AudioKey: a usable device pairing system using audio signals on smartwatches

Jiacheng Shang* and Jie Wu

Department of Computer and Information Sciences,
Temple University,
Philadelphia, PA, USA
Email: jiacheng.shang@temple.edu
Email: jiewu@temple.edu
*Corresponding author

Abstract: Smartwatches are expected to replace smartphones in some applications with better user experience because of a greater range of features and new innovations such as audio recording, activity recognition, and data transmission. In this paper, we develop a system called AudioKey, aiming to pair two smartwatches by generating a unique secret key between them. Compared with existing works, our system does not need extra infrastructure to synchronise devices and trigger the key generation process, and only uses the existing sensors (gyroscope and microphone) that are deployed on most smartwatches. AudioKey triggers the key generation process on two devices at the same time by detecting the handshake between two normal users. A secret key is extracted from both the frequency domain and the time domain of audio signals and used to authenticate each other or encrypt the sensitive data. Evaluation results collected on nine volunteers in three different scenarios show that our system can achieve a bit generation rate of 13.4 bits/s with the mean key agreement rate of 96.7% for a 128-bit secret key, while a strong attacker can only achieve a mean key agreement of 10.8%.

Keywords: human activity recognition; secret key generation.

Reference to this paper should be made as follows: Shang, J. and Wu, J. (xxxx) 'AudioKey: a usable device pairing system using audio signals on smartwatches', *Int. J. Security and Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Jiacheng Shang received his BEng in Computer Science from the Nanjing University in 2015. He is currently a PhD candidate in the Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania, USA. His current research focuses on the security and privacy issues in mobile cyber-physical systems.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell Professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications.

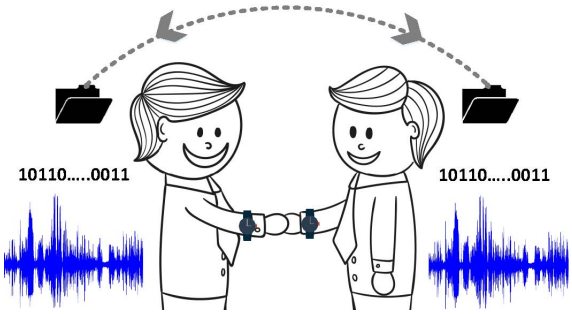
1 Introduction

During the past decade, there has been an exceptional development and fast deployment of wearable devices. Based on the data provided by Statista, sales of smartwatches are forecast to reach 141 million units in 2018 (Smartwatch Sales Worldwide, 2019). Thanks to a greater range of features and various sensors, smartwatches are expected to replace smartphones in some scenarios with better user experience and performance. For example, people who trust each other can exchange or share sensitive information (e.g., e-mail address and phone number) between smartwatches directly without taking out the smartphone. Such communication between participants should be protected against access from external devices and attackers. Existing solutions secure the communication by either using a four-digit PIN to encrypt the information or scanning a quick response code (QR code) manually. However, the PIN-based approaches suffer from brute-force attack due to limited length, and the QR code-based approaches can be attacked using a camera.

In the past few years, researchers have proposed many novel key establishment and device pairing systems to secure the communication between mobile devices and

serve as alternatives to traditional PIN code-based and cryptographic-based approaches. In these systems, multiple devices share a similar observation of a random signal. The random signal is then used to generate a secret key. The random signal can be the wireless channel information (Azimi-Sadjadi et al., 2007; Jana et al., 2009; Wang et al., 2011; Liu et al., 2012), human activity (Mayrhofer and Gellersen, 2007; Yüzugüzel et al., 2015; Ahmed et al., 2015; Xu et al., 2016; Shen et al., 2018; Xu et al., 2017), ambient noise (Mathur et al., 2011), magnetic signal (Jin et al., 2016), or electromyography (Yang et al., 2016). However, due to the characteristics of the source signal, existing systems can still be attacked in some cases. For example, wireless channel information-based systems can be attacked by blocking the line-of-sight (LOS) radio propagation between devices, which leads to various channel information observations on different mobile devices. Activity-based approaches can pair two devices by leveraging the same device movement, but the randomness of its keys is low and can be predicted. Ambient environment-based systems cannot work well if the attackers try to control the ambient signal by making predefined noises or vibrations. The electromyography (ECG)-based approach needs special hardware support for both devices. Recently, researchers proposed to pair devices by generating keys from spectrums of ambient acoustic signals (Schürmann and Sigg, 2013). However, since they use the spectrum of a wide frequency band as the random source, the rich harmonic content significantly reduce the randomness of generated keys. Moreover, most existing systems need extra infrastructure to synchronise devices and trigger the key generation process. Once the central infrastructure is attacked, these systems cannot work as expected since mobile devices do not share the same knowledge of the random signals.

Figure 1 An example application of using AudioKey for device pairing (see online version for colours)



Notes: Two normal users start generating keys by shaking hands. The smartwatches extract keys from the acoustic environment and use them to secure data exchange.

These limitations motivate us to design a usable and secure device pairing system on smartwatches at the software level. Our system enables the secure pairing on two smartwatches by fusing the gesture and the acoustic information. Specifically, our system uses the handshake activity to trigger the key generation on two smartwatches at the same time and generates a unique secret key from audio signals. There are three key insights that support our system. The first insight is that two smartwatches of two normal users share the same movement state during some states of the handshake, which enables us to trigger the key generation process on two devices at the same time without extra synchronisation infrastructure. The second insight is that the acoustic signals received by two devices have certain randomness in real scenarios. Such random audio signals can be used as the source to generate secret keys on two smartwatches that are close to each other during the handshake. In addition, most smartwatches are equipped with microphones and can collect raw audio signals in one or two channels. Therefore, such a key generation model can be supported at the software level and installed on most smartwatches. Inspired by these facts, we propose AudioKey, a system that can pair two smartwatches using audio signals during the handshake. During the handshake of two users, these two smartwatches independently generate secret keys based on the collected audio signal. The generated key can be used to authenticate each other or encrypt the sensitive data exchanged on the public channel. To realise such a system, several challenges must be addressed. First, it is not clear whether the randomness of the audio signal during the handshake is random enough to generate robust secret keys. We answer this question by checking the randomness of generated secret keys in Subsection 5.2. The second challenge is to start a key generation model on two smartwatches at the same time without extra synchronisation. To address this challenge, we design a handshake detection model in Subsection 3.3 and use it to trigger the key generation model, so that two smartwatches can start the key generation at the same time without extra synchronisation. Another challenge is to design efficient key generation and error correction models. A secret key should be generated with sufficient bit rate (BR) and have enough randomness while ensuring a high matching rate (MR) between normal users and low MR between normal users and the attacker. To address this issue, we propose a key generation scheme that leverages features in both frequency domain and time domain of the audio signals.

Our contributions in this work lie in the following aspects:

- Our system is software-based and can be easily integrated into all smartwatches.
- We propose a usable and secure device pairing system on smartwatches by fusing the gesture and audio signals. Compared with gesture-based approaches, our system greatly improves the randomness of generated secret keys. Compared with audio-based approaches, we provide a usable way to synchronise two smartwatches and a new key generation scheme that improves the randomness of keys by downsampling the harmonic content in the audio spectrum.

