully pairing with a legitimate device by the adversary. The FPR can be computed as $FPR = \frac{FP}{EN}$, where FP is the number of incorrectly generated keys, and EN is the number of expected unsuccessful attempts. The metric is used to measure resistance to impersonation attacks described in Section 5.3.

Table 1: Sensors Specifications

Sensor	Device	Sensing Mechanism	Sampling Rate (Hz)	Resolution (bit)	Unit
RIP	Hexoskin Smart Kit	Measuring pulmonary ventilation	128	16	mL
3-axis Accelerometer	Sony Xperia XA Ultra	Measuring acceleration forces	100	13	G
	HTC One E9 Plus				

Table 2: Achievable Key Generation Rates

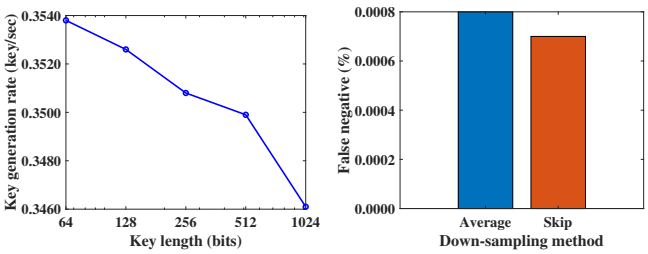
Performance Metric	128-bit Key	256-bit Key
Key Generation Rate (key/sec)	0.3526	0.3508
Entropy	0.99	0.97
False Negetive Ratio (%)	0.0003	0.0007
Bit Agreement Ratio (%)	64	60

Bit Agreement Ratio (BAR): We define this metric as the percentage of matched bits in bit strings generated by two devices. We use BAR to evaluate B2P's resistance to impersonation attacks, such as those described in Section 5.3. Since a successful pairing attempt needs a 100% BAR, the metric indicates how close an adversary is to pair with a legitimate device.

Table 2 shows all the performance metrics used in this study and the best B2P results.

5.2 Impact of Parameters and Methods Choices

B2P can generate about 0.35 keys per second across a variety of key lengths, as shown in Figure 9a. The key generation rate decreases slightly for longer keys, as pairing attempts become more likely to fail because of the sensing noise and the difference in sensing mechanisms. B2P usually makes pairing attempts at detected change points. So the number of pairing attempts does not change with the key length. For instance, a 1024-bit key can be based on a signal fragment with three respiration cycles, and there can be five CPs in the signal fragment. As a result, the 1024-bit keys generated at these 5 CPs will have some overlap. Table 3 lists the parameters and methods used to generate the results described above. In the following experiments, we study the impact of the parameters in Table 3 by varying the parameters that are being addressed in each experiment and keeping the rest of the parameters constant. Experiment results on syncrhonization parameters, quantization parameters, error correction, and key randomness can be found in Appendix A.1, A.2, A.3, and A.4 respectively.



(a) (b)

Figure 9: (a) Effect of Different Key Length on KGR, (b) Impact of Down-Sampling on False Negative Ratio

Table 3: Typical Parameters

Parameter	Value
Key length (bits)	256
BCH parameters (N,K)	(255, 115)
Number of bits per sample	2
Sync offset (ms)	250
Number of CP offsets	2
CP num per peak-valley	1
Code type	Gray
CPD method	STD
Down-sampling	skip

Table 4: Impact of Down-Sampling on Performance Metrics

Downsampling	Average	Skip
Key generation rate (key/sec)	0.2385	0.2382
Entropy	0.94	0.96
False negative (%)	0.0008	0.0007

5.2.1 ***Impact of Down-Sampling.*** In this experiment, we compare the two down-sampling methods: skip and average. We know that the RIP sensor has a much higher sampling rate than the accelerometers. The RIP data must therefore be down-sampled.

Table 4 illustrates that the key generation rates for both methods are very close. We believe this happens because the low-pass and band-pass filters used in the filtering both have a cut-off frequency much lower than the sampling rates. As a result, the differences are largely filtered out.

5.2.2 ***Impact of Filtering.*** Choosing an effective filter can significantly affect the system's performance, as it removes undesired frequencies and noises. Figure 10a shows how, in comparison with a low-pass filter, a band-pass filter with cut-off frequency in the range of (0.1, 0.5) causes a significant improvement of the KGR. The reason is that the accelerometer signal contains a lot of external disturbance that is due to the user's motion. Since most of the user motion happens at low frequency, the band-pass filter removes most of it, while the low-pass filter does not affect the user's low-frequency motions. Most of the motion noise occurs at the end of the exhalation phase (the valleys in Figure 4a) because the diaphragm usually has a short stop there [8]. This stop point affects the detected signal of both devices. The band-pass filter also smoothens these points and leads to an increase in entropy, as shown in Figure 10b. However, the FNR does not change by changing the filtering method.

As Table 5 shows, the use of a Savitzky-Golay filter generally improves all the performance metrics because it removes undesired dynamics that the band-pass filter can not.

