TOWARDS SMART AND SECURE IOT WITH PERVASIVE SENSING

By

JIAN LIU

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Yingying (Jennifer) Chen

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2019

ABSTRACT OF THE DISSERTATION

Towards Smart and Secure IoT with Pervasive Sensing

By JIAN LIU

Dissertation Director:
Yingying (Jennifer) Chen

With the advancement of mobile sensing and pervasive computing, extensive research is being carried out in various application domains of Internet of Things (IoT), such as smart home, smart healthcare, connected vehicles, and their security issues. My research work explores the power of pervasive sensing technologies to benefit people's daily lives and make impacts on society advancement, especially in the emerging areas of smart healthcare, IoT security and IoT embedded system communications. In this dissertation, I mainly study the following topics: (1) how to perform vital signs monitoring during sleep towards smart healthcare; (2) how to conduct user authentication on any solid surface for IoT applications; (3) IoT security: side-channel security leakage of typing with a nearby phone; and (4) high-throughput and inaudible acoustic communication for IoT applications.

We first propose to track the vital signs of both breathing rate and heart rate during sleep by using off-the-shelf WiFi without any wearable or dedicated devices. Our system reuses existing WiFi network of IoT and exploits the fine-grained channel information to capture the minute movements caused by breathing and heart beats. Our system thus has the potential to be widely deployed and perform continuous long-term monitoring. Our extensive experiments demonstrate that our system can accurately capture

vital signs during sleep under realistic settings and achieve comparable or even better performance comparing to traditional and existing approaches, which is a strong indication of providing noninvasive, continuous fine-grained vital signs monitoring without any additional cost.

Moreover, we propose VibWrite that extends finger-input authentication beyond touch screens to any solid surface for smart access or IoT systems (e.g., access to apartments, vehicles or smart appliances). It integrates passcode, behavioral and physiological characteristics, and surface dependency together to provide a low-cost, tangible and enhanced security solution. VibWrite builds upon a touch sensing technique with vibration signals that can operate on surfaces constructed from a broad range of materials. It is significantly different from traditional password-based approaches, which only authenticate the password itself rather than the legitimate user, and the behavioral biometrics-based solutions, which usually involve specific or expensive hardware (e.g., touch screen or fingerprint reader), incurring privacy concerns and suffering from smudge attacks. VibWrite discriminates fine-grained finger inputs and supports three independent passcode secrets including PIN number, lock pattern, and simple gestures by extracting unique features to capture both behavioral and physiological characteristics such as contacting area, touching force, and etc. Our extensive experiments demonstrate that VibWrite can authenticate users with high accuracy, low false positive rate and is robust to various types of attacks.

In addition, we explore the limits of audio ranging on mobile devices in the context of a keystroke snooping scenario. we show that mobile audio hardware advances of mobile and IoT devices can be exploited to discriminate mm-level position differences and that this makes it feasible to locate the origin of keystrokes from only a single phone behind the keyboard. The technique clusters keystrokes using time-difference of arrival measurements as well as acoustic features to identify multiple strokes of the same key. It then computes the origin of these sounds precise enough to identify and label each key. By locating keystrokes this technique avoids the need for labeled training data or linguistic context. Experiments with three types of keyboards and off-the-shelf smartphones demonstrate that our system can recover 94% of keystrokes, which to

our knowledge, is the first single-device technique that enables acoustic snooping of passwords.

Finally, we design the first acoustic communication system, which achieves high-throughput and inaudibility at the same time. The highest throughput we achieve is over 17× higher than the state-of-the-art acoustic communication systems, which could facilitate various IoT applications. Particularly, we theoretically model the non-linearity of the mobile device's inbuilt microphone and use orthogonal frequency division multiplexing (OFDM) technique together with the non-linearity model to transmit data bits over multiple orthogonal channels with an ultrasound frequency carrier. Extensive evaluations under various realistic settings demonstrate that our inaudible acoustic communication system achieves throughput as high as $47.49kbps$.

Acknowledgements

First and foremost, I have to thank my parents, Huiwei Liu and Wanghua He, for their love and support throughout my life. I love them so much, and I am forever grateful for my caring, patient, and supportive parents.

I would also like to express my deepest gratitude to my academic advisor and committee chair, Dr. Yingying (Jennifer) Chen, for her continuous support of my Ph.D. study over the past six years, for her patience, motivation, encouragement, guidance, and immense knowledge. She worked with me on every step in my research work. Without her selfless mentorship and unwavering support, this dissertation would not have been possible. In addition to the technical guidance, I received invaluable suggestions and support from her helping me improve myself in almost every aspect, such as critical thinking skills, writing/communication skills, management skills and team work collaboration, etc. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to extend my thanks to the other committee members for their help and support as always: Dr. Dipankar Raychaudhuri, Dr. Sheng Wei, Dr. Yu-Dong Yao (outside committee member at Stevens Institute of Technology), and Dr. Chung-Tse Michael Wu (proposal committee member). I am also grateful to my collaborators, Dr. Marco Gruteser, Dr. Richard P. Martin, Dr. Richard Howard, Dr. Yan Wang (Binghamton University), Dr. Jie Yang (Florida State University), Dr. Hongbo Liu (Indiana University - Purdue University Indianapolis), Dr. Xiaonan Guo (Indiana University - Purdue University Indianapolis), Dr. H. Vincent Poor (Princeton University), Dr. Nitesh Saxena (University of Alabama at Birmingham), Dr. Mooi Choo Chuah (Lehigh University), Dr. Jiadi Yu (Shanghai Jiao Tong University), whom I have published papers with. They provided me great guidance and help on my research. Without

their passionate participation and great guidance, these research studies could not have been successfully conducted.

I would also like to thank many of my friends and lab mates, who have worked with me and made my experience during the past six years exciting and fun, including Dr. Yanzhi Ren, Dr. Xiuyuan Zheng, Chen Wang, Song Shi, Dr. Chuyu Wang, Li Lu, Yang Bai, Yi Xie, Luyang Liu, Hongyu Li, Dr. Cagdas Karatas, Dr. Gorkem Kar, etc.

Dedication

To my parents for their tremendous love, continuous support and encouragement.

Table of Contents

Chapter 1

Introduction

1.1 Fine-grained Sleep Monitoring Leveraging Off-the-shelf WiFi

Vital signs, such as breathing rate and heart rate, indicate the state of a person's essential body functions. They are the essential components to assess the general physical health of a person and identify various disease problems. Correlating the vital signs with our sleep quality can further enable sleep apnea diagnosis and treatment [1], treatment for asthma [2] and sleep stage detection [3]. However, the traditional way to monitor vital signs during sleep requires a patient to perform hospital visits and wear dedicated sensors [4], which are intrusive and costly. The obtained results may be biased because of the unfamiliar sleeping environments in the hospital. Moreover, it is difficult, if not possible, to run long-term sleep monitoring in clinical settings. Thus, a solution that can provide non-invasive, low-cost and long-term vital signs monitoring without requiring hospital visits is highly desirable.

Recently, Radio Frequency (RF) based monitoring solutions [5, 6, 7, 8] have drawn considerable attention as they provide non-invasive breathing rate monitoring. For example, F. Adib et al. utilize Universal Software Radio Peripheral (USRP) and Frequency Modulated Continuous Wave (FMCW) radar to monitor a person's breathing rate by detecting the chest fluctuations caused by breathing [7, 8]. Doppler radar [5] and ultra-band radar [6] are utilized to catch a person's breathing respectively. These systems involve specialized devices with high complexity, which prevent them from large-scale and long-term deployment. Furthermore, N. Patwari et al. [9, 10] use coarse-grained channel information (i.e., received signal strength (RSS)) extracted from wireless sensor nodes to detect breathing rate. Their approach requires additional wireless network infrastructure (i.e., dedicated sensor nodes), and the coarse-grained channel

information is not able to capture the vital signs of heart rate. Another new direction is using wearable sensors (such as Fitbit [11] and Jawbone [12]) to track people's fitness at any time. But they only have the capability of performing coarse-grained sleep monitoring without capturing the breathing rate, which is critical to many sleep problem diagnosis including sleep apnea. Additionally, users are required to wear these fitness sensors even during their sleep, which could be a challenge for elder people.

To address these issues, our work aims to perform continuous long-term vital signs monitoring with low cost and without the requirement of wearing any sensor. We show that it is possible to track breathing rate and heart rate during sleep by using WiFi signals between WiFi-enabled IoT devices. This will largely increase the opportunity for wide deployment and in-home use. Indeed, our system re-uses existing WiFi network of IoT devices for tracking vital signs without dedicated/wearable sensors or additional wireless infrastructure. Furthermore, by exploiting fine-grained channel information, Channel State Information (CSI), provided by off-the-shelf WiFi-enabled IoT devices, our system captures not only the breathing rate but also heart rate. Specifically, our system utilizes the readily available channel information to detect the minute movements caused by breathing and heart beats (i.e., inhaling, exhaling, diastole and systole).

Using channel state information has significant implication on how fine-grained minute movements can be captured for vital signs monitoring. Comparing to the traditional RSS, which only provides a single measurement of the power over the whole channel bandwidth, the fine-grained CSI provides both amplitude and phase information for multiple OFDM subcarriers. For instance, the mainstream WiFi systems such as 802.11 a/g/n are based on OFDM where the relatively wideband $20MHz$ channel is partitioned into 52 subcarriers. Due to the frequency diversity of these narrowband subcarriers, the multipath effect and shadow fading at different subcarriers may result in significant difference in the observed amplitudes. This means that a small movement in physical environment may lead to the change of CSI at some subcarriers, whereas such change maybe smoothed out if we examine the signal strength over the whole channel bandwidth. Our system thus takes advantage of the fine-grained CSI provided by off-the-shelf WiFi-enabled IoT devices to capture the minute movements for vital

signs monitoring.

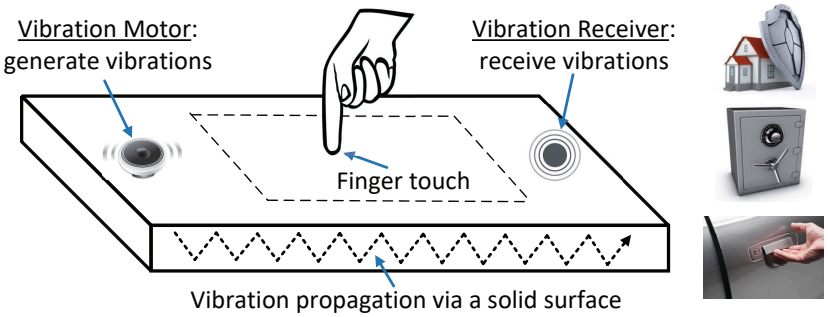
Our system uses only a single pair of WiFi-enabled IoT device and wireless AP for detecting the breathing rate, heart rate and sleeping patterns (e.g., sleeping events and postures) during sleep. The breathing rate detection algorithm first obtains time series of CSI from off-the-shelf WiFi device (e.g., desktop, laptop, tablet, and smartphone) and then analyzes the information in time domain and frequency domain. It achieves high accuracy for both single and two-person in bed scenarios. To detect heart rate, our algorithm first applies a bandpass filter to eliminate irrelevant frequency components, and then estimates the heart rate in the frequency domain by locating the frequency peak in the normal heart rate range. Additionally, we distinguish different sleep events (e.g., going to bed, turn overs during sleep) based on the CSI's variance energy and further identify people's sleep posture using a machine learning based approach. Extensive experiments are conducted in lab environment and two apartments with difference sizes. The results show that our system provides accurate breathing rate and heart rate estimation not only under typical settings but also covering challenging scenarios including long distance between the WiFi device and AP, none-line-of-sight (NLOS) situation and different sleep postures. This demonstrates that our approach can provide device-free, continuous fine-grained vital signs monitoring without any additional cost. It has the capability to support large-scale deployment and long-term vital signs monitoring in non-clinical settings.