- Evaluation results show that our system can achieve a bit generation rate of at least 13.4 bits/s with a key agreement rate (KAR) of 96.7% for a 128-bit secret key, and our system is secure enough against strong attackers who can show up near normal users, have the same smartwatch and key generation software, start the key generation at the same time as normal users, and monitor the error correction information.

2 Problem formulation

In this section, we will introduce three threat models we consider in this work. Also, we will discuss the key insights, user case and the challenges we faced.

2.1 Feasibility study

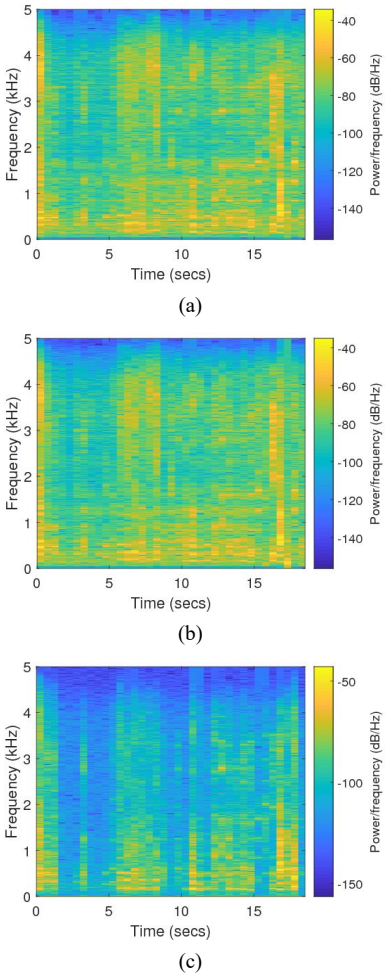
The propagation of acoustic signals in indoor environment is dynamic and complicated. Various factors (e.g., walking human beings and displacement of objects) can influence the propagation of the signal that broadcasts from each speaker (human being or loudspeaker). In this paper, we use Geometrical room acoustics theory (Siltanen et al., 2007) to model indoor acoustic propagation. For each receiver (microphone), the received acoustic signal $x(t)$ can be expressed as

$$x(t) = \sum_{i=1}^{N_s} \left(\sum_{l=1}^{N_l} H_l(s_i(t), \alpha_l, \tau_l) + \sum_{l=1}^{N_l} H_l(s_i(t), \alpha_l, \tau_l) + \sum_{l=1}^{N_l} H_l(s_i(t), \alpha_l, \tau_l) \right) \quad (1)$$

where $s_i(t)$ is the signals that is broadcasts from the i^{th} speaker, the subscripts l, r, and d represent LOS, reflection and diffusion paths, respectively, $H(\cdots)$ is the channel response of each path with the path gain α and path delay τ. We can see that the received acoustic signal $x(t)$ is influenced by many factors. In order to get the same acoustic signal, all parameters (e.g., the number of speakers, the number of paths, and so on) should remain the same. Even the attacker can control or predict the source signal $s(t)$, received signal $x(t)$ still cannot be completely estimated due to the dynamics of indoor environment (e.g., movement and displacement), which means that the keys generated from acoustic signals can be random.

To further validate the insights we find from the indoor acoustic signal model, we collect data from two normal users who are shaking hands and an attacker who is 1.2 metres away from the normal users and collect the acoustic signals at the same time as normal users. Figures 2(a) and 2(b) illustrate the spectra of two normal users. We can see that two spectra are almost the same although the energy distributions are different in some small frequency bands. This indicates that two normal users who are in close proximity can receive almost the same audio signal, which enables us to generate the same secret key for them based on their shared knowledge. Figure 2(c) shows the spectrum of the attacker. We find that the spectrum of the attacker is quite different from that of any normal user. Although the spectrum of the attack has partial information on normal users' spectra, the attacker cannot recover normal users' spectra without extra knowledge. Moreover, the interactions between two normal users also introduce unpredictable variances (e.g., normal user's), which makes it harder for the attacker to generate the same spectra. These key insights inspire us to use the spectrum to generate secrets for a pair of normal users while ensuring that the attacker will not generate the same secret key.

Figure 2 Spectra of audio signals collected on two normal users and the attacker, (a) Alice (b) Bob (c) the attacker (see online version for colours)



2.2 Threat model

Our system consists of two normal users, Alice and Bob, and one attacker. Each of the normal users carries one smartwatch that our system is implemented on. Both of them trust each other and want to pair their devices by shaking hands. A public channel is available, and everyone can exchange data through it. Alice and Bob want to

generate the same secret key using our system to protect their communication. Each of the two smartwatches works independently and has no prior knowledge of the other's existence. We assume the attacker is able to:

- know every detail of our handshake detection, key generation, and error correction algorithms
- stand in close proximity (1.2 metres) and can observe the handshake activity performed by normal users during the whole process
- eavesdrop on and decode all the packets sent via a public communication link, e.g., Wi-Fi, Bluetooth, or NFC
- have the same audio recorder that normal users have.

In this paper, we consider three threat models. In the first threat model, the attacker has no capacity except randomly guessing. In the second threat model, the attacker can be around normal users in close proximity and record the acoustic signals using the same microphone that normal users use. However, the attacker does not know the exact start time of key generation. In the third threat model, the attacker has all capacities in the second threat model and knows the exact start time of key generation (e.g., with the help of extra cameras). In the three threat models, we assume the attacker cannot perform the jamming attack by playing extremely loud music since the victims can easily notice the abnormal situations.

2.3 Use case

In the use case of our system, two normal users trust each other and agree to share information between two devices in a social event. The agreement is established by shaking their hands where their smartwatches are worn. If a normal user only wants to shake hands with a person without exchanging information, he or she can disable this functionality in advance or cancel the file transfer after the key generation (we assume the system will ask for confirmation before exchanging information). The key generation process is triggered on two devices at the same time by leveraging the shared movement of two hands during handshakes. The keys are automatically generated on each smartwatch by leveraging the microphone and motion sensors embedded in most smartwatches and wearable devices. The microphone captures the surrounding acoustic signals, and the motion sensors are used to collect raw gyroscope data. The distance between two normal users is about 0.6 metres, and the attacker is at least 1.2 metres away from normal users.