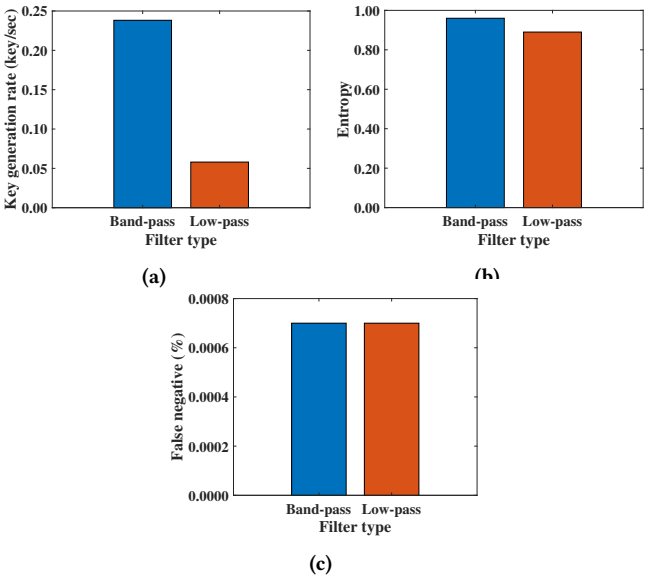


Figure 10: Impact of Filtering on Performance Metrics (a) Key Generation Rate, (b) Entropy, (c) FNR.

Table 5: Impact of Applying the Savitzky-Golay Filter

	Without Savitzky-Golay filter	With Savitzky-Golay filter
Key generation (key/sec)	0.2382	0.2532
Entropy	0.88	0.96
False negative (%)	0.0008	0.0007
Bit agreement (%)	59.1	58.66

5.3 Resistance to Attacks

An adversary may try to compromise pairing with the B2P protocol by launching impersonation attacks, which attempt to pair a malicious device with a legitimated device. We evaluate B2P's resistance to impersonation attacks with a third party's signals and impersonation attacks with remote biometric signal collection below.

Resistance to Impersonation Attacks with a Third Party's Signals It is easy to imagine scenarios where the biometric signals of a third party may lead to a third-party device to establish keys in an unauthorized fashion. This may be the case for a device of a random nearby person with a similar breathing pattern, or for a device of an adversary who attempts to pair with a victim's device with a third party's respiration signals. We performed a series of experiments to measure what the probability is to have a device erroneously pair based on another person's respiration signals. In these experiments, the typical parameter setting in Table 3 is used, and we vary the number of bits per sample. Our results show zero false positive rates. In other words, the adversary cannot generate any key with a third party's signal. To find out how close the adversary is to a successful pairing with a legitimate device through this form of impersonation attack, we measure the *bit agreement rate*, defined as the percentage of matched key bits. Since different users have different breathing rates and volume, and due to the lack of synchronization between them, the maximal bit agreement rate is 60.09%, as shown in Figure 11.

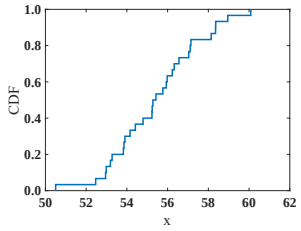
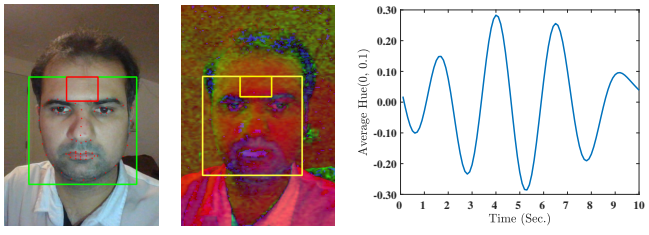


Figure 11: CDF of Bit Agreement Rate with 3rd-Party Signal Impersonation

Resistance to Impersonation Attacks with Remote Biometric Signal Collection We assume that the adversary is able to remotely extract respiration signals, for example using video analysis of the victim's breathing behavior. Similar impersonation attacks based on video analysis have been studied in [17, 23]. In [21], the authors demonstrated that one can monitor the respiration and heart-beat signals from a video. They demonstrated the accuracy of their method by comparing it with the RIP-based method, where the video-based approach achieved a low root mean square error (RMSE). We replicated the implementation in [21]. Our results show that this method can accurately estimate respiration rates with an average of 8.95% estimation error. Given these accurate estimation results, we extended the method for impersonation attacks based on video analysis. The proposed method detects the user's face and facial landmarks using the histogram of oriented gradients (HoG) features with the help of a support vector machine (SVM). Then, the method determines the region of interest (ROI), which is a part of the user's forehead, based on the facial landmarks, as shown in Figure 12a. Afterward, as shown in Figure 12b, it converts the frame to hue saturation value (HSV) color space and calculates the average hue of all ROI pixels. The variation in the average values is due to the heart-beat, respiration, and changes in ambient. A band-pass filter with a cut-off frequency in the respiration frequency range $(0.1Hz, 0.5Hz)$ is applied on the array of the average values to extract the respiration signals, as shown in Figure 12c. The adversary now can launch the impersonation attack to pair with a legitimate device attached to the victim.