1.2 Towards Finger-input Authentication on Ubiquitous Surfaces via Physical Vibration

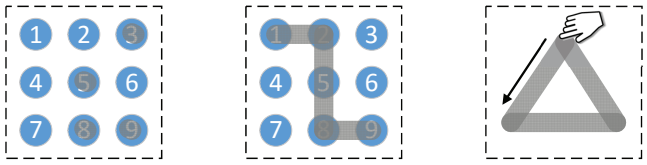
The process of authentication verifies a user's identity and is frequently deployed at almost every corner of our daily lives. In particular, the increasingly wide deployment of smart access systems of IoT, which are defined as those used for keyless controlling access to corporate facilities/apartment buildings/hotel rooms/smart homes/vehicle doors, require the authentication process to play a broader role in numerous daily activities beyond the common form authentication on touch screen devices, such as

mobile phones. The classic physical-key based access methods do not possess user authentication functionality. A market report shows that the deployment of smart security access systems is expected to grow rapidly at an annual rate of 7.49% and will reach a market value of $9.8 billion by the year of 2022 [13]. The current authentication process in smart security access systems mainly relies on traditional solutions supported by intercom, camera, card, or fingerprint based techniques. These approaches however involve expensive equipment, complex hardware installation, and diverse maintenance needs. The trend of employing low-cost low-power tangible user interfaces (TUI) on IoT devices to support user authentication in various facility entrances, apartment doors and vehicles has gained industry attentions recently. For example, token devices (e.g., smart ring, glove or pen) could be utilized for associating identities of their touch interactions [14, 15], and an ultra-thin sensing pad can be deployed in automobiles to perform driver authentication [16]. Moreover, isometric buttons appearing on new models of microwave ovens and stove tops and rotary inputs (e.g., used by iPod) can replace the regular physical buttons to provide better functionality and flexibility [17]. These new approaches appear promising of conducting user authentication and operating appliances/devices in smart systems leveraging capacitive sensing. However, these techniques require that the touched surface possesses electric conductivity and an electric field that produces/stores electrical energy, which largely limits the wide deployment of such solutions.

Along this direction, we start a new search in developing a low-cost general user authentication approach, which has the capability to work with any solid surface for smart access and IoT systems. The convenience of executing user authentication via touching any surface is enticing. For instance, a driver can just place his palm against the driver side window to access and start the vehicle. This has already been visualized in the popular movie "Mission Impossible 5", in which the featured BMW muscle car can be unlocked instantly when the lead actor pressed his palm against the side window. In another instance, a user can place his hand on the door panel of his apartment to perform authentication and unlock the entrance door without card access. Furthermore, electronic appliances in smart homes have a growing need to provide customized services



(a) Finger touching on a vibrating surface



(b) Three types of secrets

Figure 1.1: Illustration of a finger touching on a solid surface under physical vibration, and three independent types of secrets for pervasive user authentication.

for advanced safety needs such as prohibiting children and elderly people to operate risky appliances (e.g., oven and dryer), adjusting room temperature/lighting conditions and recommending TV content. A low-cost solution of tangible user authentication enabled on any solid surface could eliminate the need of installing touch screens on such electronic devices and make the customized services easy to deploy. Toward this end, our work seeks a general user authentication solution with smart access capability that can work with any solid surface (such as a door, a table or a vehicle's window), not limited to touch screens, and with minimum hardware and maintenance cost.

The traditional authentication solutions are based on passwords (i.e., texts and graphical patterns) [18, 19, 20, 21, 22]. However, all these approaches are based on the knowledge of the passwords, and thus suffer from password theft or shoulder surfing. Another direction of authentication involves physiological biometrics (e.g., fingerprints, iris patterns and face) [23, 24, 25, 26]. These mechanisms are less likely to suffer from identity theft. However, they usually require installation of expensive equipments and stir privacy concerns of the users. Furthermore, recent studies [27, 28, 29] allow users to rely on their familiar biometric-associated features (e.g., a sequence of 2D

handwriting and corresponding pressure) extracted from mobile devices' sensitive touch screens instead of tedious passwords for user authentication. These approaches rely on touch screens, and are hard to be extended to general security access systems such as accessing corporate facilities, apartment buildings and smart homes when touch screens are not always available. In addition, oily residues, or smudges, on the touch screen surface may be used to recover user's graphical password (i.e., smudge attacks) [30].

In this work, we introduce a new authentication system grounded on low-cost, low-power tangible user interface, called *VibWrite*, which has the flexibility to be deployed on ubiquitous surfaces. VibWrite leverages physical vibration to support authentication to emerging smart access security systems. To enable touching and writing on any surface during the authentication process, VibWrite builds upon a touch sensing technique using vibrations that is robust to environmental noise and can operate on surfaces constructed from a broad range of materials. As shown in Figure 1.1(a), when a vibration motor actively excites a surface resulting in the alteration of the shockwave propagation, the presence of the object or finger touching in contact with the surface can thus be sensed by analyzing the vibrations received by the sensor. VibWrite supports generalized vibration sensing based on a low-cost single sensor prototype that can be attached to solid surfaces (such as a door, a table or an appliance) and sense user touches and perform authentication flexibly from anywhere. By relying on the vibration signals in a relatively high frequency band (i.e., over $16kHz$), the system is hardly audible or distracting to the user, and is less susceptible to environmental interference from acoustic (i.e., mainly within a lower frequency band [31]) or radio-frequency noise. More importantly, vibration propagation is highly dependent on the surface material and shape in specific scenarios. VibWrite thus provides enhanced security by integrating location/surface uniqueness through such low-cost and tangible vibration-based user-interface. As another example, the vibration response of an office door is different from that of a house door. The unique behavioral information is embedded in both the behavioral biometrics as well as the surface being touched (e.g., the specific door in the office), making the system hard to be forged by attackers.

VibWrite provides users to choose from three different forms of secrets including

PIN, lock pattern, and gesture (and signature in the future) to gain secure access as shown in Figure 1.1(b). The authentication process can be enabled on any solid surface beyond touch screens and without the constraint of the limited screen size. It is resilient to side-channel attacks when an adversary places a hidden vibration receiver on the authenticating surface or a nearby microphone to capture the received vibration signals. It is also robust to various adversarial activities, including the seemingly very powerful ones that observe the legitimate user's input multiple times and are aware of the passcode secret. It can authenticate the legitimate user and reject attacks well because of the following insights: 1) our study shows that vibration signals have the capability to perform cm-level location discrimination; and 2) unique features are embedded in a user's finger pressing at different locations on a solid surface. Such unique features reflect the characteristics of the user's finger touching on the medium (e.g., a door panel or a desk surface) including locations of touching, contacting area, touching force, and etc., making them capable to discriminate different touching locations of the same user and different users when touching on the same location. Thus, VibWrite enables users to finger-input (i.e., touch or write) on solid surface and is robust to passcode theft or passcode cracking by integrating 1) passcode, 2) behavioral and physiological characteristics (e.g., touching force and contacting area), and 3) surface dependency (e.g., house door or office desk) together to provide enhanced security.

1.3 Snooping Keystrokes with mm-level Audio Ranging on a Single Phone

Mobile and IoT device hardware is increasingly supporting high definition audio capabilities targeted at audiophiles. In particular, this includes microphone arrays for stereo recording and noise cancellation as well as 4x improvement in audio sampling rates. For example, the Samsung Galaxy Note 3 includes three microphones and its audio chips are capable of 192kHz playback and recording. One can debate whether all these advances actually lead to improvements in music playback and audio recording quality that are perceivable by the human auditory system and not all these hardware capabilities are currently made available by drivers and operating system software. Such

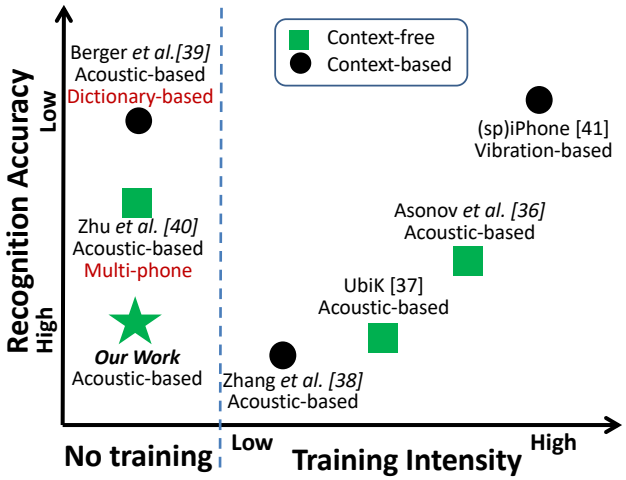


Figure 1.2: Design Space: comparing to related work.

advances, however, could have a significant impact on the accuracy of audio ranging and localization.

Audio localization has been explored with mobile devices to achieve centimeter level accuracy in various applications, such as phone-to-phone ranging and 3D localization [32, 33], mobile motion games [34], and driver phone use detection [35]. Will advanced mobile audio hardware capabilities lead to order of magnitude improvements and let us achieve mm-level accuracy or do the limiting factors lie in multi-path distortions and the accuracy of signal detection techniques?

We explore these questions in the context of keystroke snooping, a particularly challenging localization technique and one with important security implications. To eavesdrop on keystrokes, an adversary can inconspicuously leave a phone near a keyboard of the target user. Or, an adversary can co-opt the target users own phone, for example by adding malware into an app with microphone access. Keystroke snooping is particularly challenging because of the large number of different keys to distinguish and the small cm-level separation between individual keys. It has important security implications because using keyboard is still an important way of entering sensitive information into computing systems and crucially, passwords remain the primary means to authenticate with remote systems, including financial- and health-related services. Besides these security and privacy breaches there is also potential to create improved

input methods for mobile devices that do not directly require typing on the confined mobile screens. Additionally, the proposed solution also has the capacity to facilitate other applications that benefit from fine-grained localization, such as extending interactions with the touch screen of a mobile device to its adjacent surfaces for controlling music players or video games; tracking speakers in multiparty conversations in a meeting room; and locating trapped disaster victims. The proposed audio ranging solution leveraging geometry based information (i.e., time difference of arrival) and unique acoustic characteristics extracted from potential sound sources could deal with many limitations of mobile devices, such as only two stereo recording microphones, limited sampling rate, and restrained distance between two microphones.

Prior work. Existing research has already recognized the significance of this question and found limited potential to recover keystrokes from audio recordings. In particular, Asonov *et al.* [36] conducted an initial study that observed that each key produces unique acoustic emanations and designed a supervised learning method to recognize individual keys. UbiK [37] improves accuracy for keystrokes on solid surfaces (i.e., a paper keyboard on a table) fingerprinting acoustic differences due to multi-path fading. These approaches require extensive labeled training data from the exact keyboard setting to learn the acoustic profiles for each key, which can be challenging to obtain in adversarial scenarios. Later, Zhuang *et al.* [38] propose to add language constraints to improve recognition accuracy. Berger *et al.* [39] further trades off training requirements for accuracy through a dictionary-based approach that leverages the similarity of acoustic signals from nearby keys. Such methods, however, improve keystroke recognition only for natural language and fail for strong passwords composed of random characters. Zhu *et al.* [40] proposes to utilize microphones on three phones to identify keystrokes of a nearby keyboard based on time difference of arrival (TDoA) measurements. The requirements of three collaborating phones and the achieved moderate accuracy make their approach less feasible for real attack scenarios. There is also a related line of work that has explored vibrations sensing of keystrokes using accelerometers such as (e.g., [41]). The accuracy of such approaches generally remains lower than that of audio sensing. Figure 1.2 illustrates the design space and the results offered by existing

work. Due to the limited accuracy, the use of multiple recording devices, the need for linguistic context, or training with extensive labeled data, none of these techniques can easily be applied to snoop on passwords.

Approach. This work demonstrates that the mobile audio hardware advances can indeed be exploited for high accuracy mm-level ranging and that practical scenarios exist where it is possible to localize keystroke sounds with an accuracy sufficient to snoop on passwords. It explores a novel point in the keystroke recognition design space by showing the feasibility of keystroke snooping that is (i) training-free, (ii) context-free, (iii) based on single phone. The approach is training-free because it does not require a-priori labeled training data, which is often difficult to obtain for an adversary. Comparing to the training-based keystroke recovering solutions, e.g., using labeled keystroke data to train a neural network to recognize subsequent keystrokes [36], our work develops unsupervised algorithms without any labeled data to cluster a set of keystrokes. The approach is context-free because it does not require on any linguistic models such as letter, letter sequence (n-gram), or word likelihoods and can therefore be applied to random key sequences such as passwords. And the approach is based on a single phone because it does not require multiple phones or recording devices to be placed around the keyboard; it only relies on two microphones in a single phone.