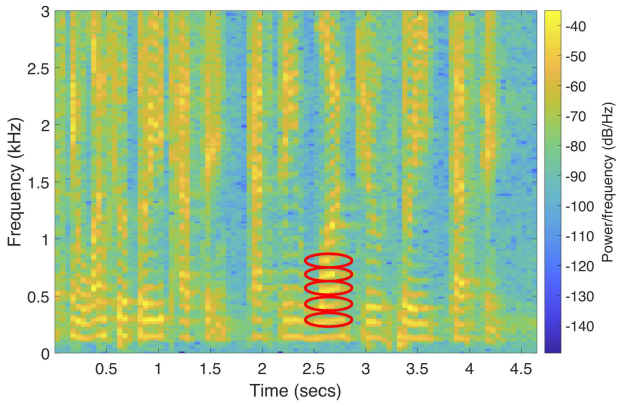
2.4 Challenges

Although we get insights in Subsection 2.1 that show we can use acoustic signals to generate secret keys for normal users, it is still challenging to design a distributed key generation system. The first challenge is how to ensure there always exists synchronisation service in practice. Without proper synchronisation, two devices cannot trigger key generation at the same time, which will influence system performance to a large degree. Previous works (Schürmann and Sigg, 2013; Yang et al., 2016) use the extra server for synchronisation among devices. However, we cannot ensure this kind of synchronisation is always available for users. As a result, a new scheme needs to be designed to trigger two smartwatches to start generating the secret key at the same time without introducing extra synchronisation. To address this issue, we study the general process of the handshake and leverage the consistency of devices' movements during the second stage of the handshake to trigger the key generation model on two devices at the same time.

The second challenge is how to select proper features of audio signals on the time-frequency domain for key generation. The key generation scheme should maximise the MR between the keys of normal users and minimise the MR between the keys of normal users and that of the attacker. A simple idea is to perform quantisation algorithms on time-domain signals, which is not suitable for our system due to the high-frequency noise in acoustic signals. In order to generate secret keys that are robust against background noise, we propose an efficient scheme that extracts the secret key from the audio spectrum on a narrow frequency band.

The third challenge is that the generated secret keys should be random enough so that the attacker in the first threat model cannot easily guess normal users' secret key. The spectrum of the audio signal received at each smartwatch contains rich information of acoustic environment (e.g., normal users' voices). However, as shown in Figure 3, the spectrum reveals its rich harmonic contents and contains duplicated information across multiple frequency bands, which would greatly reduce the randomness of the secret keys if we use the whole spectrum for key generation. To address this problem, we only leverage the spectrum on a narrow frequency band. To further avoid too many subsequences that contain consecutive and identical bits (e.g., '00000000' and '11111111'), we downsample the secret keys.

Figure 3 Duplicated information exist across multiple frequency bands of the spectrum (see online version for colours)



3 System design

In this section, we will introduce the signal pre-processing, handshake and start point detection, pairing process triggering, key generation, and error correction schemes.

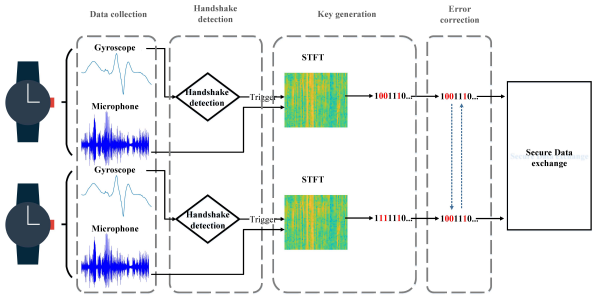
3.1 Approach overview

The key idea underlying our key generation model is to fully leverage the shared acoustic environment and the handshake activity between two normal users. As shown in Figure 4, the gyroscope in each user's smartwatch keeps measuring the angular velocity along the z-axis, and the microphone records the acoustic signals. Whenever a handshake activity is detected, each smartwatch triggers the key generation process. For each user, a secret key is generated based on the knowledge on both frequency domain and time domain of the audio signals. Then, we utilise error correcting codes to further improve the MR of generated secret keys between normal users. After error correction, two normal users generate the same secret key, and this key can be used to pair two smartwatches or secure the sensitive data transmission between two normal users.

3.2 Pre-processing

The received gyroscope signal contains high-frequency noise. Such noise needs to be removed before handshake detection in order to improve the detection performance. In our system, we use a moving average window with the window size of 20 samples to smooth received sensor data. By applying the low-pass filter, important features are maintained, while high-frequency noise is removed. In the following subsections, we will show how to use filtered gyroscope signal to detect handshake and trigger the key generation on two devices.

Figure 4 System pipeline (see online version for colours)

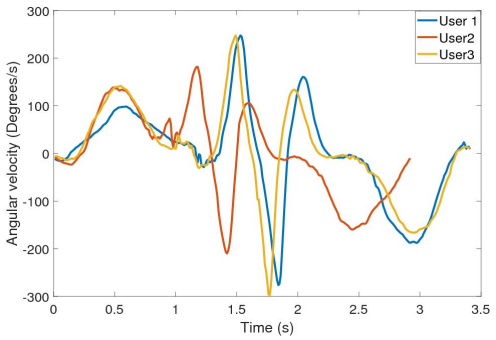


3.3 Handshaking detection

In our system, we use the handshake to trigger the key generation process on two smartwatches. Generally, the handshake activity can be divided into three stages: raising the arm, a brief up-and-down movement of the grasped hands, and putting the arm down. These activities introduce various influences on motion sensors (accelerator and gyroscope). Although both acceleration data and gyroscope data can be potentially used to detect the handshake, we only use gyroscope data along the z-axis to detect handshake in our system. The reason is that the acceleration data is more sensitive to minor arm movement during the handshake compared with gyroscope data. Even if we accurately detect the handshake, we cannot estimate the accurate starting time of any stages and trigger the key generation on two smartwatches at the same time, which further influences the MR between to normal users. Moreover, users can enable our system only when they need to securely exchange or send information, so the energy cost will not be too high even if we only use the gyroscope sensor.

We can observe the corresponding effect on gyroscope data along the z-axis in Figure 5. The arm raising activity will cause the first significant positive pulse on the gyroscope waveform. The next several positive and negative pulses are caused by the brief up-and-down movement, and the last significant negative pulse indicates that the user puts the arm down. We also find that different people share this handshake pattern in our experiments. To detect handshake, a simple idea is to use a moving time window and dynamic time warping (DTW) algorithm to match the filtered gyroscope waveform with the existing pattern. Once the DTW distance is less than a threshold, a handshake activity is claimed to be detected. However, this scheme cannot ensure good detection accuracy since the amplitudes and speeds of handshake may be quite different for different people. Also, we do not know how many times the user will repeat the up-and-down movement, which means we cannot have a common pattern of the handshake. As shown in Figure 5, the first user shakes hands more slightly than the other two users. If we match the pattern of the first user with that of the second user, the DTW distance could be very large.

Figure 5 Handshake patterns of three users (see online version for colours)



To address this problem, we need to extract features that are robust to different handshake behaviours. We notice that the whole handshake involves at least three positive peaks and two negative peaks on the gyroscope signal along the z-axis. Instead of detecting the handshake based on the shape, our handshake detection model focuses on counting the significant positive and negative peaks. We

set the window size to 3.5 seconds in order to capture the whole handshake activity. Within each time window, we implement the peak finding algorithm to find all positive peaks and negative peaks. After that, we count the number of significant positive peaks whose amplitude is higher than 80 degrees/s and the number of significant negative peaks whose amplitude is lower than –150 degrees/s. A handshake is claimed to be detected if two conditions are satisfied:

1 the number of significant positive peaks is higher than 3 and the number of significant negative peaks is higher than 2
2 the first negative significant peak appears after the second positive peak.