(a) (b) (c)

Figure 12: Video Attack Procedure (a) Forehead Detection Based on Face Detection and Facial Landmarks, (b) HSV Color Space, (c) Detected Respiration Signal

In our experiments on resistance to the impersonation attack with video analysis, a video camera of resolution 640 ∗ 480 pixels

and capture rate of 30 RGB frames per second is used to take a video of the victim's face. We use the method described above to extract the victim's respiration signal. Since the method extracts the respiration signal in real-time, it loses some frames due to the processing time. Our results show that the method can process nine frames per second to extract respiration signals. Hence, the extracted signal will lose some information of the actual respiration signal. The extracted signals are also affected by ambient light and the tone of the user's skin color, as described in [21].

Our results show that the FPR of this impersonation attack is zero. This means that the adversary can not generate a shared key. Similarly to the case of impersonation based on 3rd-party signals, we use the bit agreement rate to measure the effectiveness of this attack. As shown in Figure 13, the maximal bit agreement rate is 61.56%.

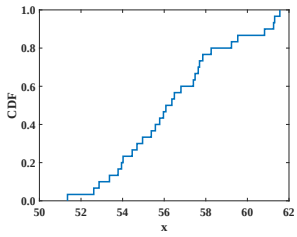


Figure 13: CDF of Bit Agreement Rate with Video-Based Impersonation

This is because the extracted signal from the user's video cannot preserve the original signal's features, such as change points. Hence, the signal generated for the video-based impersonation attack cannot be synchronized to the victim's signal. The loss of synchronization causes too many mismatches for the error correction method to correct.

6 SECURITY ANALYSIS

A recent study by Hamme *et al.* [24] shows that an adversary can reduce the complexity of brute force attacks with knowledge of biometric distributions. By sampling from the biometric distribution, the adversaries, also called biometric hackers, can launch two stronger attacks: 1) biometric binary representation statistical attack and 2) hash preimage statistical attack. Both attacks utilize biometric samples statistics to decrease the complexity of brute force attacks. The complexity of the first attack, denoted as $C^{\beta-Bi}_{bio-hacker}$, can be theoretically estimated according to Equation 5 in [24] with the assumption of the β-binomial distribution of bit mismatches. The empirical values for this complexity, denoted as $C^{emp}_{bio-hacker}$, can be obtained based on Equation 2 in [24]. The authors demonstrated that the theoretical bound of the biometric hacker hash preimage attack, denoted as $C^{hash-preimage}_{bio-hacker}$, can be computed by Equation 6 in [24].

Similar stronger attacks can be launched to attack the pairing protocol if the biometric hacker has knowledge of biometric distributions. We evaluate the complexity of these kinds of attacks based on Equation 2, 5, and 6 in [24] with the data collected for the experiments described above. In order to show the worst case, we choose respiration data from the dataset closest to a victim's respiration pattern and the adversary is assumed to launch the attacks with the chosen data. We also manually synchronize the adversary's data with the vicitm's data for the worst case. Table 6 shows the results of the security analysis with the attacks. The results indicate significant reduction of complexity due to the attacks. The complexity decreases as error correction capability increases because the error correction can also help the biometric hackers establish keys with the victim. Therefore, the security strength in case of biometric attacks is significantly lower than the key length.

7 DISCUSSION

To the best of our knowledge, this paper is the first study that uses the respiration signal for pairing. We compare the B2P results with other biometric-based pairing approaches, such as heartbeat-based or gait-based approaches. As Table 7 shows, the B2P protocol can generate a 128-bit key per 2.83 seconds and a 256-bit key at 2.85 seconds while the bit agreement for the attacker tops out at 64% and 60%, respectively. In comparison, H2B [17], a *heart-beat based* pairing protocol based on the interval between two peaks in the heart-beat signal (inter-pulse interval (IPI)) needs more than 40 seconds to generate a 128-bit key, and its bit-agreement rate is 75%. The authors in [35] proposed a *shaking-based* pairing approach that generates a shared key every 6.58 seconds. The authors did not report on the attacker's bit agreement rate. Although the aforementioned studies used signals that have higher frequencies than the respiration signal, their key generation rates are less than our protocol.

Dealing with different types of sensors to generate a shared key is a main contribution of this paper. We showed how it is possible to use RIP and accelerometer sensors to generate a shared secure key in only one breathing cycle. RIP sensors have been embedded in various smart devices including smart shirts such as Athos Shirt [1], smart bras, smart straps, and IMDs. With the increasing popularity of the smart wearable devices, we believe the B2P protocol can enable more devices to pair based on respiration.

The B2P protocol can also be extended for pairing based on other types of physiology processes, especially for devices sensing the same physiology process with different sensing mechanisms. For example, the cardiac process can be sensed with both the traditional photoplethysmography (PPG)-based sensing mechanism and the recently proposed acoustic sensing mechanism [28]. The B2P protocol can potentially be extended to pair devices equipped with these two different types of sensors.

8 RELATED WORK

The use of biometric activities has become a popular approach for pairing wearable devices. A number of biometric activities have been studied, such as heart-beat, shaking, and gait, and a variety of sensor technologies have been used to observe (i.e., collect signals of) these activities. These sensors include ECG, PPG, EMG, and accelerometer. Indeed, these signals can provide auxiliary out-of-band (OOB) channels [19] as a feasible option to facilitate device pairing. In this section, we provide a summary of studies that used the aforementioned signals. As far as we know, this study is the first in using respiration activity for the pairing of wearable devices.