Our work achieves this by discriminating keystrokes based on the time-difference-of-arrival (TDoA) of the keystroke sound at the two phone microphones and by refining such estimates using acoustic differences in the sound emitted by each key. For certain placements of the phone, relative to the keyboard, there exist measurable differences in TDoA value between most keys. Different from general acoustic TDoA localization approaches which require at least three distributed microphones, our work only uses two microphones with highly constrained distances on a single phone, which produce a limited range of single-dimensional TDoA measurements for locating the keystroke. While a single TDoA measurement will not allow determining a unique 2D location for the keystroke, it does restrict the possible locations for this keystroke to a hyperbola. Given multiple keystrokes of the same key and information about the keyboard geometry, this hyperbola can be placed with mm-level precision so that it uniquely identifies a key. To

obtain multiple audio samples of the same key, even under random typing, the approach clusters keystrokes based on the observed TDoA and mel-frequency cepstral coefficients (MFCCs), which capture (slightly) different acoustic signatures of each keystroke such as those due to physical imperfections across keys. Since the acoustic signatures are only used for improving clustering, there is no need for training of acoustic signatures. Further, since the final TDoA values describe relative locations, they can be directly used to label keystrokes if the keyboard geometry and phone position is known (e.g., keyboard with phone/tablet stand) or if it can be inferred (i.e., enough keystrokes can be observed to derive the key layout). The labeling process only requires finding a best match between the measured TDoA and the expected TDoA for each key, given the geometry and placement.

1.4 High-throughput and Inaudible Acoustic Communication with Non-linearity of Microphones

Short-range wireless communication of mobile devices has become increasingly popular recently, which is targeted to support various mobile applications and services, such as mobile advertisement, mobile payment, and device pairing, etc. In particular, Near-Field Communication (NFC), Bluetooth, screen-camera Quick Response (QR) codes are the most common choices of wireless short-range communication technologies. However, the deployment of NFC chips in mobile devices is far from satisfactory. For instance, there are still over 1 billion mobile devices that are incapable with NFC infrastructures in 2018 [42]. For Bluetooth and QR codes, the requirement of high user-intervention (e.g., complex touchscreen operations and image alignment) leads to incredible inconvenience for users in many application scenarios. As an alternative, acoustic communication has gained considerable attention recently [42, 43, 44]. Different from the aforementioned technologies, acoustic communication builds on the inbuilt microphone and speaker of the devices, without the requirement for user intervention. Because of these benefits and ever-growing market demands, many companies (e.g., Verifone [45], Paytm [46], ToneTag [47]) have started developing acoustic communication

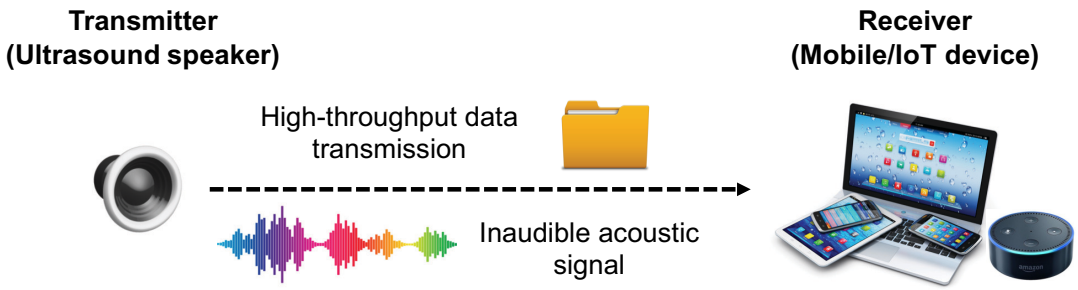


Figure 1.3: Inaudible acoustic communication with off-the-shelf mobile/IoT devices.

techniques for many applications (e.g., highly-secure proximity payments, customer engagement services). Alipay even has launched acoustic communication mobile payments systems on vending machines [48].

In an effective and reliable acoustic communication system, high-throughput (i.e., high-speed) and inaudibility are the two key metrics affecting the possible IoT/mobile applications being supported and their user experiences. For instance, high-throughput communication could enable the delivery of large digital files (e.g., audio, image, PDF files) instead of the limited text message or *url* link. It would also be convenient for developers to design more robust security protocols requiring more overhead for the communication system. Additionally, it is essential to keep the communication process inaudible to humans, making the system not annoying to users and usable anytime and anywhere. Current state-of-the-art audio hardware on mobile devices usually supports up to $48kHz$ sampling rate, thus the upper frequency in the communication frequency band is $24kHz$ according to Nyquist theorem [49]. In order to achieve inaudible communication, existing efforts use near-ultrasound frequency band (i.e., approximately 18-20kHz) [50, 51, 52]. However, using this limited near-ultrasound frequency bandwidth cannot achieve satisfactory high throughput.

This work proposes the first acoustic communication system, BatComm, which can achieve inaudibility and high-throughput simultaneously by using the non-linearity of microphones (i.e., described with details in Section 5.2). As shown in Figure 1.3, an ultrasound speaker transmits acoustic signals modulated on an ultrasound frequency carrier (e.g., $> 40kHz$). Relying on the non-linearity of microphones, a nearby mobile

or IoT device could pick up the signals of the entire audio frequency band (i.e., 0-$24kHz$) that are modulated onto this ultrasound frequency carrier to receive data. The proposed solution could facilitate many mobile and IoT applications requiring high-speed data delivery functionalities, such as contents sharing between devices and inaudibly broadcasting messages (e.g., any digitized files and advertisements) in public facilities such as auditoriums, elevators, theaters, libraries and art museums.

Existing studies achieve a relatively high throughput (e.g., $1kbps$ [53, 54], $2.4kbps$ [55]) through using audible acoustic frequency band (e.g., 0-$18kHz$ [53], 0-$22kHz$ [55], 8-$10kHz$ [54]). However, due to the audible acoustic signal used for communication, these approaches are annoying to users thus degrade the user experience. In addition, several approaches [56, 57, 58] embed the data signals for communication underlying the daily sounds (e.g., music, speech) leveraging the information-hiding technique. These approaches use 6-$20kHz$ frequency band, which is non-overlapped with those daily sounds' frequencies, for data communication. Additionally, some studies [50, 51] directly utilize near-ultrasound band (e.g., 18-$20kHz$) to realize an inaudible acoustic communication. However, the aforementioned approaches can only achieve a relatively low throughput (e.g., $1kbps$) due to the narrow bandwidth (i.e., around $2kHz$) that can be used. In this work, we design a communication solution with both high-throughput (i.e., $> 40kbps$) and inaudibility, which provides a powerful communication channel for the devices equipped with microphones.

To achieve high-throughput communication for general mobile devices with limited audio sampling rate (e.g., $48kHz$), BatComm (1) applies orthogonal frequency division multiplexing (OFDM) to transmit the data bits on multiple subcarriers concurrently; and (2) uses wider bandwidth (e.g., entire audio frequency band) rather than the limited near-ultrasound band. To make the whole communication process inaudible, we use amplitude modulation (AM) to modulate the low-frequency signals (e.g., $< 24kHz$) on an ultrasound frequency band (e.g., $> 40kHz$) at the transmitter end, while integrating the non-linearity of device's microphone to fully recover the low-frequency signals from the ultrasound signals at the receiver end. Additionally, we theoretically and empirically demonstrate that the unrelated residual signals produced by AM modulation under the

non-linearity of microphones would interfere with the signals transmitting on other OFDM subcarriers, which induces errors in the received data. To address this issue, we propose an elimination scheme, which elaborately modifies the OFDM waveform before AM, to eliminate the unrelated residual signals in the recorded signals. To make BatComm robust in realistic settings, we also apply a series of techniques (e.g., DPSK, preamble, cyclic prefix, channel estimation, BCH code, interleaving) to handle various interference and fading problems including frequency selective fading, time selective fading, inter-symbol interference and practical ambient noises.

Chapter 2

Fine-grained Sleep Monitoring Leveraging Off-the-shelf WiFi

2.1 System Design

In this section, we discuss the preliminaries, design challenges and overview of our system design.

2.1.1 Preliminaries

While proliferating WiFi networks are usually used for wireless Internet access and connecting local area networks, such as an in-home WiFi network involving both mobile and stationary devices (e.g., laptop, smartphone, tablet, desktop, smartTV), they have great potential to sense the environment changes and capture the minute movements caused by human body [59]. Indeed, WiFi signals are affected by human body movements at various scales during sleep, such as large scale movements involving going to bed and turn over, minute movements including inhaling/exhaling for breathing and diastole/systole for heart beats. By extracting and analyzing the unique characteristics of WiFi signals, we could capture and derive the semantic meanings of such movements including both breathing rate and heart beats during sleep. We are thus motivated to re-use existing WiFi network to monitor the fine-grained vital signs during sleep as it doesn't require any dedicated/wearable sensors or additional infrastructure setup.

To monitor the minute movements of breathing and heart beats, we exploit the Channel State Information (CSI) provided by off-the-shelf WiFi devices as opposed to the commonly used Received Signal Strength (RSS). While the coarse-grained channel information of RSS provides the averaged power in a received radio signal over the whole

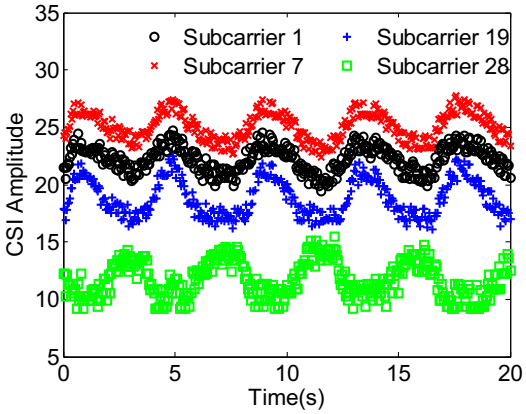


Figure 2.1: CSI amplitude of four subcarriers over time when a person is asleep.

channel bandwidth, the fine-grained CSI of WiFi signal (based on OFDM) describes at each subcarrier how a signal propagates from the transmitter to the receiver and represents the combined effect of, for example, scattering, fading, and power decay with distance. For example, in 802.11 a/g/n, a relatively wideband $20MHz$ OFDM channel (or carrier) is partitioned into 52 subcarriers. And we could examine the amplitude and phase at each subcarrier, which could be thought of as a narrowband channel, for extracting the minute movements. Due to the relative narrowband channel, the scattering and reflecting effects caused by minute movements could result in totally different amplitudes and phases at each subcarrier. Such difference however is usually smoothed out if we look at the averaged power over the whole channel bandwidth (i.e., RSS). Analyzing the CSI at each subcarrier thus provides great opportunity to capture the minute movements from not only breathing but also heart beats.

Figure 2.1 shows the CSI amplitude of four subcarriers (i.e., subcarrier 1, 7, 19 and 28) extracted from a laptop in a 802.11n network over time when a person is asleep. His bed is in between an AP and the laptop with 3 meters apart. The person does not carry any sensor in his body. We observe that the CSI amplitude of these four subcarriers exhibits an obvious periodic up-and-down trend. Such a pattern could be caused by the person's breathing during sleep. This observation strongly suggest that we may achieve device-free fine-grained vital signs monitoring by leveraging the CSI from off-the-shelf WiFi devices.

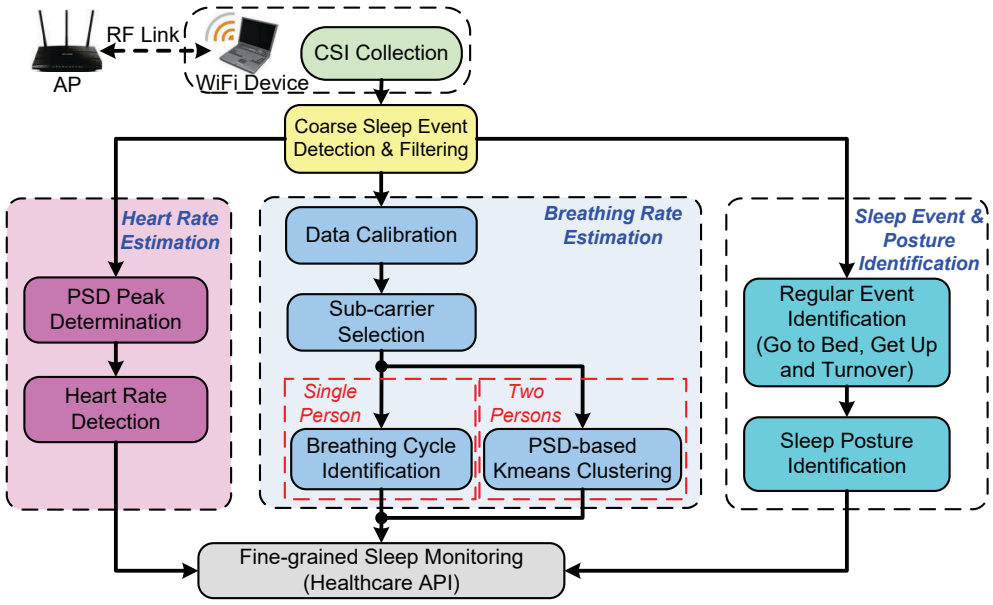


Figure 2.2: Overview of system flow.