3.4 Triggering the pairing process

Since our system is distributed without extra synchronisation, two wearable devices need to sample the audio signal at the same time in order to establish the same secret key. Although we can accurately detect the handshake, it is hard to know the exact start time of the handshake of each user. Also, two users may not raise their arms at the same time, which means we cannot use the start time of the handshake to trigger the key generation process. To solve this problem, we leverage the consistency of the two devices' movement during the brief up-and-down movement. Since two hands are held tightly, they should move up and down almost at the same time. In our system, we trigger two devices at the same time by robustly exploring the first negative peak in the second stage, which is the start of the second stage. After handshake detection, we locate the first negative peak whose amplitude is lower than –150 degrees/s. The timestamp of the located peak is the start time of the secret key generation.

3.5 Key generation

After obtaining the audio signal, we need to improve the randomness of the audio signal and find efficient key generation schemes to extract secret keys with a high BR. Some existing works adopt quantisation algorithms for key generation. Quantisation schemes parse the sample values into binary bits based on different rules. Some systems (Liu et al., 2012; Zan et al., 2012) adopt quantisation schemes that use one or two thresholds to quantise samples to 0 or 1. However, these schemes limit the diversity of generated keys by generating a series of 1 or 0 sequences. Other systems (Yang et al., 2016) use advanced schemes that divide the whole waveform into several sub-sequences. In each sub-sequence, waveforms are converted to binary code based on their trend (rising, dropping, or steady). Though the key diversity is improved compared with traditional approaches, only three directions are considered in these schemes, which leads to the diversity of key to not be as expected. Besides quantisation, audio fingerprinting is an approach to deriving a characteristic pattern from an audio sequence (Cano et al., 2012). Some works leverage music-specific properties (e.g., rhythm information and pitch). However, such features might be lost in ambient audio, which makes these methods not applicable to our system.

In our system, we extract secret keys from both the frequency domain and the time domain of the audio signal. As we discuss in Subsection 2.4, the whole spectrum of the audio signal contains duplicated information, so we focus on one narrow frequency band that only contains one harmonic for key generation. Basically, the audio signal is first divided into a set of non-overlapping time frames. Within each time frame, we apply a short-time Fourier transformation (STFT) and split the spectrum into a set of non-overlapping frequency bands. Then, we calculate the sum of energy in each frequency band and store the sum to an energy matrix, E that has energy per time frame per frequency band:

$$\begin{aligned} &E[i,j] = \sum bandfilter_{(w_s,w_e)}(S_i) \\ &w_s = w \times j \quad w_e = w \times (j+1) \\ &i = 1,2,\ldots,N \quad j = 1,2,\ldots,M \end{aligned} \tag{2}$$

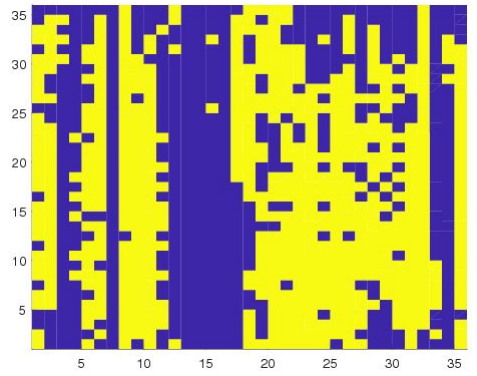
where S_i is the spectrum of the narrow frequency band we choose of the i^{th} time frame, w_s is the starting frequency of the j^{th} frequency band, w_e is the ending frequency of the j^{th} frequency band, $bandfilter_{(w_s,w_e)}(S_i)$ is a bandpass filter that only reserves the spectrum S_i from w_s to w_e, M is the number of frequency bands of each time frame, and N is the number of time frames. Also, we let $M \times N = C$, where C is constant. With a larger time frame, more samples will be included in STFT, and the granularity of STFT is better. As a result, the number of frequency bands should be larger.

After getting the energy matrix E, the most naive way to extract a secret key is to define a threshold. Each entry of E is parsed to '0' if its energy is less than the threshold. Otherwise, it is parsed to '1'. This scheme is easy to implement, but it limits the randomness of generated secret keys. As shown in Figure 6(a), the entries with high energy are usually clustered together, which produces several long sequences of continuous '1' or '0'. Instead of using a single threshold, we extract a binary representation of audio from changes in the energy of both successive frequency bands and time frames. A key x is generated based on the following equation:

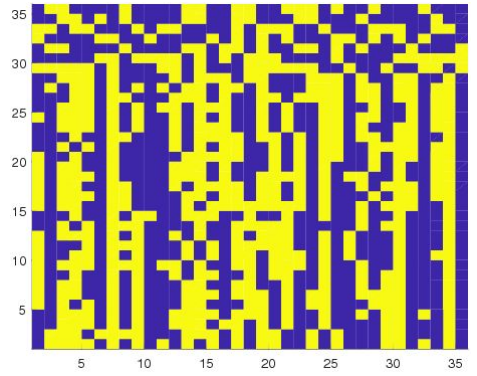
$$x[u,v] = \begin{cases} 1\,(E[u+1,v] > E[u,v]) \wedge (v > \lfloor (v/2) \rfloor) \\ 0\,(E[u+1,v] \leq E[u,v]) \wedge (v > \lfloor (v/2) \rfloor) \\ 1\,(E[u,v+1] > E[u,v]) \wedge (v \leq \lfloor (v/2) \rfloor) \\ 0\,(E[u,v+1] \leq E[u,v]) \wedge (v \leq \lfloor (v/2) \rfloor) \end{cases} \tag{3}$$

where u = 1, 2, ..., $N-1$ and v = 1, 2, ..., $M-1$. Each entry of the right half of x represents the change in the energy of successive frequency bands. If $x[u, v] = 1$, it means current frequency band has more energy than successive frequency band. Similarly, each entry of the left half of x represents the change in the energy of successive time frames within the same time frame. If $x[u, v] = 1$, it means current time frame has more energy than successive time frame within the same frequency band.

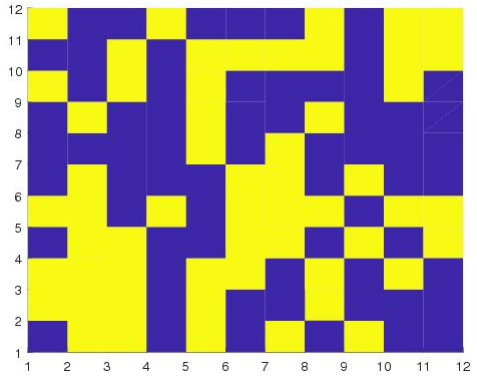
Figure 6 Key generation results of three different schemes, (a) based on single threshold (b) based on our scheme (c) after downsampling (see online version for colours)



(a)



(b)



(c)

Figures 6(a) and 6(b) show the generated keys of the single threshold-based scheme and our scheme. We can see that the single threshold-based scheme produces long '0's or '1's since the entries with high energy are usually clustered together, which greatly reduces the randomness of the secret keys. Since our scheme considers the difference between two neighbouring entries instead of only one entry, this issue can be greatly improved. However, long '0's or '1's still exist along each column, which largely influence the randomness of generated keys. To further improve the randomness of generated keys, we downsample the generated key by an integer factor D along columns. Figure 6(c) illustrates the key after downsampling the results in Figure 6(b). We can see that the average length of continuous '0's or '1's is further reduced, which indicates that the randomness of generated keys is better.