Table 6: Security Analysis with Different Error Correction Capability (N=255)

Error correction capability (bit)	K	$C^{\beta-Bi}_{bio-hacker}$	$C^{emp}_{bio-hacker}$	$C^{hash-preimage}_{bio-hacker}$
21	115	12.41	8.35	6.34
13	155	16.31	10.13	8.55
10	179	18.49	11.56	9.87
9	187	19.37	12	10.31

Table 7: B2P Performance *vs.* Other Pairing Methods

Scheme		**Key Length (bit)**	**User KGR (key/sec)**	**Attacker bit agreement (%)**
H2B	[17]	128	0.0239	75
Shake Me	[35]	128	0.1520	?
B2P	This paper	128	0.3526	64
B2P	This paper	256	0.3508	60

ECG and PPG: The heart-beat is a promising option for wireless body area networks (WBANs) authentication and key generating schemes because its properties are unique, and their features differ from person to person [30]. Heart-beat signals can be easily collected, and they are hard to copy by other people in comparison to simple PIN codes. Heart-beat signals can usually be collected by ECG and PPG sensors. ECG sensors collect the electrical activity of heart muscles through electrodes attached to the body. PPG sensors detect the blood level transforms in the microvascular cot of tissue.

Various features extracted from heart-beat signals can be used for authentication and key generating purposes. The most important feature used in WBANs is heart rate variability (HRV) [14], indicating the fluctuations of heart rate around an average rate.As has been demonstrated [23], HRV is highly random and can be used as a random source to generate keys.

Rostami *et al.* [23] proposed an HRV-based pairing method to authenticate external medical device controllers and programmers to IMDs. The authors introduce a touch-to-access policy using a time-varying physiological value (PV) by ECG readings. They utilized statistical characterization of ECG for pairing wearable devices. Another pairing system called H2B is presented by Lin *et al.* [17], which utilizes piezo sensors to detect heart-beat signals and generate a secret key.

EMG: The EMG or electromyogram signals are the electrical signal generated by contractions of human muscles. According to medical research [20], the EMG signal is a quasi-random process, i.e., the average value of EMG is correlated to the generated force of the muscle, but it has a random amplitude variation under a given force. In other words, there are stochastic variations of EMG amplitude for a unique gesture and force. Therefore, the EMG signals can be used as a secure source to generate secret keys in physically close contact for some wearable devices like Myo armband [3]. Since detecting this kind of signal needs physical contact in proximity, it is extremely difficult for an adversary to perform an eavesdropping attack. EMG-KEY is an EMG-based method proposed by Yang *et al.* [34] which leverages EMG variation signal to generate a secret key for pairing two wearable devices.

Gait: Muscular-skeletal structures vary across individuals, and so do gait patterns [36]. Hence, one can determine if two devices are carried by the same person [16]. Various techniques exploit different features of gait to generate a common key for pairing wearable devices. Sun *et al.* [29] proposed a method to generate a symmetric key based on the timing information of gait. The authors used the IPI of consecutive gait as a common feature between the two devices. Schürmann *et al.* [27] presented a secure spontaneous authentication scheme that exploits correlation in acceleration sequences from devices worn or carried together by the same person to extract always-fresh secure secrets. Their method, BANDANA, utilizes instantaneous variations in gait sequences. Walkie-Talkie [33] exploits users' gait for shared secret key generation. The authors use independent component analysis (ICA) for blind source separation (BSS) to separate accelerometer signals from different body movements such as arm swing and walk.

In contrast to the aforementioned studies, our protocol, B2P, is the first one to utilize the breathing signal to enable secure pairing in wearable devices.

9 CONCLUSION

We presented the design, implementation, and evaluation of the Breathe-to-Pair (B2P) protocol, a respiration-based pairing approach for wearable devices. B2P enables generating the same secure keys based on the user's breathing signal detected by different types of sensors, RIP and accelerometer, in independent wearables. We explored several techniques to address the challenges in pairing distinctive wearables that use different types of sensors. B2P utilizes a change point detection technique for synchronizing independent wearables. It uses a Lloyd-Max quantization and an improved BCH error correction method to optimally digitize the respiration signal and correct mismatches between bit-strings. Extensive experiments on 30 participants indicate that B2P can generate a secure 256-bit key every 2.85 seconds, which is about one breathing cycle. We also proposed a sparse segment selection method to prevent the adversary. Experiment results show that B2P is secure against different types of attacks.

10 ACKNOWLEDGEMENTS

We thank our shepherd Dave Singelee and the anonymous reviewers for their feedback to improve this research. A special acknowledgment to Enrique Argones Rúa for help on the security analysis.

REFERENCES

[1] [n.d.]. Athos Shirt. liveathos.com.
[2] [n.d.]. Hexoskin smart shirt. hexoskin.com/collections/all/products/hexoskin-smart-kit-mens.
[3] [n.d.]. Myo armband. myo.com.