2.1.2 Challenges

Our goal is to track human vital signs of breathing and heart rates simultaneously using CSI measurements from a single pair of WiFi devices. To build such system under realistic settings as a typical in-home scenario, a number of challenges need to be addressed.

Robustness to Real Environments. The placement of WiFi devices in real environments could change over time, and different persons present different sleeping postures. Our system should be able to provide accurate vital sign monitoring under such challenging conditions including various distances between the AP and WiFi devices, presence of walls between WiFi devices (creating none-line-of-sight (NLOS) scenarios), and different sleeping postures. In addition, our system should be able to identify regular sleep related events (such as turnover or getting out of bed) to facilitate vital signs monitoring.

Tracking Breathing & Heartbeat Simultaneously. Both breathing and heart beat only involve small body movements, presenting significant challenges when tracking such vital signs simultaneously under realistic settings. Even if the repeatable CSI changing pattern caused by breathing could be detected as shown in Figure 2.1, it is

difficult to capture heartbeat movements using WiFi links at the same time. Because the noisy environments will also affect CSI measurements, making it much harder to distinguish the minute movements caused by breathing (i.e., inhaling and exhaling) and heart beats (i.e., diastole and systole).

Sensing with Single Pair of AP and WiFi Device. Our approach should work with existing WiFi infrastructure, which may have only a single wireless link (between the AP and the device) across the human body. This presents additional challenges when two people are in-bed together. Our system should be able to distinguish and measure breathing rates coming from two people. Furthermore, the system should use WiFi traffic as little as possible, such as only utilizing existing beaconing traffic.

2.1.3 System Overview

The basic idea of our system is to track vital signs during sleep through capturing the unique patterns embedded in WiFi signals. As illustrated in Figure 2.2, the system takes as input time-series CSI amplitude measurements, which can be collected at an off-the-shelf WiFi device by utilizing existing WiFi traffic or system-generated periodic traffic (if network traffic is insufficient) during people's sleep. The data is then processed to filter out the CSI measurements that contain sleep events (e..g, going to bed and turn over) or large environmental changes such as people walking by via *Coarse Sleep Event Detection and Filtering*. The measurements belonging to the regular sleep events can be further classified to detailed events such as going to bed, getting off bed and turnovers. Additionally, sleep posture plays an important role for people's sleep status/quality. For instance, some bad sleep postures (e.g., sleeping on the stomach) may be the cause of people's back and neck pain, stomach troubles [60]. The system thus would identify people's sleep posture using a machine learning based approach via *Sleep Posture Identification*. Moreover, our work is based on the fact that breathing and heart rates of resting people have different frequency ranges (e.g., breathing rate ranges from 10 to 37 bpm [61, 62], and heart rate ranges from 60 to 80 bpm [63]). This useful information leads us to work on different frequency bands of the CSI measurements for accurate vital signs estimation.

The core components of our system are *Breathing Rate Estimation* and *Heart Rate Estimation*. After coarse sleep event detection and data filtering, based on the different frequency information embedded inside the CSI measurements, the input is fed into *Breathing Rate Estimation* and *Heart Rate Estimation* respectively. In particular, the lower-frequency information of the CSI measurements is processed by the *Breathing Rate Estimation* component. Our system first performs *Data Calibration* and *Subcarrier Selection* to preprocess the data and select only the subcarriers sensitive to minute human body movements (i.e., subcarriers with large variances). We then develop two methods, *Breathing Cycle* and *PSD-based K-means Clustering*, to estimate the breathing rate for single and two-person in-bed scenarios respectively. *PSD* denotes power spectral density. Following the similar principle, *PSD-based K-means Clustering* can be easily extended to handle the case of estimating breathing rates for multiple people simultaneously given the number of people under study is known. The higher-frequency information of the CSI measurements is fed into the *Heart Rate Estimation* component. The heart rate is then derived in the frequency domain by examining the peaks in power spectral density (PSD) of CSI measurements. We leave the detailed presentation of *Breathing Rate Estimation* and *Heart Rate Estimation* to Section 2.2 and Section 2.3, respectively.

2.2 Breathing Rate Estimation

We first describe *Data Calibration* and *Subcarrier Selection*, and then present *Breathing Cycle Identification* for estimating an individual's breathing rate. We finally show how to estimate breathing rates for two persons in-bed case.

2.2.1 Data Calibration

Data calibration is used to improve the reliability of the CSI by mitigating the noise presented in the collected CSI samples in real environments. The noise sources could come from environment-related changes, radio signal interference, etc. Our data calibration first utilizes the Hampel filter [64] to filter out the outliers which have

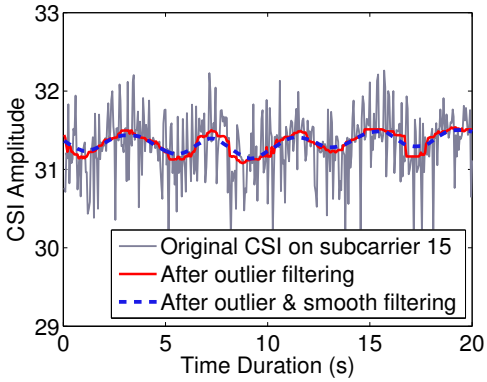


Figure 2.3: Illustration of data calibration of a single subcarrier.

significant different values from other neighboring CSI measurements. Specifically, we apply the Hampel filter with a sliding window at each subcarrier to remove such outliers.

For CSI amplitude sequence $(x_1, x_2, ..., x_N)$ at each subcarrier, the Hampel identifier defines outliers as those data points x_i whose absolute difference from the median value is greater than a pre-determined threshold, as defined by

$$\begin{cases} |x_i - x^*| > t \cdot M & \text{outlier;} \\ |x_i - x^*| \leq t \cdot M & \text{normal measurement,} \end{cases} \tag{2.1}$$

where i is from 1 to N, and x^* represents the median value of the rank-ordered samples of a data sequence of length N. t is a scalar threshold and M is the median absolute difference (MAD) scale estimate, as defined by equation (2.2):

$$M = 1.4286 \cdot \text{median}\{|x_i - x^*|\}, \tag{2.2}$$

where the constant value 1.4286 ensures that the expected value of M equals the standard deviation of normally distributed data [65].

After that, we further apply a moving average filter, which further removes high-frequency noise that is unlikely to be caused by breathing or heart beats as the corresponding minute movements usually present in a fixed frequency range. Figure 2.3 illustrates the effectiveness of our data calibration by comparing the CSI amplitude before and after data calibration under a none-line-of-sight case with severe signal outliers: the CSI amplitude shown in the figure is from a single subcarrier collected from a WiFi device, which trasmits/receives packets from an AP with a wall between them.

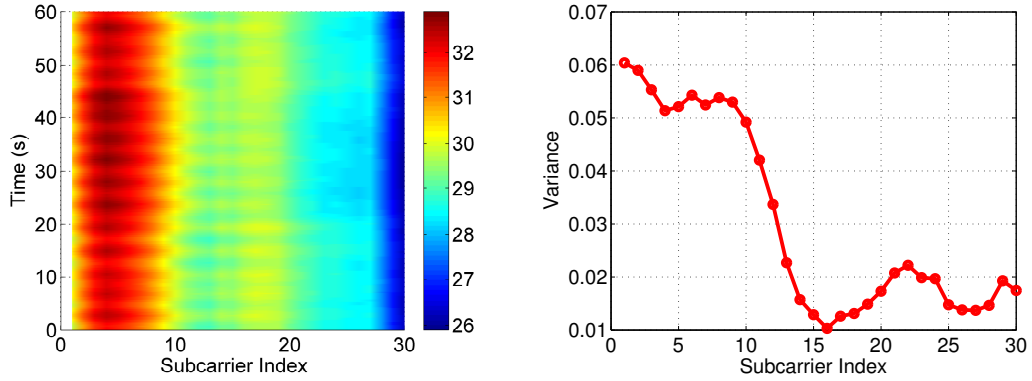
As we can see from the figure, after data calibration, the sinusoidal waves in CSI amplitude can clearly reflect the periodic up-and-down chest and belly movements caused by breathing.

2.2.2 Subcarrier Selection Strategy

We observe that the amplitudes of different subcarriers have different sensitivity to inhaling and exhaling caused by breathing due to frequency diversity. Figure 2.4(a) presents an example of CSI amplitude over time on 30 subcarriers extracted from a laptop in WiFi network when a person is asleep. We find that the CSI from the smaller subcarrier indices is significantly affected by the minute movements caused by breathing, while CSI from the higher subcarrier indices (i.e., from 15 to 30) is less sensitive. This is because different subcarriers have different central frequencies, which have different wavelengthes. Combining the effect of multipath/shadowing with different frequencies, CSI measurements at different subcarriers thus have different amplitudes. Those subcarriers not sensitive to the breathing activity should be filtered out. We utilize the variance of CSI amplitude in a moving time window to quantify the subcarrier's sensitivity to minute movements. Figure 2.4(b) shows the variance of 30 subcarriers. We can see that subcarriers with higher variance are more sensitive to minute movements. We thus use a threshold based method to select subcarriers having large variance of CSI amplitude in a time window for breathing rate estimation.

2.2.3 Breathing Cycle Identification

As breathing involves periodic minute movements of inhaling and exhaling, our breathing cycle identification aims to capture the periodic changes in CSI measurements caused by breathing. From Figure 2.1, we observe the CSI amplitude on the selected subcarrier indeed presents a sinusoidal-like periodic changing pattern over the time due to breathing. This observation suggests that we can identify breathing cycles by measuring the peak-to-peak time interval of sinusoidal CSI amplitudes. We thus first identify peaks of sinusoidal CSI amplitude patterns to calculate peak-to-peak intervals. We then combine the peak-to-peak intervals from multiple subcarriers to improve the



(a) CSI time series patterns after data calibration. (b) Variance of each subcarrier.

Figure 2.4: Example of CSI amplitude pattern at 30 subcarriers and the corresponding variance.

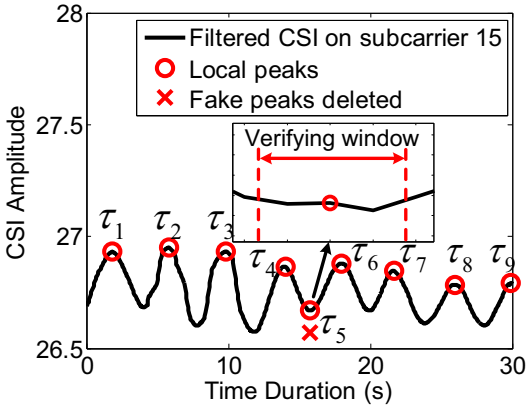


Figure 2.5: Illustration of fake peak removal.

robustness and the accuracy of breathing cycle identification.

Local Peak Identification. A typical peak finding algorithm determines a data sample as a peak if its value is larger than its two neighboring samples. However, such simple method produces many *fake peaks* (i.e., the identified peaks that are not at the location of real peaks of the sinusoidal CSI amplitude pattern) as illustrated in Figure 2.5. The peak τ_5 has larger value than its two neighboring samples, yet, it is a fake peak among these nine identified peaks. In order to filter out the fake peaks, we apply a threshold to the minimum distance between two neighboring peaks based on human's maximum possible breathing rate. In addition, we develop a *Fake Peak Removal* algorithm to further reduce the number of fake peaks.

Specifically, adults usually breathe at 10-14 breathes per minute (bpm) [62], while new born babies breathe at around 37 bpm [61]. We therefore set the range of breathing rates being considered in our work to 10-37 bpm which covers a broad range including fast and slow breathing rates. We further adopt a minimum acceptable interval σ_{mpd} that corresponds to the maximum possible breathing rate as a threshold to remove the peaks that are too close to each other. If a peak has its backward interval (i.e., the interval between previous peak and current peak) less than the minimum acceptable interval length, it will be identified as a fake peak. In particular, we set the minimum acceptable interval $\sigma_{mpd} = 60 \cdot f/37$ samples, which corresponds to the maximum possible breathing rate for infants. The parameter f is the sampling rate of CSI measurements that corresponds to WiFi packet transmission rate.

In addition, we confirm the identified peaks by comparing its value to multiple data samples within a verification window centered at the peak. The system only keeps the identified peak when its value is greater than all the data samples in the verification window. The algorithm of fake peak removal is provided in Algorithm 1. In our experiments, we observe that a short verification window of one second is good enough to remove fake peaks.