3.6 Error correction

After key generation, the two keys on two smartwatches are expected to be identical for a successful pairing. However, due to the potential bias in triggering key generation and random noise in audio signals, mismatched bits may exist in generated secret keys. Suppose two smartwatches A and B generate two keys k_A and k_B, respectively. To correct the mismatched bits between two keys, we utilise the Golay code G = (24, 12) (Golay, 1949) to correct the mismatched bits between two smartwatches. Since the G = (24, 12) cannot directly deal with a long key (e.g., 128 bits), we first equally segment each of k_A and k_B into a sequence of non-overlapped subkeys, and each subkey has the same length of λ bits. The λ is a positive integer that is at least 4 and no more than 12. To meet the requirement of G = (24, 12), we extend each subkey with λ bits to 12 bits by adding 0. We correct the mismatched bits in each pair of subkeys based on the following scheme.

Assume a and b are a pair of subkeys of k_A and k_B. We first generate a 24-bit code by adding 12-bit redundant information δ after a. Instead of sending the subkey a to smartwatch B, the smartwatch A only sends the redundant information δ to reduce the information leakage. Since δ contains limited knowledge of a, the smartwatch B can correct the mismatched bits in b by applying decoding function on b and δ. The value of λ determines the ability of error correction. Since the Golay code G = (24, 12) can correct at most three error bits, a small λ can ensure a high bit MR between normal users after error correction, but the attacker can also have more knowledge about the victim's secret key. If the value of λ is large, it is harder for the attackers to guess the key, but the bit MR between normal users may decrease since more than four error bits may exist in a subkey.

4 Evaluation

In this section, we first will show the detailed implementation of our system. Testing scenarios and evaluation metrics are then introduced in the following subsections. Finally, we will discuss the system's performance in different settings to show that AudioKey is accurate and robust enough for smartwatch paring.

4.1 Experiment setup

Figure 7(a) shows a prototype of AudioKey. Each user wears a smartwatch on their wrist and shakes hands with each other. In order to help to collect data, we build an application on Samsung Gear 3 smartwatches with a UI design, as shown in Figure 7(b). Both devices of normal users and the attacker keep collecting gyroscopes and raw

audio signals with sampling frequencies of 100 Hz and 16,000 Hz, respectively. Audio signals are collected on a single channel.

4.2 Testing scenarios and data collection

To evaluate the effectiveness and robustness of our system, we conduct various experiments including nine volunteers in four different scenarios, including a quiet office room, supermarket, roadside, and noisy places. Nine volunteers are university students who age from 22 to 29. In each scenario, we evaluate our system performance between two normal users and the resistance of our system to three attacks we consider. Across all experiments, the attacker is equipped with the same smartwatch that the normal users wear. In order to evaluate the influence of the attack distance, we adopt the Edward T. Hall's proxemics theory to emulate how closely an attacker can show up around the normal users. As shown in Figure 7(c), for a successful attack, the attacker cannot appear within 1.2 metres away from the normal users since it is the personal space and the attacker can be easily noticed. The attacker in our experiments can appear anywhere as long as he or she is at least 1.2 metres away from any normal user.

4.3 Evaluation metrics

In all the experiments, three metrics are used to measure the performance of AudioKey.

- *BR:* In our experiments, the BR is defined as the average number of bits our scheme can generate in one second before error correction.

  $$BR = ((M \times N) \mod D)/T \tag{4}$$

  where T is the time duration of audio signal, M is the number of frequency bands of each time frame, N is the number of time frames, and D is the integer factor for downsampling.

- *MR:* The MR is defined as the ratio of matched bits divided by the total length of the generated secret key after error correction. It reflects the level of consistency between two secret keys.

  $$MR = N_{match}/L \tag{5}$$

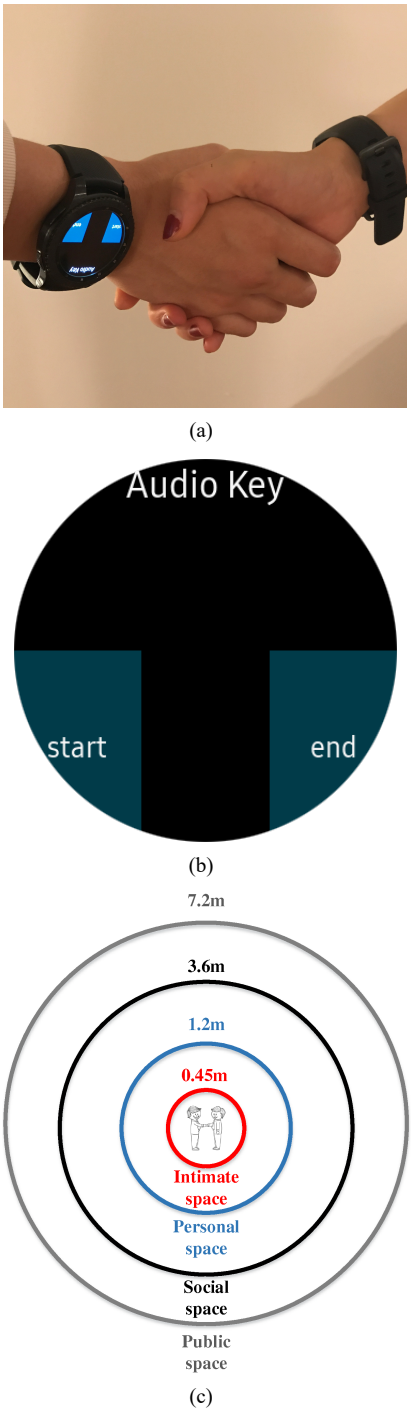
  where N_{match} is the number of matched bits and L is the length of generated secret key.

- *KAR:* The KAR is defined as the rate at which two keys generated on two smartwatches are identical.

5 System performance

In this section, we evaluate the performance of our handshake detection model and key generation model. Also, we will examine the randomness of generated secret keys in our system.

Figure 7 Experiment setup, (a) prototype (b) UI (c) illustration of attackers' locations (see online version for colours)



(a)

(b)

(c)

5.1 Handshake detection

Since our key generation model is triggered by the handshake detection model, the accuracy of the detection is an important performance indicator. In our experiments, we ask the nine volunteers to repeat handshake 20 times and evaluate the handshake detection performance. Our results show that our handshake detection model can achieve average detection accuracies of 98.9%, which means in

most cases our handshake detection model can accurately trigger the key generation model.