[4] [n.d.]. Respiratory system, Wikipedia, The Free Encyclopedia. en.wikipedia.org/wiki/Respiratory_system.

[5] Edwin KP Chong and Stanislaw H Zak. 2004. *An introduction to optimization*. John Wiley & Sons.

[6] Christian F Clarenbach, Oliver Senn, Thomas Brack, Malcolm Kohler, and Konrad E Bloch. 2005. Monitoring of ventilation during exercise by a portable respiratory inductive plethysmograph. *Chest* 128, 3 (2005), 1282–1290.

[7] Marco Di Rienzo, Giovannibattista Rizzo, Zeynep Melike Işilay, and Prospero Lombardi. 2020. Seismote: A multi-sensor wireless platform for cardiovascular monitoring in laboratory, daily life, and telemedicine. *Sensors (Switzerland)* 20, 3 (2020). https://doi.org/10.3390/s20030680

[8] Colin Anthony Francis, Joaquín Andrés Hoffer, and Steven Reynolds. 2016. Ultrasonographic evaluation of diaphragm thickness during mechanical ventilation in intensive care patients. *American J. Critical Care* 25, 1 (2016).

[9] G. George. 2017. *Device-to-device Communication and Wearable Networks: Harnessing Spatial Proximity*. Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra. https://books.google.com/books?id=w2QlygEACAAJ

[10] Yunghsiang S Han. 2010. BCH codes. *Graduate Institute of Communication Engineering, National Taipei University Taiwan* (2010).

[11] Mojtaba Jafari Tadi, Tero Koivisto, Mikko Pänkäälä, and Ari Paasio. 2014. Accelerometer-based method for extracting respiratory and cardiac gating information for dual gating during nuclear medicine imaging. *Internat. J. Biomed. Imaging* 2014 (2014). https://doi.org/10.1155/2014/690124

[12] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. 2015. Magpairing: Pairing smartphones in close proximity using magnetometers. *IEEE Trans. Inform. Forensics and Security* 11, 6 (2015), 1306–1320.

[13] F. John Dian, Reza Vahidnia, and Alireza Rahmati. 2020. Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey. *IEEE Access* 8 (2020), 69200–69211. https://doi.org/10.1109/ACCESS.2020.2986329

[14] Wahiba Ben Abdessalem Karaa. 2015. *Biomedical image analysis and mining techniques for improved health outcomes*. IGI Global.

[15] George J Klir and Mark J Wierman. 2013. *Uncertainty-based information: elements of generalized information theory*. Vol. 15. Physica.

[16] Jonathan Lester, Blake Hannaford, and Gaetano Borriello. 2004. "Are you with me?" - Using accelerometers to determine if two devices are carried by the same person. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3001 (2004), 33–50. https://doi.org/10.1007/978-3-540-24646-6_3

[17] Qi Lin, Weitao Xu, Jun Liu, Abdelwahed Khamis, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. 2019. H2B: Heartbeat-based secret key generation using piezo vibration sensors. *arXiv* (2019), 265–276.

[18] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Informat. Theory* 28, 2 (1982), 129–137.

[19] Rene Mayrhofer, Jürgen Fuß, and Iulia Ion. 2012. UACAP: A unified auxiliary channel authentication protocol. *IEEE Trans. Mobile Comp.* 12, 4 (2012).

[20] Roberto Merletti and Philip J Parker. 2004. *Electromyography: physiology, engineering, and non-invasive applications*. Vol. 11. John Wiley & Sons.

[21] Jafar Pourbemany, Essa Almabrok, and Ye Zhu. 2021. Real Time Video based Heart Rate and Respiration Rate Monitoring. *arXiv preprint arXiv* (2021).

[22] S. P. Preejith, Ahamed Jeelani, Paresh Maniyar, Jayaraj Joseph, and Mohanasankar Sivaprakasam. 2017. Accelerometer based system for continuous respiratory rate monitoring. *2017 IEEE International Symposium on Medical Measurements and Applications, MeMeA 2017 - Proceedings* (2017), 171–176. https://doi.org/10.1109/MeMeA.2017.7985870

[23] Masoud Rostami and Ari Juels. 2013. Authentication for Implanted Medical Devices Categories and Subject Descriptors. *Ccs* (2013), 1099–1111.

[24] Enrique Argones Rúa, Davy Preuveneers, Wouter Joosen, et al. 2021. On the Security of Biometrics and Fuzzy Commitment Cryptosystems: A Study on Gait Authentication. *IEEE Transactions on Information Forensics and Security* 16 (2021), 5211–5224.

[25] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. 2001. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Technical Report. BAH, Inc. McLean, VA.

[26] Shourjya Sanyal and Koushik Kumar Nundy. 2018. Algorithms for monitoring heart rate and respiratory rate from the video of a user's face. *IEEE Journal of Translational Engineering in Health and Medicine* 6 (2018), 1–11.