Breathing Cycles Combination. Once we capture all the local peaks from the selected subcarriers, a more clear pattern can be obtained as shown in Figure 2.6. The referenced signal is derived from the NEULOG Respiration Monitor Logger Sensor [66], which is connected to a monitor belt attached to the user's ribcage while asleep. Next, our system estimates the breathing rate by combining peak-to-peak intervals obtained crossing all selected subcarriers. We denote a set of peak-to-peak intervals obtained from P selected subcarriers as $L = [l_1, ..., l_i, ..., l_P]'$, where $l_i = \{l_i(1), ..., l_i(N_i - 1)\}$ is a vector of N_i peak-to-peak intervals obtained from the i^{th} subcarrier. Then the estimated breathing cycle E_i from the i^{th} subcarrier can be obtained by using the following equation:

$$\arg\min_{E_i} \sum_{n=1}^{N_i-1} |E_i - l_i(n)|^2. \tag{2.3}$$

Considering the subcarriers with larger variance are more sensitive to the minute movements, we utilize a weighted mean of estimated breathing cycles crossing all selected

Algorithm 1 Fake Peak Removal.

Require:
CSI time series on subcarrier i: $c_i = \{c_i(1), ..., c_i(M)\}$;
Local peak set: $MaxSet = \{\tau_k, 1 \leqslant k \leqslant K\}$;
Length of the verifying window: N;
Ensure:
$MaxSet$ after removing fake maximums;

```
for k=1: K do
  locs := location(τ_k);
  amp := amplitude(τ_k);
  for m := locs-⌊(N-1)/2⌋ : locs+⌊(N-1)/2⌋ do
    if m < 0 || m > M then
      continue;
    end if
    if amp < c_i(m) then
      delete τ_k from MaxSet;
      break;
    end if
  end for
end for
return MaxSet;
```

subcarriers to obtain a more accurate estimation of breathing cycle E, which is defined as follows:

$$E = \sum_{i=1}^{P} \frac{\text{var}(c_i) \cdot E_i}{\sum_{i=1}^{P} \text{var}(c_i)}, \tag{2.4}$$

where P is the number of validated subcarriers, c_i is the CSI amplitude measurements on the i^{th} subcarrier. The breathing rate finally can be identified as $60/E$ bpm.

2.2.4 Breathing Rate Estimation of Two Persons Scenario

Estimating breathing rate becomes challenging when there are two persons in bed as the CSI measurements would be affected by two independent movements simultaneously due to breathing. It is hard to observe a clear sinusoidal pattern in the time series of CSI amplitude. Nevertheless, the frequency of the breathing coming from two persons is still preserved if we transfer the time series of CSI to the frequency domain. We therefore develop a mechanism to determine two people's breathing rates simultaneously by examining the frequency components in CSI measurements.

In particular, our system analyzes the time series of CSI amplitude in frequency

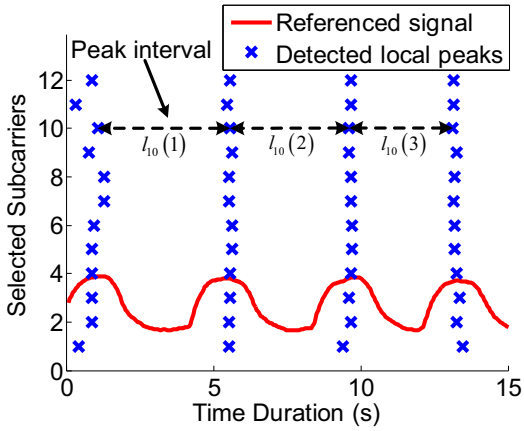


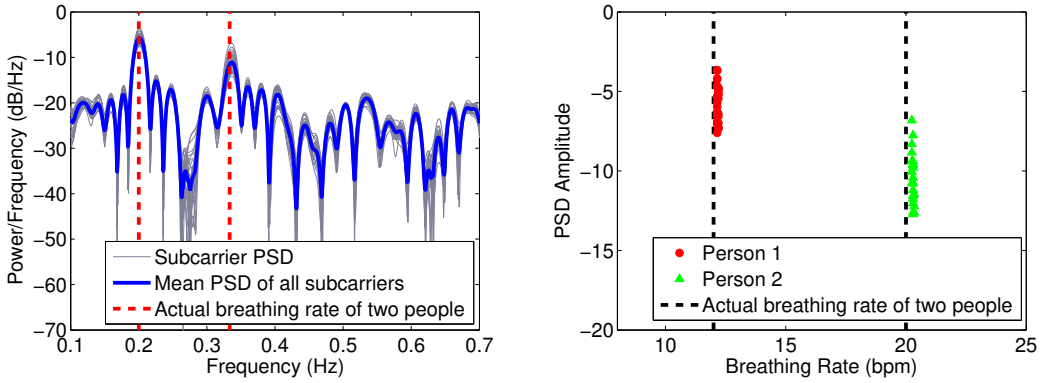
Figure 2.6: Local peaks of all selected subcarriers.

domain by using the power spectral density (PSD). The PSD transforms the time series of CSI measurements on each subcarrier to its power distribution in the frequency domain. It is used to identify the frequencies having strong signal power. A strong sinusoidal signal generates a peak at the frequency corresponding to the period of the sinusoidal signal in PSD. Therefore, the CSI amplitude measurements collected when two persons in bed should present two strong peaks at the frequency corresponding to the breathing rate of two persons, respectively. The PSD on the i^{th} subcarrier with N CSI amplitude measurements can be calculated with following equation:

$$PSD_i = 10 \log_{10} \frac{(abs(FFT(c_i)))^2}{N}, \tag{2.5}$$

where c_i is the vector of CSI measurements on subcarrier i.

For each selected subcarrier, we utilize a threshold based approach to identifying the candidate peaks within its PSD. We then use a K-means clustering method to classify the candidate peaks into two clusters based on two dimensional feature including PSD amplitude and corresponding frequency. The number of targeted people (i.e., K in K-means) can be either estimated using existing work (e.g., [67]) or entered manually from the users. The average values of the frequencies in two clusters are identified as the breathing rates of these two people. Figure 2.7 shows an example of estimating two persons' breathing rate using PSD based method. The ground truths of two persons' breathing rates are $12bpm$ and $20bpm$ (i.e., $0.2Hz$ and $0.33Hz$ respectively).



(a) PSD of CSI measurements.

(b) K-means clustering of subcarriers' peaks.

Figure 2.7: Illustration of two people breathing at different frequencies ($12bpm$ and $20bpm$).

Figure 2.7(a) depicts that there are two strong peaks in the PSD of selected subcarriers near these two frequencies, respectively. Figure 2.7(b) shows that our PSD based K-means clustering method can effectively estimate the breathing rates of two persons in bed simultaneously. We note that the proposed approach still works even when two people have the same breathing rates. Under such scenario, our approach returns two close-by PSD peaks on each selected subcarrier in the frequency domain after K-means clustering. In addition, the person's chest or belly that is closer to the wireless link has bigger impact on the CSI changes, which creates more obvious periodic changes of CSI. This leads to the stronger peak corresponds to that person's breathing or heart beat rate. We thus can map the detected breathing or heart rates to each individual based on the strength of the peak and the proximity of the individual to the wireless link.

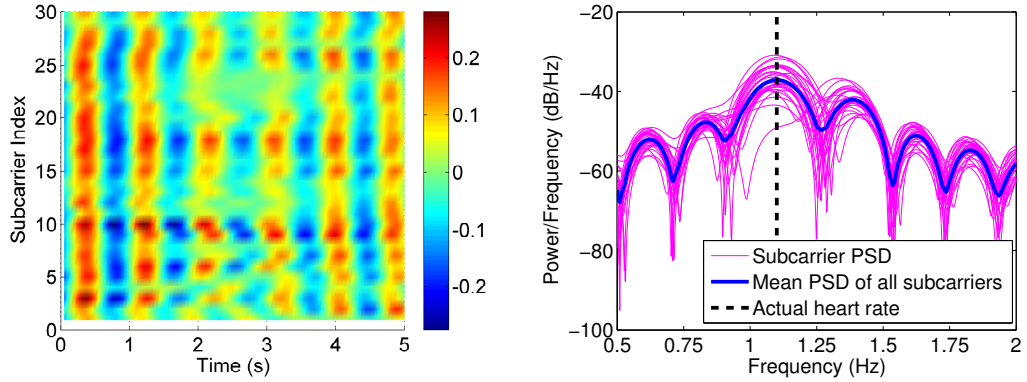
2.3 Heart Rate Estimation

Heart rate is a very important indicator of the persons' sleep status, quality and overall health condition. While the breathing patterns can be observed in the CSI measurement, the heart rates don't produce observable periodic CSI change patterns in the time series CSI measurements. This is because the vibration of blood vessels caused by heart beat (i.e., diastole and systole) are smaller minute movements than that of breathing. Thus, the effect of minute movement of heartbeat is overlapped

with and covered by the chest and belly movements of breathing. On the other hand, the heartbeat has much higher frequency than breathing. We thus can filter out the interference of breathing in order to facilitate the heart rate estimation.

In particular, after Coarse Sleep Event Detection and Filtering, the CSI measurements with the frequency range related to normal heart rate range of resting people (i.e., $60bpm$ to $80bpm$ which corresponds to $1Hz$ to $1.33Hz$) will be separated and served as input to our Heart Rate Estimation. The patterns of CSI measurements of all subcarriers after such band-pass filtering are illustrated in Figure 2.8(a), from which we can observe the CSI changing that accompany the heart beats. With the aid of the band-pass filter, the mean PSD curve for all subcarriers displays a noticeable peak in the PSD graph at the frequency of $1.095Hz$, namely $65.7bpm$, in Figure 2.8(b). In the same figure, there is a black dashed line representing the ground truth of $66bpm$ measured by a commercial fingertip pulse sensor during such time period. We then analyze the CSI amplitude on each subcarrier in frequency domain and generate the power spectral density (PSD) (refer to Equation 2.5) to identify the frequencies having strong signal power. We can thus determine the heart rate by locating the maximum power in the average PSD of all subcarriers in the normal heart rate range. For two person's heart rates monitoring, we can identify two heart rates simultaneously by using the similar approach to the breathing rate estimation of two persons illustrated in Section 2.2.4.

In addition to heart rates, fine-grained heart movement metrics (e.g., the heart rate variability and R-R interval) have been shown to be good predictors for many possible heart diseases [68]. We find that the normalized CSI can well capture the detailed heart movement information from WiFi signals. Particularly, we pre-process the raw CSI readings on each subcarrier via the aforementioned band-pass filter, and sum each subcarrier's readings together to get the normalized CSI. In the experiment, we placed the WiFi device and AP at two sides of the bed with the distance of $5ft$, and the line of sight between the WiFi device/AP is crossing the person's chest, so that our system can well capture the user?s minute body movements associated with the heart beats. Due to the vibrations of blood vessels caused by the diastole and systole of a



(a) CSI time series patterns.

(b) PSD of CSI measurements on all subcarriers.

Figure 2.8: Recovered heart beats by applying pass band filtering and PSD of CSI measurements.

heart, the human body usually has slight movements when the heart beats. Similar to the body movements caused by breathing, the even smaller movements associated with heart beats also result in different amplitudes and phases at each subcarrier of WiFi signals. After the band-pass filtering with the pass band limited to the frequency range of human heart rate, the peaks/valleys in the CSI patterns can be used to measure the heart contracts and cardiac diastole motions. Figure 2.9 compares the normalized CSI patterns to a wrist-worn photoplethysmogram (PPG) sensor's readings when the user is asleep. The PPG sensor is usually used in clinical scenarios for collecting accurate heart rates and detailed heart movement metrics. From Figure 2.9 we can see that the changing pattern of the normalized CSI is highly correlated with the readings from the PPG sensor, indicating that the normalized CSI obtained from WiFi signals could be utilized to extract the fine-grained heart movement metrics such as heart contracts and cardiac diastole (i.e., peak/valley in the corresponding CSI patterns [69]).

2.4 Sleep Event & Sleep Posture Identification

2.4.1 Coarse Sleep Event Detection & Environmental Change Filtering

Coarse sleep event detection and filtering is used to detect and filter out the sleep events or environmental changes that interfere with the minute movements of breathing

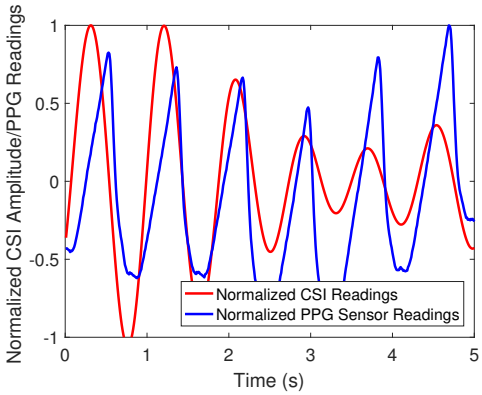


Figure 2.9: Comparison of normalized CSI patterns and PPG sensor readings.

and heart beat during sleep. These sleep events, such as turnovers (i.e., changing sleeping postures) and getting up, and occasional changes of environments, such as people walking by, involve large scale body movements which significantly affect the CSI measurements and are irrelevant to vital signs monitoring. Our system thus performs coarse determination of CSI segments containing such inference factors and filters them out to facilitate accurate vital signs monitoring during sleep.