5.2 *Randomness of generated keys*

To ensure the randomness of generated keys, we leverage the standard randomness test suit from NIST to examine the randomness level of our secret keys after downsampling. NIST randomness test suit involves a series of randomness tests with a null hypothesis that the input key is random and computes the p-value. If the p-value is less than a 1% then the hypothesis is rejected, and the keys we generate is non-random.

Table 1 Randomness test

Tests	*p-value*	*Tests*	*p-value*
Frequency	0.659937	Cum. sum (forward)	0.027023
Block frequency	0.665073	Cum. sum (backward)	0.787595
Approximate entropy	0.030886	FFT	0.926884
Runs	0.114544	Serial (p-value1)	0.890679
Longest run	0.583223	Serial (p-value2)	0.961107

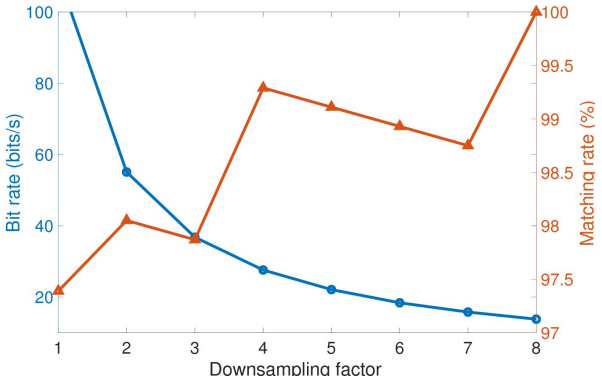
We collect the audio signals in different scenarios including office room, road, and so on, and use them to generate secret keys with a downsampling factor of 8. Table 1 shows the p-values across all tests. We can see that the keys generated in different scenarios can pass all tests with a high p-value, which indicates enough randomness of generated secret keys. Our experimental results also show that when the downsampling rate is 4 or less, generated secret keys cannot pass the NIST randomness test.

5.3 *Bit rate*

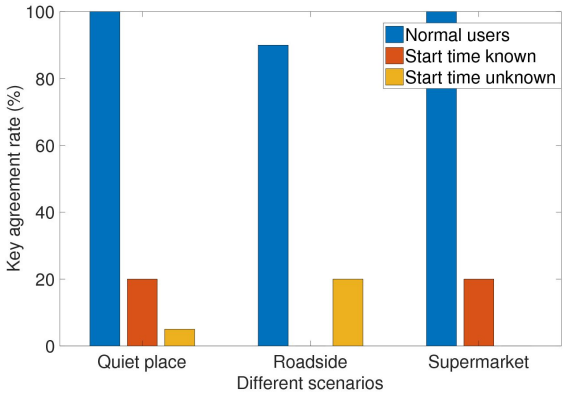
One important performance indicator of key generation scheme is how fast it can generate secret bits. In our system, the BR before error correction largely depends on the downsampling factor D. With a small D, we can get a higher BR, but more noise bits will be included and the randomness decreases. With a large D, we can expect a key with better randomness, but the BR will be low. In order to understand how downsampling factor influences the BR, the MR, and the randomness of secret keys, we conduct experiments to find the optimal downsampling factor D. Figure 8(a) illustrates the BR under different downsampling factors. When we do not perform downsampling, our scheme can achieve an extremely high BR of 110.2 bits/s. When the downsampling factor is 8, the BR is 13.8 bits/s, which we argue is the minimal BR we can accept in our use case.

We also evaluate whether downsampling can influence the MR between two normal users after error correction. It is clear in Figure 8(a) that the downsampling operation will not reduce the MR between two normal users. In contrast, it can improve the MR. When we do not perform downsampling on the secret keys, the MR after error correction is 97.39%, while it is at least 97.87% after downsampling. It is because the error bits are not uniformly distributed on original secret keys and are usually clustered together. By downsampling, we can reduce the average number of error bits in every 12-bit word.

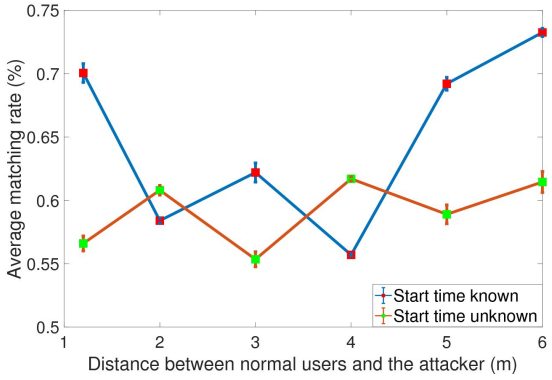
Figure 8 System performance, (a) the influence of downsampling factor on BR before error correction and MR (b) system performance in different scenarios (c) bit MR between normal users and the attacker when the attacker is at different distances from normal users (see online version for colours)



(a)



(b)



(c)

As shown in Table 2, we also compare the BR of AudioKey with existing key generation systems. We can see that the BR of our system is only lower than those of Magpairing (Jin et al., 2016) and Shake-n-Shack (Shen et al., 2018). However. Shake-n-Shake (Shen et al., 2018) only discuss the resistance against mimicry attacks rather than

the randomness of generated keys. Since Shake-n-Shake generates keys using the behaviour of handshake that only contain minor changes across different people, generated keys may not be random so that attackers can have much knowledge on the victim's keys. Magpairing (Jin et al., 2016) proves the security of generated keys by checking the approximate entropy of keys. However, it is not clear if their keys can pass more randomness test such as block frequency test and FFT test. Compared with existing works, our system can generated keys that can pass the NIST randomness test with a better BR of 13.4 bits/s. With less randomness requirement, our system can also provide a high BR of about 100 bits/s.

Table 2 BRs of different key generation systems

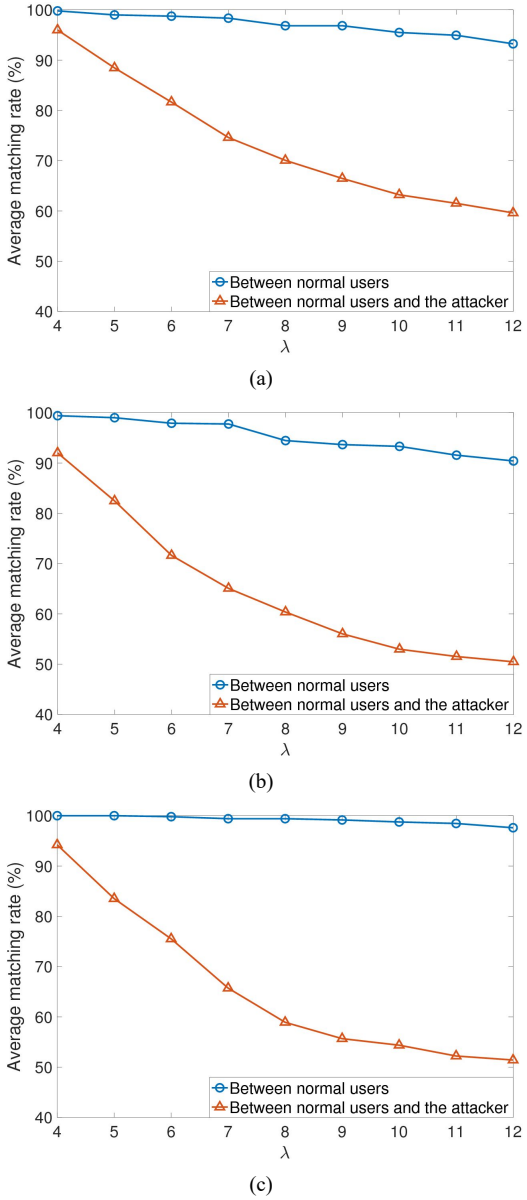
Systems	*AudioKey*	*ProxiMate (Mathur et al., 2011)*	*ShakeMe (Yüzugüzel et al., 2015)*
BR (bits/s)	13.4	1.8	8
Systems	*EMG-KEY (Yang et al., 2016)*	*Magpairing (Jin et al., 2016)*	*Shake-n-Shack (Shen et al., 2018)*
BR (bits/s)	5.51	28.4	90