[27] Dominik Schürmann, Arne Brüsch, Ngu Nguyen, Stephan Sigg, and Lars Wolf. 2018. Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing. *Perv. and Mobile Comp.* 47, May 2017 (2018), 1–12. https://doi.org/10.1016/j.pmcj.2018.03.006

[28] Piyush Sharma, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. 2019. Acoustic Sensing as a Novel Wearable Approach for Cardiac Monitoring at the Wrist. *Scientific Reports* 9, 1 (27 Dec 2019), 20079. https://doi.org/10.1038/s41598-019-55599-5

[29] Yingnan Sun, Charence Wong, Guang Zhong Yang, and Benny Lo. 2017. Secure key generation using gait features for Body Sensor Networks. *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2017* (2017), 206–210. https://doi.org/10.1109/BSN.2017.7936042

[30] Kasun Maduranga Silva Thotahewa, Jean Michel Redouté, and Mehmet Rasit Yuce. 2014. *Ultra wideband wireless body area networks*. Vol. 9783319052878. 1–173 pages. https://doi.org/10.1007/978-3-319-05287-8

[31] JA Unar, Woo Chaw Seng, and Almas Abbasi. 2014. A review of biometric technology along with trends and prospects. *Pattern recognition* 47, 8 (2014).

[32] Erik Vanegas, Raul Igual, and Inmaculada Plaza. 2020. Sensing systems for respiration monitoring: A technical systematic review. *Sensors (Switzerland)* 20, 18 (2020), 1–84. https://doi.org/10.3390/s20185446

[33] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2016. Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication. In *2016 15th ACM/IEEE Internat. Conf. on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

[34] Lin Yang, Wei Wang, and Qian Zhang. 2016. Secret from muscle: Enabling secure pairing with electromyography. *Proceedings of the 14th ACM Conference on Embedded Networked Sensor Systems, SenSys 2016* (2016), 28–41. https://doi.org/10.1145/2994551.2994556

[35] Hidir Yuzuguzel, Jari Niemi, Serkan Kiranyaz, Moncef Gabbouj, and Thomas Heinz. 2015. ShakeMe: Key Generation from Shared Motion. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing* (Oct 2015). https://doi.org/10.1109/cit/iucc/dasc/picom.2015.316

[36] Yu Zhong and Yunbin Deng. 2014. Sensor orientation invariant mobile gait biometrics. In *IEEE international joint conference on biometrics*. IEEE, 1–8.

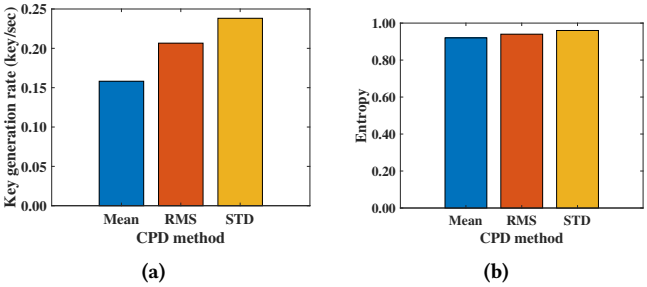


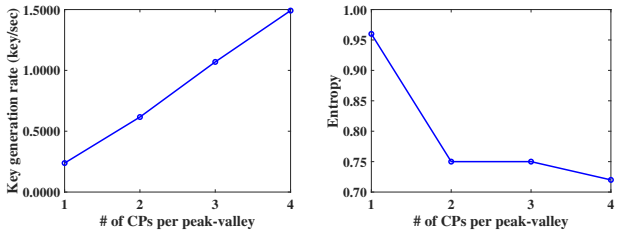
Figure 14: Effectiveness of CPD Method *vs.* Key Generation Rate and Entropy

A DETAILED EXPERIMENTAL RESULTS

A.1 Impact of Synchronization

Synchronizing the signals collected from independent devices is always challenging for pairing [7, 12]. This is more difficult when pairing with sensors with different sensing mechanisms. As described in Section 4.2, we utilize the change-point detection (CPD) method to detect the significant changes in the signal observed by both devices and, we use change points to synchronize the two signals. We experimented with five parameters that affect the detected change points and pairing performances: CPD method, number of change points per peak-valley, CPD threshold, CP offset, and synchronization offset.

CPD Method: Since the respiration signal is a low-frequency signal, even small changes in its frequency can be detected using the STD-based CPD. Its advantages over other statistical metrics are shown in the Figures 14a and 14b. In the rest of the experiments on CPD parameters, we focus on STD-based CPD only. **Number of Change Points per Peak-Valley:** To select a change point in a peak-valley duration, the B2P first detects all the change points in that duration, then sorts them and selects the first n_{CP} most significant CPs. Figures 15a and 15b show that although increasing



(a) (b)

Figure 15: Number of CP per Peak-Valley *vs.* Key Generation Rate and Entropy

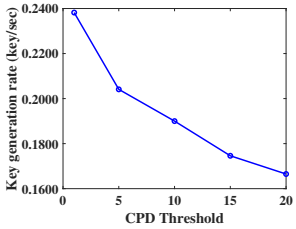
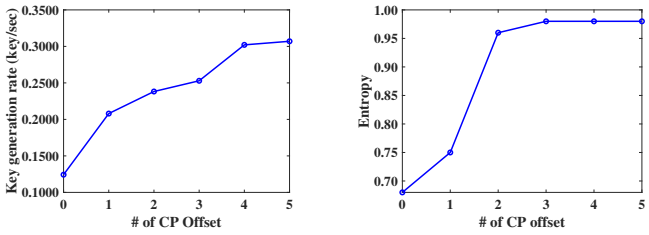