In particular, we employ a threshold-based approach to determine whether a segment of CSI measurements contains sleep events/environmental changes or not by examining the short-time energy of the moving variance of the CSI measurements. The rationale behind this is that the sleep events or environmental changes involving large body movements (e.g., going to bed and turn over) result in much larger changes of CSI measurement than that of minute movements of breathing and heart beat. The large movements thus can be detected once the variance energy of the corresponding CSI measurements exceeds a particular threshold.

We denote the CSI samples of P subcarriers as $C = [C_1, \ldots, C_p, \ldots, C_P]'$, where $C_p = \{c_p(1), \ldots, c_p(T)\}$ represents T CSI amplitudes on the p^{th} subcarrier. We further denote the moving variances of the P subcarriers as $V = [V_1, \ldots, V_p, \ldots, V_P]'$, where $V_p = \{v_p(1), \ldots, v_p(T)\}$ are the moving variances derived from C_p. Our system can then calculate the cumulative moving variance energy of CSI samples accessing P subcarriers

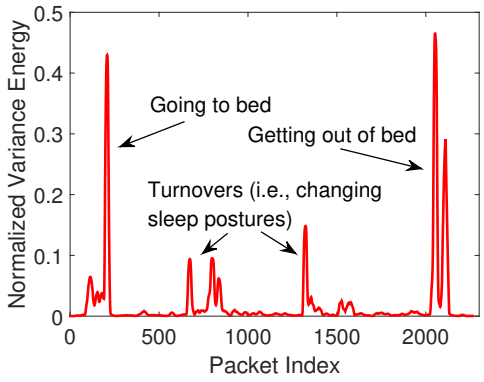


Figure 2.10: Short time energy of the variance of difference sleep events.

as:

$$E = \frac{1}{NP} \sum_{p=1}^{P} \sum_{n=1}^{N} |v_p(n)|^2, \tag{2.6}$$

where N denotes the window length of short time energy.

We empirically determine the variance energy to be 0.02 as the threshold in this work. Figure 2.10 illustrates the normalized moving variance energy of CSI measurements that are collected when the participant involves different sleep events during sleep. We observe that all sleep events generate significantly large variance energy comparing to that of the minute movements of only breathing and heart beats.

2.4.2 Regular Sleep Event Identification

Given the detected sleep events, we further classify them into detailed events such as going to bed, getting off bed and turnovers. Generating statistic of such detailed events can help quantify the sleep quality. For example, frequent getting up or turning overs may suggest that the person has difficulty falling asleep. This information contributes to many healthcare applications such as elderly care and medical diagnosis. As shown in Figure 2.10, sleep events involving relative larger-scale movements (i.e., going to bed and getting out of bed) result in much larger variance energy than those involving relative smaller-scale movements (i.e., turn overs). We thus can distinguish sleep events with larger-scale movements from those with smaller-scale movements by comparing the variance energy from Equation (2.6). To further distinguish larger-scale movements, we can exploit the changes of the number of persons in bed to infer these two events. The

number of persons in bed can be obtained by using a profile based approach as studied in existing work (e.g., [67]).

2.4.3 Sleep Posture Identification

Sleep posture/position also plays an important role on a good night's sleep. A comfortable sleep posture could make a person easier to align his head, neck, spine, and keep them in a neutral position, whereas some bad sleep postures (e.g., sleeping on the stomach) may be the cause of people's back and neck pain, stomach troubles, etc. [60]. Moreover, some researchers also found that different sleep postures incur different health effects. For example, the freefall posture is good for digestion, while the starfish and soldier positions are more likely to lead to snoring and a bad night's sleep [70]. This encourages us to identify and track people's sleep postures using WiFi signals, which could provide additional sleep information to assist identifying potential reasons of sleep difficulty or health problems. Intuitively, different sleep postures have inevitable influence to WiFi signals, therefore we propose to match the features extracted from CSI with the trained profiles to differentiate sleep postures.

Feature Extraction & Selection. In particular, we use a sliding window whose length is 5 seconds on the calibrated CSI time series (after the *Data Calibration* that is discussed in Section 2.2.1) and extract nine basic features including *mean, maximum, minimum, variance, skewness, range, mode, median* and *kurtosis* on each subcarrier group. Therefore, for the 30 subcarrier groups, we could have 270 features in total for each time window. In addition, since not all wireless signal transmission paths would be influenced by people's different postures, we find that only a few subcarriers or features are distinguishable enough to differentiate these sleep postures. We thus select a subset of features that are more unique between different sleep postures from the 270 extracted features on 30 subcarrier groups based on Fisher Score [71]. The fisher score of the i-th feature is defined as follows:

$$F_i = \frac{\sum_{j=1}^{c} n_j(\mu_j - \mu)^2}{\sum_{j=1}^{c} n_j \delta_j^2}, \tag{2.7}$$

where n_j is the number of instances in sleep posture class j, $j = 1, ..., c$, μ_j and δ_j^2 denote

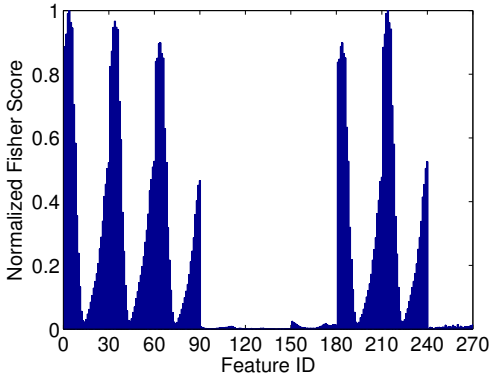


Figure 2.11: Derived fisher score of the extracted 270 features on 30 subcarrier groups (features $1-30$ are the first feature *mean* on 30 subcarrier groups, and so on so forth).

the mean and variance of class j corresponding to the i-th feature, and μ denotes the mean of i-th feature candidates in the whole training data sets. Figure 2.11 shows the normalized fisher scores of those 270 features spanning on 30 subcarrier groups that we extract to discriminate different sleep postures. Figure 2.11 shows the normalized Fisher scores of the nine types of features extracted from 30 subcarrier groups, every 30 Fisher scores in this figure correspond to one type of the features. From the figure, we observe that, for a particular type of feature, not all the subcarriers have high Fisher scores (e.g., presenting a *V-shape* pattern), which means they are not equally sensitive to human body movements. Note that such sensitivity differences are often caused by the relative position of the AP and WiFi device to the human body. In addition, we observe that the features *variance, skewness, range* and *kurtosis* (i.e., feature ID $91-180$ and $241-270$) with low fisher score are not representative for each posture. In order to reduce the impact of non-sensitive features and subcarrier groups to the sleep posture identification, we empirically choose a threshold (i.e., $\tau_f = 0.1$) and only use the features having Fisher scores larger than the threshold for the sleep posture identification.

PCA Dimension Reduction. In order to further reduce the computational cost in the later classification process, we adopt Principal Component Analysis (PCA) [72] which not only converts original feature vectors into a set of linearly uncorrelated principal components but also removes uncorrelated noise components in the features. Specifically, we adopt PCA to convert the selected features in each time window into

20 linearly uncorrelated principal components.

Posture Training and Identification. Our system mainly focus on identifying four typical sleep postures, including *curl up, supine, prone,* and *recumbent*, which are illustrated in Figure 2.12(b). Given a specific WiFi device setup, our system first constructs the four sleep posture profiles with the extracted CSI features. Then the four posture profiles are respectively used to train a machine learning based classifier. Finally, in the sleep posture identification phase, CSI measurements and their corresponding features collected while the user is sleeping are fed into the classifier to identify the user's posture. We compared the performance of using four different classifiers including Discriminant Analysis, k-nearest neighbors (k-NN), Support Vector Machine (SVM), and Random Forest, which are described in Section 3.6.

2.5 Performance Evaluation

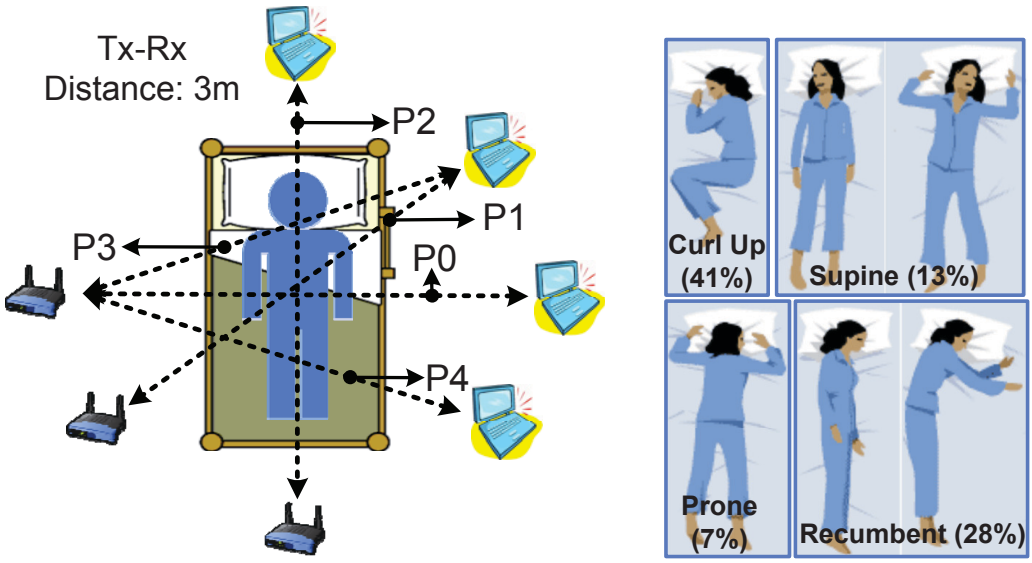
In this section, we evaluate our system of tracking vital signs during sleep in both lab and two apartments.

2.5.1 Device and Network

We conduct experiments in an 802.11n WiFi network with a single off-the-shelf WiFi device (i.e., Lenovo T500 Laptop) connected to a commercial wireless Access Point (AP) (i.e., TP-Link TL-WDR4300). The laptop runs Ubuntu 10.04 LTS with the 2.6.36 kernel and is equipped with an Intel WiFi Link 5300 card for measuring CSI [73]. Unless mentioned otherwise, the packet transmission rate is set to $20 pkts/s$. How the packet rate affects the performance will be discussed in Section 2.5.6. For each packet, we extract CSI for 30 subcarrier groups, which are evenly distributed in the 56 subcarriers of a $20MHz$ channel [73].

2.5.2 Experimental Methodology

The experiments are conducted in both lab and two apartments with 6 participants over a three-months time period. The lab environment is a large room with office cubic



(a) Setup of WiFi device-AP pair with different relative positions.

(b) Different sleep postures in bed [70].

Figure 2.12: Setup of relative position of WiFi device and AP and sleeping postures.

around. It is used to study the impact of various factors such as obstacles, the various distances between the AP and the WiFi device, and sleep postures. In breathing rate estimation experiments, the participants lie on a bed and control their breathing rate to follow various steady beats from a metronome, which is set to a rhythm ranging from $12bpm$ to $18bpm$.