5.4 *Influences of different scenarios*

To evaluate the system performance in different scenarios, we examine the KAR after error correction in three scenarios, including quiet office room, roadside, and supermarket. In each scenario, we trigger the key generation for 20 times to generate 20 128-bit secret keys on each smartwatch. In these experiments, the attacker is about 1.2 metres away from two normal users, and the λ is set to 4. Here, we consider the attackers in both the second and the third threat models. The attacker in the third threat model starts the key generation at the same time as normal users, and the attacker in the second threat model starts 0.75 seconds after the normal users generate keys. Figure 8(b) shows the KAR in three scenarios and two threat models. We can see that the keys of normal users have much higher KAR (at least 90%) than that between the normal users and the attacker (no more than 20%) in any threat model, which indicates that the attacker only has limited knowledge of the keys of normal users in all scenarios even if the attacker is close enough to the normal users and can monitor the error correction information. On average, when $\lambda = 4$, our system can achieve a mean key agreement probability of 96.7% for normal users and a low key agreement of 10.8% for attackers in terms of a 128-bit secret key.

Also, we notice that the KAR between normal users and the attacker in quiet places is higher than that in noisy places (roadside and supermarket). In quiet places, generated secret keys largely depend on the voice of two normal users and much less background noise will be included, so that the attacker can acquire more knowledge about the secret key. Even if the attacker can show up closely around normal users and hear the content of what normal users say, our system can still ensure a mean low KAR of 12.5% between normal users and the attacker.

Figure 9 Key MR after 1-round error correction with different λs in three scenarios, (a) quiet office room (b) roadside (c) supermarket (see online version for colours)



5.5 *Influences of distances between normal users and the attacker*

In real scenarios, the attacker will appear at any location without being noticed by normal users. The attacker can be very close to normal users (about 1.2 metres) or very far from them. To evaluate the attacker's knowledge about normal users' key after error correction when the attacker is at different distances from normal users. Figure 8(c) shows the average MR with error bars between normal

users and the attacker after correcting 3-bit errors in every 8-bit key ($\lambda = 8$) when the attacker is at six different distances away from normal users. As the reference, the MR between normal users is above 98.5% in the same scenario. We can see that all attackers have very limited knowledge about normal users' secret keys when they are at least 1.2 metres away. The attacker who starts the key generation at the same time has the highest MR of 73.25% when the attacker is 6 metres away, and the attacker who starts the key generation 0.75 seconds later than normal users has the highest MR of only 61.45%. We also notice that the MRs of attackers do not drop with the increase of the distances. This is because all these attackers generate keys based their own spectrum, and the MR depends on how similar their spectrum is to that of normal users, which is random. That is why the attacker who is 6 metres away can have a higher MR than the attacker who is 1.2 metres away from normal users. Also, note that the attackers in this experiment almost have the highest ability. Although the attacker can have a bit MR of about 75%, the number of unknown bits are about 32 bits. As a reference, the number of unknown bits of PIN-based Bluetooth encryption (used by current smartwatches) is 20 bits. Moreover, the locations of mismatched bits are unknown to the attackers. These facts imply the keys generated by our system is much more secure than existing solutions.

5.6 *Influence of different λ*

In Subsection 3.6, we improve the MR after error correction between two normal users by using shorter sub-keys for error correction. However, it is not clear what is the proper value of λ in different scenarios. If the λ is too small, the attacker can acquire too much knowledge about normal users' keys and can recover it easily. If the λ is too large, we cannot ensure good MR between normal users, although the attacker gets very limited knowledge about the keys. To understand what is the proper value of λ in different scenarios, we adjust the value of λ from 4 to 12 and see its influence on the MR between two normal users and between normal users and the attacker after 1-round error correction, and evaluation results are shown in Figure 9. We can see that the MRs drop with the decrease of λ in all scenarios. This is in line with expectations since a smaller λ can provide better error correction capability. Moreover, we notice that the difference between two MRs rises with the increase in λ. Although more error bits cannot be corrected in 1-round error correction with larger λ between normal users, the attacker gets much less knowledge of normal users' secret keys. If the normal users pay more attention to the security, we can set λ to a large integer and perform error correction for multiple rounds between normal users to ensure they generate the same keys. If the users value time efficiency more and accept keys with lower security, we can set λ to a small integer. In all cases, we can ensure the security of generated secret keys. The users can choose different security levels and time efficiency levels by adjusting the values of λ.

6 Related work

- *Wireless channel information:* Recently, various system are proposed to generate secret keys by using wireless channel information (Azimi-Sadjadi et al., 2007; Jana et al., 2009; Wang et al., 2011; Liu et al., 2012). The work in Azimi-Sadjadi et al. (2007) is the first one to generate a secret key from wireless signal strength. Liu et al. (2012) propose star-based and chain-based approaches to generate a secret key among multiple wireless devices to ensure secure group communication. However, these approaches can still be attacked by blocking the LOS radio propagation between devices, which leads to different channel information observations on different mobile devices. Moreover, accurate channel information measurements rely on special devices, which are not supported on current smartwatches.
- *Human activity:* Mayrhofer and Gellersen (2007) propose ShaVe and ShaCK that can generate a secret key from an accelerometer waveform through shaking two devices together. A similar system called ShakeMe (Yüzugüzel et al., 2015) is designed for key generation from a shared motion. Another system called Checksum Gestures (Ahmed et al., 2015) uses a single-continuous gesture to generate an authentication code in order to replace the traditional PIN input for wearables. Gait is exploited in Xu et al. (2016) to generate a shared secret key for all on-body devices. Those systems are designed for mobile devices that are held together, which is not convenient for users during interactions. Also, these approaches can be attacked if target's activities are captured by a hidden camera.
- *Ambient environment:* ProxiMate presented in Mathur et al. (2011) allows wireless devices in proximity to securely pair with one another autonomously by generating a common cryptographic key directly from their shared time-varying wireless environments. However, ProxiMate asks mobile devices to emit a trigger signal for synchronisation on a private channel before key generation. A system using ambient sound is proposed in Schürmann and Sigg (2013) by putting several mobile devices together with synchronisation. These ambient noise-based approaches can still be threatened by making predefined noises.
- *Magnatic signal:* The system designed in Jin et al. (2016) attaches tiny magnets to a mobile device and exploits the correlated magnetometer readings for key generation. The new implementation in this work is not suitable for small mobile devices.
- *Electromyography:* EMG-KEY (Yang et al., 2016) leverages electrical activities caused by human muscle contraction to generate a secret key. This approach requires that both the transmitter and the receiver are equipped with EMG sensors, which introduces more implementation costs.