Figure 16: CPD threshold *vs.* Key Generation Rate



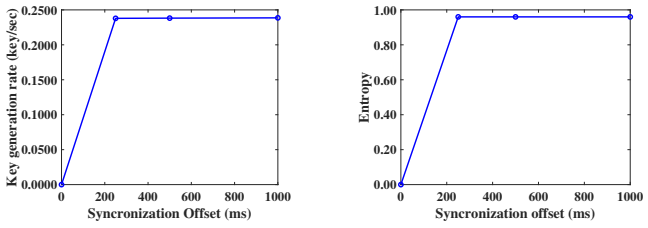
(a) (b)

Figure 17: CP Offset *vs.* Key Generation Rate and Entropy

the number of change points (n_{CP}) can increase the KGR significantly, it biases the final key and decreases its entropy in a way that is not reliable as a secure key. **CPD Threshold Level:** By changing the CPD threshold, we can control the number of change points. Increasing the CPD threshold leads to the loss of some possible synchronization points. Therefore, the KGR will decrease, as shown in Figure 16. However, it reduces the chance of the adversary generating a common key.

CP Offset: Another important parameter for synchronization is the offset of change points, which is the number of CPs adjacent to the selected CP that are allowed to be considered as an alternative to the selected CP. We denote this as n_o. A non-zero offset allows for some inconsistency between change point detection on the two devices. For example, if one device misses a change point, future pairing attempts can be based on change points that are out of sync. The parameter n_o specifies the number of neighboring change points are used for pairing attempts. A value of 1 for n_o means that both the previous and the next change point are used for pairing attempts. Figures 17a and 17b show that increasing the CP offset Number can significantly increase the KGR and entropy.

Synchronization Offset: In order to have more control over the synchronization and pairing process, we also parameterize the synchronization offset. This parameter specifies how much lag or lead between two signals is accepted. This can be checked at the confirmation stage by sending the CP's index in which the common key has been generated. Using the generated key, *DevB* encrypts its CP's index and sends it to *DevA*. *DevA* decrypts it and checks whether it is within the range of the synchronization offset. If it is, *DevA* will accept the key as a shared key. Experimental results in the Figures 18a and 18b show that increasing this parameter until 250 msec can significantly raise the KGR and entropy. Increasing the synchronization offset beyond 250 msec will only increase the attacker's success rate.



(a) (b)

Figure 18: Synchronization Offset *vs.* Key Generation Rate and Entropy

A.2 Impact of Quantization

In this set of experiments, we vary the number of bits per sample and experiment on the bit representation types. Figure 19a shows that the key generation rate decrease when the number of bits per sample increases. The maximal key generation rate is achieved at 2 bits per sample. The result is consistent with our theoretical results presented in Section 4.3. Figure 19b shows the decrease in entropy when the number of bits per sample increases. The decrease is because fewer samples are needed to generate a key of a fixed length and because consecutive samples generally generate key bits with less randomness. Figure 19c shows that the false negative rate increases with the number of bits per sample. More bits per sample lead to more quantization levels and, in turn, to a reduction of the distance between two successive quantization levels. As expected, an increase of the number of bits per sample leads to less tolerance of sensing noise or other differences caused by the different sensing mechanisms in the devices. Figure 19c shows that false negative rates are less than 0.0011%.

Figure 20a and Figure 20b compare the results for Gray coding and binary coding used to represent bits generated by the quantizer. Figure 20a shows that Gray coding can achieve higher key-generation rates than binary coding. The advantage is mainly because of the error correction capabilities of Gray coding [17].

A.3 Impact of Error Correction

A key parameter of the error correction and conformation module is the error correction rate (ECR), defined as the ratio between the number of errors that can be corrected and the length of a codeword in bits. B2P uses BCH mainly to correct mismatches in key bits

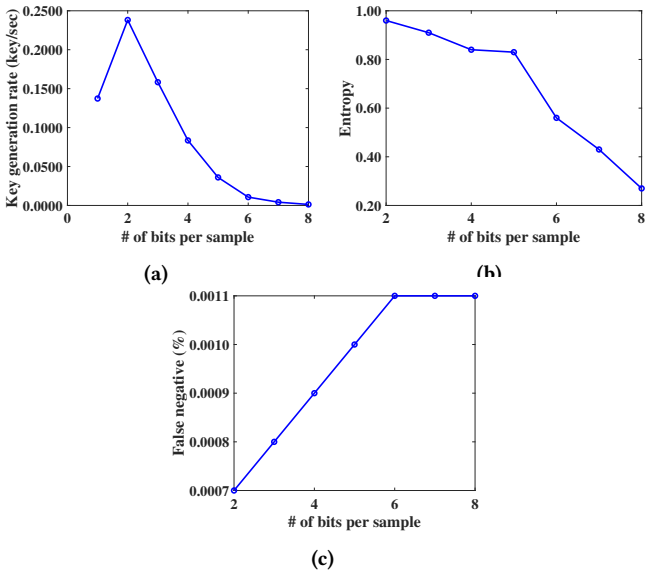


Figure 19: Impact of Quantization on Performance Metrics. (a) Number of Bits per Sample *vs.* Key Generation Rate, (b) Number of Bits per Sample *vs.* Entropy, (c) Number of Bits per Sample *vs.* FNR