We also conduct experiments in two apartments with different bedroom sizes. Figure 2.13 illustrates the environmental setup in two bedrooms, in which both beds are queen size. The smaller one (i.e., bedroom-1) has the size of about $12ft \times 9ft$, whereas the larger one (i.e., bedroom-2) is about $15ft \times 12ft$. As shown in Figure 2.13, we have three setups in both apartments: setup 1 is the ideal scenario where the AP and WiFi device are placed at two sides of the bed. This setup is useful for persons who want to optimize the performance of the vital signs monitoring during sleep. Setup 2 represents a typical scenario where there is a AP inside the room and a WiFi device, such as smartphone, laptop or tablet, is put on the bed table. Setup 2 has larger distance between the AP and the WiFi device than setup 1. Setup 3 is a challenging scenario where the AP and the WiFi device are placed in different rooms with a concrete wall between them. The distance between the AP and the WiFi device is the largest among

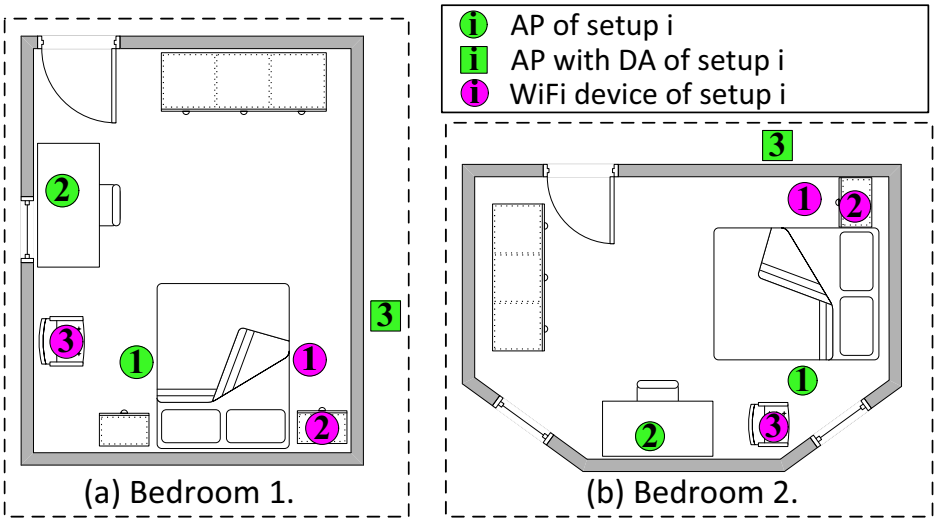


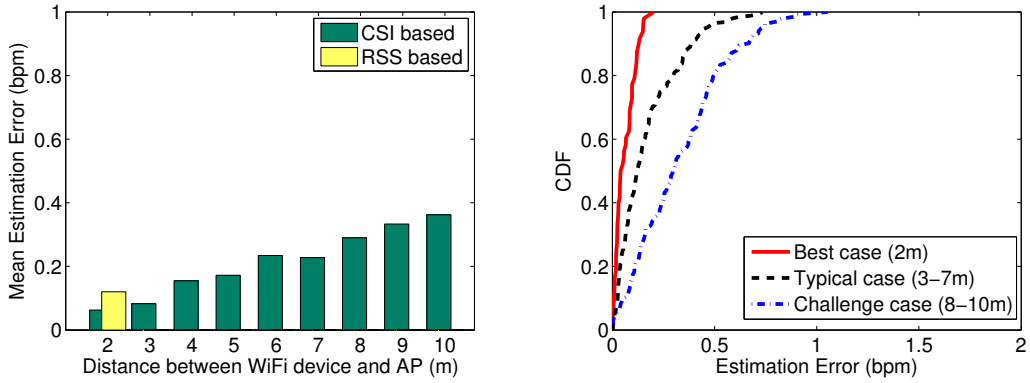
Figure 2.13: Two apartment setup.

three setups. In this setup, we utilize directional antennas (i.e., TL-ANT2406A) to enhance the reception of WiFi signals. Specifically, the distances between the AP and the WiFi device in the three setups of the bedroom-1 are $5ft$, $13ft$ and $11ft$, respectively. And the distances in the three setups of the bedroom-2 are $5ft$, $14ft$ and $12ft$, respectively. The ground truths of breathing and heart rates are monitored by the NEULOG Respiration Monitor Logger Sensor [66] and a fingertip pulse oximeter, respectively.

For the sleep posture identification experiments, we collect the CSI measurements when a participant lies in bed in the lab environment and perform four common sleep postures, which include prone, supine, curl-up and recumbent as shown in Figure 2.12(b). The participant stayed in each posture for about $40mins$. The relative position of the AP/WiFi device to the human body is same as the setup 2 in Figure 2.13 (a), and the distance of the AP and WiFi device is around $10ft$.

2.5.3 Evaluation of Breathing Rate Estimation

We evaluate the overall performance of breathing rate estimation under different scenarios including different distances between the AP and WiFi, evaluation in two real apartments and two persons in bed case.



(a) Mean estimation error.

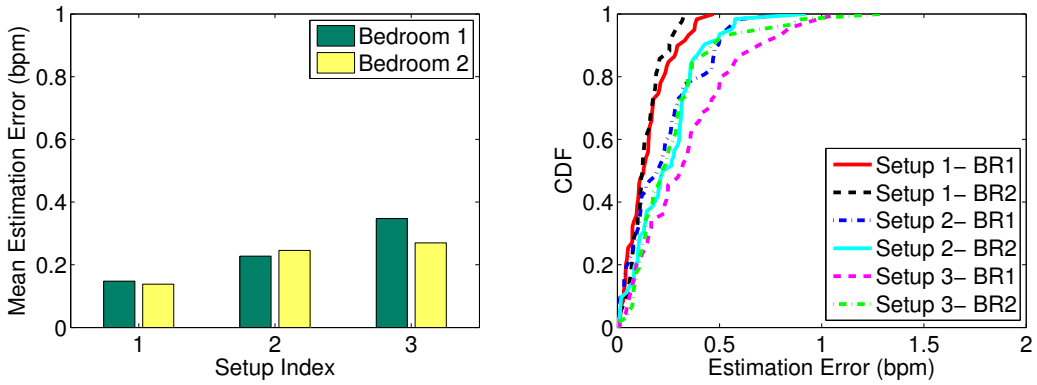
(b) CDF of estimation error.

Figure 2.14: Performance under different distances between WiFi device and AP.

Effect of Device Distance

As typical bedroom has limited space, we choose a large lab environment to study the performance of breathing rate estimation under various distances. The AP and the WiFi devices are placed at two sides of the bed (i.e., $P0$ setup in Figure 2.12(a)) with distances from 2 to 10 meters. Figure 2.14(a) presents the mean error in terms of beat per minute (bpm) of breathing rate estimation under different distances when there is a single person in bed. Overall, we observe that the mean estimation error of our breathing rate estimation is lower than $0.4bpm$, which demonstrates that our system is very accurate across different distances including very large distances such as 5 to 10 meters. In addition, shorter distance between the AP and the WiFi device results in better performance. For example, the mean error is within $0.2bpm$ when the distance is under 5 meters. This is because the received WiFi signals are stronger with shorter communication distances, providing more reliable measurements to capture the minute movements of breathing. Comparing to the result of existing work using RSS [10] which only tested with the distance of $2m$, as shown in yellow bar in Figure 2.14(a), our system provides significantly better performance (i.e., the error is reduced by about 67%).

Figure 2.14(b) depicts the Cumulative Density Function (CDF) of the breathing rate estimation error for three categories of distances between the AP and WiFi device: *best case* (i.e., $2m$), *typical case* (i.e., $3m$-$7m$ covering mid-sized bedrooms), and *challenging case* (i.e., $8m$-$10m$ covering huge-sized bedrooms). As we can see that for both *best*



(a) Mean estimation error.

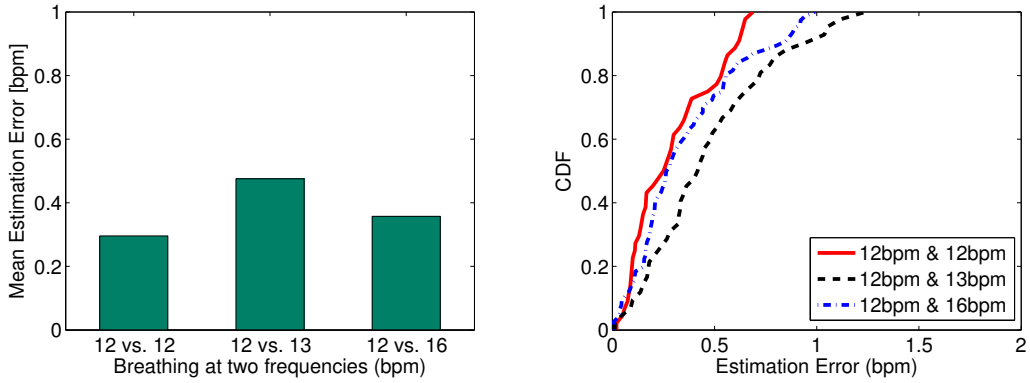
(b) CDF of estimation error.

Figure 2.15: Performance in two real apartments.

case and *typical case*, over 90% estimation errors are less than 0.4*bpm*. Even for the *challenging case*, over 80% of estimation errors are smaller than 0.5*bpm*. This suggests that our system can achieve highly accurate breathing rate estimation by using a single pair of AP and WiFi device. And it supports large distance between them.

Evaluation in Real Apartments

We next evaluate the breathing rate estimation in two different-size apartment bedrooms with different deployments of the AP and WiFi device, as shown in Figure 2.13. Figure 2.15(a) presents the mean estimation error for each setup in two bedrooms. We find that the setup 1 achieves the lowest estimation error of about 0.15*bpm* in both bedrooms due to the shortest distance between the AP and WiFi device. The estimation error of setup 2 increases as the distance between two devices increases. Still, setup 2 has the estimation error as low as 0.22*bpm* and 0.24*bpm* in bedroom 1 and bedroom 2, respectively. In addition, we observe that although setup 3 involves the obstacle (i.e., a 6-inch wall) that blocks the line-of-sight signal transmission and longer distance between the AP and WiFi device, we can still achieve less than 0.3*bpm* mean estimation error with a single pair of AP and WiFi device. Moreover, Figure 2.15(b) shows that more than 80% estimation errors are less than 0.5*bpm* for all of those three setups in two real bedrooms, indicating that our system is accurate and robust in real apartment environments. The above results show that our system provides effective



(a) Mean estimation error.

(b) CDF of estimation error.

Figure 2.16: Breathing rate estimation of two persons in bed.

breathing rate monitoring under various distances of WiFi device and AP and is robust across different environments.

Two Persons in Bed Case

We further test our system with two persons in bed using bedroom 1 setup. The AP and WiFi device are placed at two sides of the bed with the distance of $3m$. Two participants are breathing with different rates as: $\{12bpm, 12bpm\}$, $\{12bpm, 13bpm\}$ and $\{12bpm, 16bpm\}$). Figure 2.16 depicts the mean estimation error and the CDF of the breathing estimation error. We observe that the mean error is within $0.5bpm$ for all combination of different breathing rates. In addition, we find that over 90% of estimation errors are less than $1bpm$, which is comparable to that of commercial physical contact devices (e.g., zephyr[74]). Given that we only use a single pair of AP and WiFi device, such accuracy of breathing rate monitoring is very encouraging.

2.5.4 Performance of Heart Rate Estimation

Figure 2.17 illustrates the CDF of heart rate estimation error when one person is in bed using setup 1 in bedroom 1 with the AP equipped with directional antennas. We observe that about 57% of estimation errors are less than $2bpm$ and over 90% of estimation errors are less than $4bpm$. The results are very encouraging as our system achieves comparable accuracy to that of commercial sensors, e.g., Zephyr [74] and

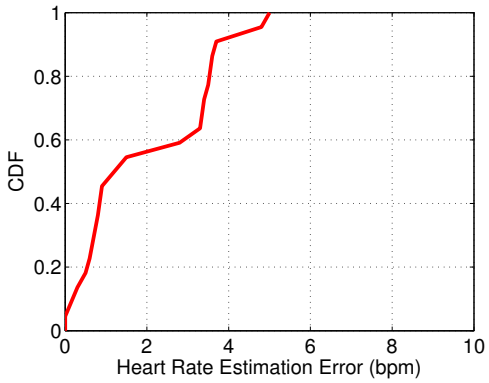


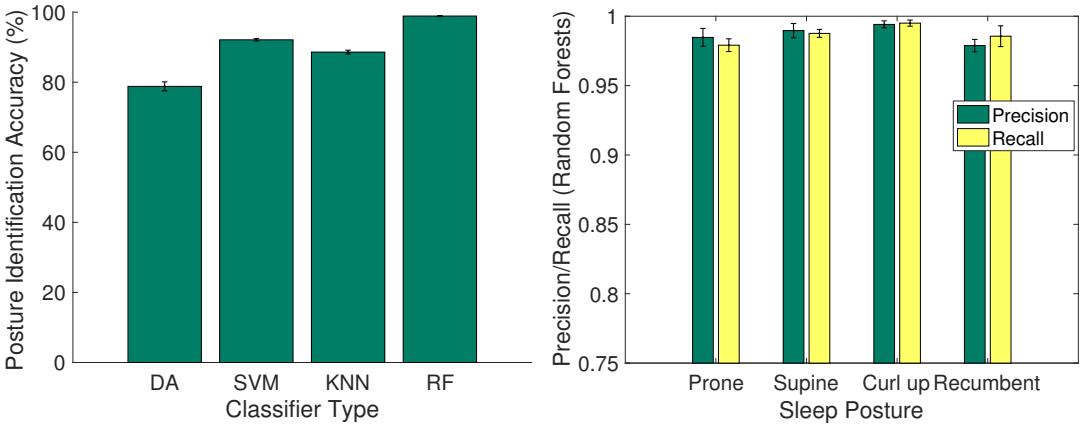
Figure 2.17: Performance of heart rate estimation.