- *Visual information:* The visual information-based approaches (e.g., QR code) are widely adopted by current mobile applications to establish the trust between two devices. A similar idea is proposed in McCune et al. (2005). However, these approaches still suffer from shoulder-surfing attacks.

7 Discussion

In this section, we will discuss the two factors that may influence system performance and limitations of our system.

- *Threat of jamming attack:* In practice, there exists the attacker who just wants to generate the same key of normal users by altering the acoustic environment. For example, the attacker can carry a loud speaker and play sounds with high volumes. In this case, it is very likely the attacker can acquire higher MRs. However, this attack cannot work since we use low frequency part of the spectrum for key generation (under 1,000 Hz). This kind of attack can be easily noticed by the normal users who are in the key generation process.
- *Threat of body movements during handshake detection:* During handshake, the users do not always keep steady, and some body movements may be involved, which may generate influences on motion sensors. We argue that the handshake detection scheme will not be influenced by body movements as long as the users follow the three steps of handshake. The reason is that we only use the gyroscope measurements on the z-axis. Since people usually wear their smartwatches on wrists, gyroscope readings on the z-axis are mainly influenced by raising and putting down the arm during handshake.

Next, we will discuss the limitations and future work of our system. Our system involves a limited number of participants, and all users are university students. To better understand the performance of our system, it will be necessary to involve more participants with a more diverse background. Also, the experiments are conducted within one month. Since our system uses handshake activity to trigger the key generation model, a long-term evaluation can be conducted considering that the human behaviour and habits may change.

8 Conclusions

In this paper, we develop a usable and secure device pairing system called AudioKey for smartwatches. AudioKey fuses the gesture and audio signals to pair two smartwatches. More specifically, AudioKey detects the handshake activity between two normal users and triggers the key generation process on two devices at the same time. After handshake detection, a secret key is extracted from both the frequency domain and the time domain of audio signals and used to authenticate each other or encrypt the sensitive data exchanged on the public channel. Evaluation results performed on nine volunteers under three different scenarios show that our system can generate keys with sufficient randomness and robustness against strong attackers, and achieve a high handshake detection accuracy of 98.9% and a bit generation rate of 13.4 bits/s with a mean KAR of 96.7% for a 128-bit secret key. The generated secret keys can pass the standard randomness test suit from NIST with high p-values.

Acknowledgements

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS-1651947, and CNS 1564128.

References

Ahmed, I., Ye, Y., Bhattacharya, S., Asokan, N., Jacucci, G., Nurmi, P. and Tarkoma, S. (2015) 'Checksum gestures: continuous gestures as an out-of-band channel for secure pairing', in *Proc. of UbiComp*, ACM, pp.391–401.

Azimi-Sadjadi, B., Kiayias, A., Mercado, A. and Yener, B. (2007) 'Robust key generation from signal envelopes in wireless networks', in *Proc. of CCS*, ACM, pp.401–410.

Cano, P., Batle, E., Kalker, T. and Haitsma, J. (2002) 'A review of algorithms for audio fingerprinting', in *Proc. of MMSP*, IEEE, pp.169–173.

Golay, M.J.E. (1949) 'Notes on digital coding', *Proc. IEEE*, Vol. 37, p.657.

Jana, S., Premnath, S.N., Clark, M., Kasera, S.K., Patwari, N. and Krishnamurthy, S.V. (2009) 'On the effectiveness of secret key extraction from wireless signal strength in real environments', in *Proc. of MobiCom*, ACM, pp.321–332.

Jin, R., Shi, L., Zeng, K., Pande, A. and Mohapatra, P. (2016) 'Magpairing: pairing smartphones in close proximity using magnetometers', *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 6, pp.1306–1320.

Liu, H., Yang, J., Wang, Y. and Chen, Y. (2012) 'Collaborative secret key extraction leveraging received signal strength in mobile wireless networks', in *Proc. of INFOCOM*, IEEE, pp.927–935.

Mathur, S., Miller, R., Varshavsky, A., Trappe, W. and Mandayam, N. (2011) 'Proximate: proximity-based secure pairing using ambient wireless signals', in *Proc. of MobiSys*, ACM, pp.211–224.

Mayrhofer, R. and Gellersen, H. (2007) 'Shake well before use: authentication based on accelerometer data', *Pervasive Computing*, Vol. 4480, pp.144–161.

McCune, J.M., Perrig, A. and Reiter, M.K. (2005) 'Seeing-is-believing: using camera phones for human-verifiable authentication', in *Proc. of S&P*, IEEE, pp.110–124.

Schürmann, D. and Sigg, S. (2013) 'Secure communication based on ambient audio', *TMC*, Vol. 12, No. 2, pp.358–370.

Shen, Y., Yang, F., Du, B., Xu, W., Luo, C. and Wen, H. (2018) 'Shake-n-shack: enabling secure data exchange between smart wearables via handshakes', in *Proc. of PerCom*, IEEE.

Siltanen, S., Lokki, T., Kiminki, S. and Savioja, L. (2007) 'The room acoustic rendering equation', *The Journal of the Acoustical Society of America*, Vol. 122, No. 3, pp.1624–1635.

Smartwatch Sales Worldwide (2019).

Wang, Q., Su, H., Ren, K. and Kim, K. (2011) 'Fast and scalable secret key generation exploiting channel phase randomness in wireless networks', in *Proc. of INFOCOM*, IEEE, pp.1422–1430.

Xu, W., Revadigar, G., Luo, C., Bergmann, N. and Hu, W. (2016) 'Walkie-talkie: motion-assisted automatic key generation for secure on-body device communication', in *Proc. of IPSN*, IEEE Press, p.3.

Xu, W., Javali, C., Revadigar, G., Luo, C., Bergmann, N. and Hu, W. (2017) 'Gait-key: a gait-based shared secret key generation protocol for wearable devices', *TOSN*, Vol. 13, No. 1, p.6.

Yüzugüzel, H., Niemi, J., Kiranyaz, S., Gabbouj, M. and Heinz, T. (2015) 'Shakeme: key generation from shared motion', in *Proc. of CIT/IUCC/DASC/PICOM*, IEEE, pp.2130–2133.

Yang, L., Wang, W. and Zhang, Q. (2016) 'Secret from muscle: enabling secure pairing with electromyography', in *Proc. of SenSys*, pp.28–41.

Zan, B., Gruteser, M. and Hu, F. (2012) 'Improving robustness of key extraction from wireless channels with differential techniques', in *Proc. of ICNC*, IEEE, pp.980–984.