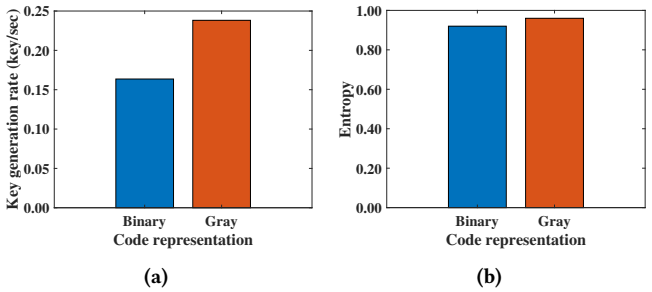


Figure 20: Code Representation Type *vs.* Key Generation Rate and Entropy

between two devices to be paired. So the error correction rate directly affects B2P's key generation rate. For 127-bit codewords and 255-bit codewords, *ECR*s are in the ranges of between 0.79% and 24.4% and between 0.39% and 24.7%, respectively.

As we expected, the key generation rates, shown in Figure 21a, increase with *ECR* simply because more bit mismatches can be resolved when *ECR* increases. The low key-generation rates for both 128-bit and 256-bit keys when $ECR < 1\%$ indicate that the mismatches are very unlikely to be removed completely with low-*ECR* BCH codes. Figure 21b shows that B2P can generate 128-bit keys and 256-bit keys with entropy close to 1, the ideal entropy, when $ECR \geq 5\%$ and $ECR \geq 8\%$, respectively. Figure 21c shows the decrease of FNR when *ECR* increases. This decrease is consistent with our intuition, as larger *ECR* can resolve more bit mismatches.

We did experiment with other lengths as well. Since bit-string selection is based on the CPs, the KGR relies on the number of CPs: the more CPs in a specific signal duration, the higher the KGR. On

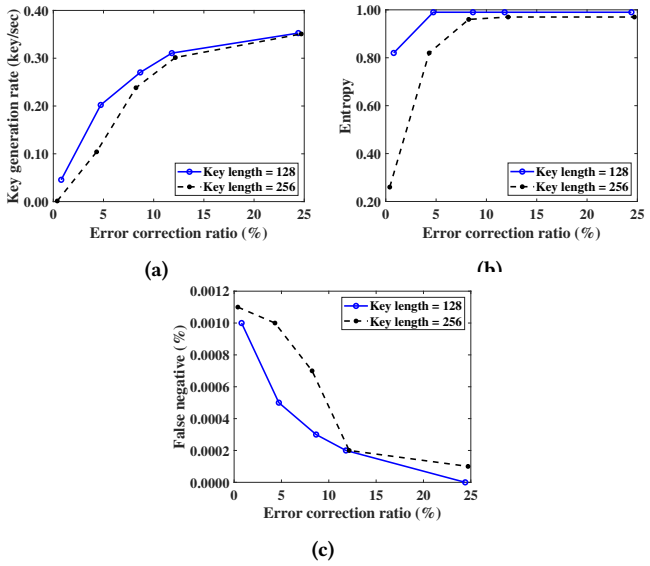


Figure 21: Impact of Error Correction on Performance Metrics. (a) ECR *vs.* Key Generation Rate, (b) ECR *vs.* Entropy, (c) ECR *vs.* FNR

Table 8: Randomness Results of NIST Statistical Test Suite

NIST test	p-value
Approximate entropy test	0.989514
Cumulative sums forward	0.896542
Cumulative sums reverse	0.754125
FFT test	0.194852
Frequency	0.235263
Linear complexity	0.935244
Longest runs	0.553284
Non overlapping Template	0.214827
Runs	0.425865

the other hand, using band-pass and Savitzky-Golay filters, we have relatively smooth signals with some mismatches that can be corrected in the error correction step. Hence, although increasing the key length can decrease the KGR in principle, the effect in practice is not significant, as exemplified in Figure 9a. Using low-frequency biometric activities brings fundamental trade-offs between key generation rate and key length. For example, for breathing sensors with a sample rate of 64 Hz, a low-frequency respiration signal (around 0.35 Hz), and a quantizer that represents each sample with 2 bits, key lengths lower than 128 bits can not have a high entropy rate. This is because the keys are generated from a small number of samples (there are 128 bits per second, which makes for about one-third of a respiration cycle). Also, key lengths greater than 256 bit may have significant overlap with consecutive keys. For instance, a key length of 1024 bits covers around three respiration cycles, which has at least six CPs.

A.4 Key Randomness

In addition to the entropy metric defined in Section 5.1, we also use the widely-accepted NIST Statistical Test Suite [25] to evaluate the randomness of the key bits. Table 8 shows the testing results . All the tests results yield a p value significantly larger than the 1% threshold, meaning that the randomness in the key bits is high.