SleepIQ [75]. Comparing with these commercial products, our system re-uses existing WiFi network without dedicated/wearable sensors or additional cost. Our system thus is able to support large-scale deployment and long-term vital signs monitoring in non-clinical settings. To the best of our knowledge, our work is the first to achieve device-free heart rate estimation leveraging off-the-shelf WiFi.

2.5.5 Performance of Sleep Posture Identification

We adopt a variety of machine learning classifiers to perform sleep posture identification, including Discriminant Analysis (DA), Support Vector Machines (SVM) with linear kernel, K-Nearest-Neighbors ($K = 5$) and Random Forests (RF). Figure 2.18(a) presents the overall accuracies of sleep posture recognition models built upon multiple classifiers. We find that all classifiers yield the accuracies over 80%. Specifically, KNN ($K = 5$), SVM and Random Forest classifiers result in the sleep posture identification accuracies over 90%, which also verifies the robustness of aforementioned feature extraction and selection techniques. We then look into the precision and recall rates of our sleep posture recognition model trained by the RF that outperforms all other classifiers, which are shown in Figure 2.18(b). We notice that even the lowest precision and recall rates across all four sleep postures are still higher than 0.95, which again demonstrates the decent accuracy achieved by our system in identifying user various sleep postures in bed.

We further examine the confusion matrix that describes the identification accuracy



(a) Posture classification accuracy. (b) The precision/recall of posture identification.

Figure 2.18: Performance of sleep posture identification.

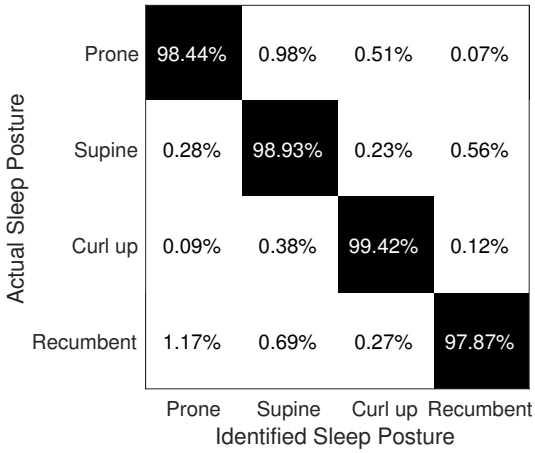
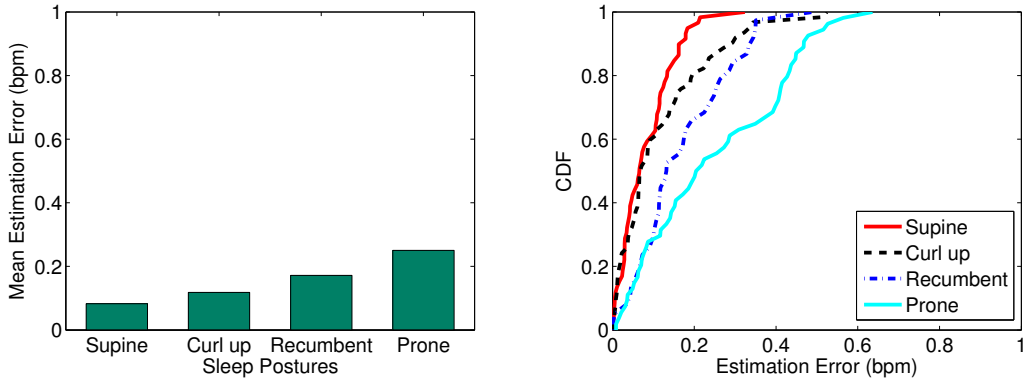


Figure 2.19: The confusion matrix of sleep posture identification.

for each of four sleep postures using the RF classifier, which is shown as Figure 2.19. Each row represents the actual user sleep posture and each column shows the posture that is predicted by our system. Each cell in this confusion matrix contains the percentage of the actual user sleep posture in the row that is classified as the postures in the column. We note that our sleep posture classification model using Random Forests can estimate each of sleep postures with accuracy over 98%. The above evaluation results collectively show that our system is able to estimate user sleep postures with high accuracy using a single pair of WiFi devices.

2.5.6 Impact of Various Factors

We also perform detailed study of breathing rate estimation under various factors.



(a) Mean estimation error.

(b) CDF of estimation error.

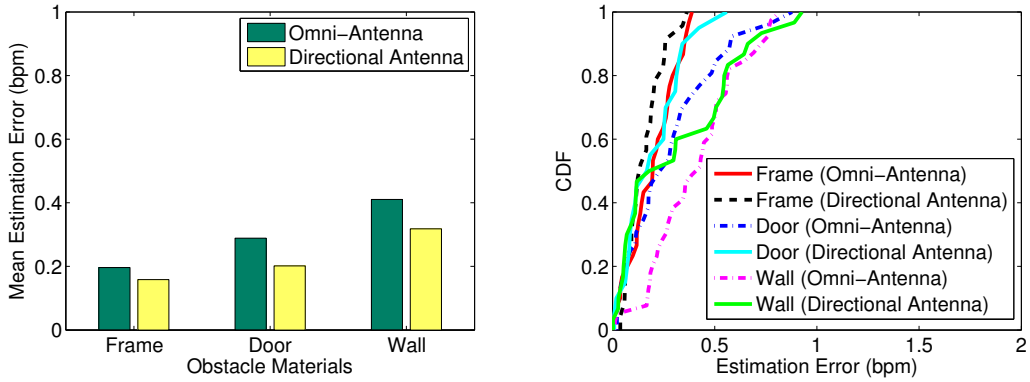
Figure 2.20: Impact of sleep postures on the breathing rate estimation.

Sleep Postures

We experiment with different sleep postures as shown in Figure 2.12(b). The AP and laptop are placed at two sides of the bed with the distance of $3m$. Figure 2.20(a) compares the mean error of breathing rate estimation resulted from different sleep postures. Overall, our system achieves less than $0.3bpm$ mean error for all sleep postures, which demonstrates the effectiveness and robustness of our system. In particular, the mean estimation errors of supine, curl up, and recumbent positions are about $0.07bpm$, $0.1bpm$ and $0.158pm$, respectively. Figure 2.20(b) shows the CDF curves of estimation error for all postures. We find that our system can obtain less than $0.2bpm$ error for more than 80% of typical sleep postures. The prone posture has the largest mean estimation error of about $0.25bpm$ for the reason that the body movements, which are caused by breathing, are mainly in the chest and belly and would be absorbed and blocked by the soft mattress. Still, our system achieves 93% of estimation errors less than $0.5bpm$ for prone posture.

Obstacles/Walls

We evaluate our system with obstacles of different materials in between of AP and WiFi device with $P0$ deployment in Figure 2.12(a). These obstacles are commonly used materials in home environments including a plastic frame of $1inch$, a solid wood door of $2inches$, and a concrete wall of $6inches$. As more and more people use directional



(a) Mean estimation error.

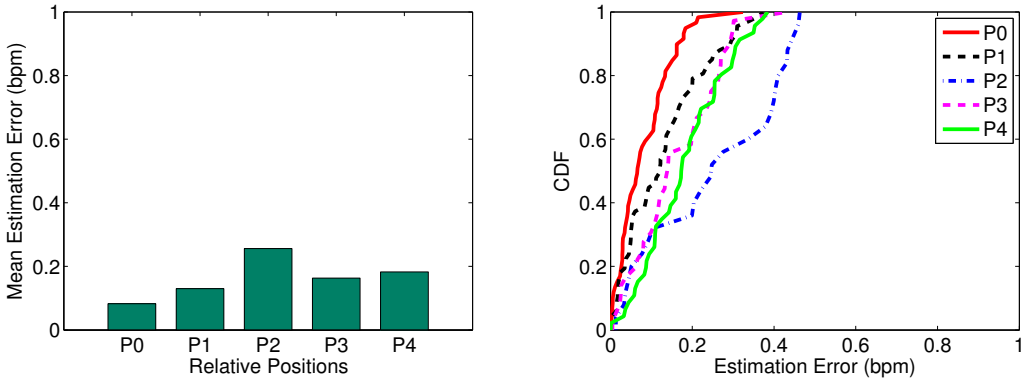
(b) CDF of estimation error.

Figure 2.21: Impact of the types of obstacles between WiFi device and AP on the breathing rate estimation.

antenna to boost the wireless signal reception in home WiFi network, we use both directional and omnidirectional antennas in the experiments. From Figure 2.21(a), we observe that the mean error is less than $0.4bpm$ for all materials. Obviously, with the concrete wall, the performance is slightly worse than that of door and plastic frame. In addition, by using the directional antenna, the mean error decreases about $0.1bpm$, indicating the directional antenna can enhance the performance of breathing rate estimation due to stronger received signals. Figure 2.21(b) shows the CDFs of estimation error. We observe that the error is always within $0.5bpm$ and $1bmp$ for the plastic frame and wall respectively. The results show that our system can work under different obstacles and the directional antenna could improve the performance. A more comprehensive study of the system performance in various environments with more obstacles and walls will be presented in our future work.

Relative Position of WiFi device and AP

Figure 2.22(a) shows the mean error of breathing rate estimation under different relative positions of Tx-Rx pair (i.e., the AP and WiFi device), as shown in Figure 2.12(a). We find that the deployment $P2$ has the largest mean error at about $0.26bpm$ among all deployments (i.e., $P0, P1, P2, P3, P4$) since the WiFi signals are partially blocked by the human body (i.e., head and feet). In addition, Figure 2.22(b) depicts the CDFs of breathing rate estimation errors. We observe that the estimation errors are all within



(a) Mean estimation error.

(b) CDF of estimation error.

Figure 2.22: Effect of relative position of WiFi device and AP.

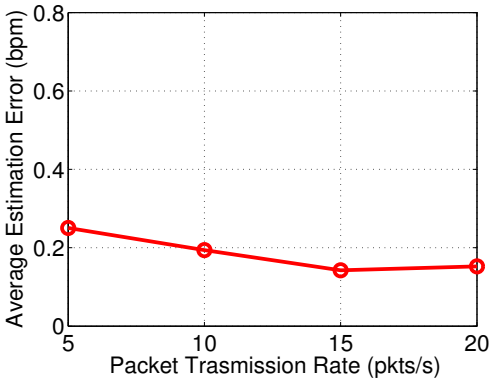


Figure 2.23: Effect of packet transmission rate.

than $0.5bpm$ even for the worst case deployment $P2$. Above results show that our system is effective under different relative positions of WiFi device-AP pair.

Packet Transmission Rate

As higher packet transmission rate results in more CSI measurements for vital signs monitoring, we are interested in how the packet rate affects the performance of our system. Furthermore, we study whether our system can work with existing WiFi beaconing packets. Figure 2.23 presents the mean breathing rate estimation error versus packet transmission rate when varying the transmission rate from $5pkt/s$ to $20pkt/s$ using the dataset from apartment experiment (i.e., Bedroom 1, setup 1). We observe that high packet transmission rate slightly improves the performance. Overall, our system is not very sensitive to packet transmission rate, given the range from $5pkt/s$ to $20pkt/s$.

Specifically, when the packet transmission rate is as low as $5pkts/s$ or $10pkts/s$, our system has about $0.24bpm$ and $0.2bpm$ mean estimation error, respectively. As the commercial access points have the beaconing of $10pkts/s$ to broadcast their SSID, our system is able to use such beacons for accurate breathing rate estimation. These results show that our system can not only work with existing AP beaconing packets but also provide accurate breathing rate monitoring with even less packet rate, such as $5pkts/s$.

2.6 Conclusion

In this paper, we show that the WiFi network could be exploited to track vital signs during sleep including breathing and heart rates using only one AP and a single WiFi device. In particular, our system exploits fine-grained channel state information from off-the-shelf WiFi devices to detect the minute movements associated with breathing and heartbeat activities. Our algorithms grounded on CSI information in both time and frequency domain have the capability to estimate the breathing rate of a single person as well as two-person in bed cases. Additionally, the existing WiFi links can also be used to track people's sleeping events (e.g., turnovers, getting up) and sleeping postures. Extensive experiments in both lab and two apartments confirm that our proposed approach using the existing WiFi network can achieve comparable or even better accuracies as compared to existing dedicated sensor based approaches. This WiFi-based approach opens up a new direction in performing device-free and low-cost vital sign monitoring during sleep in non-clinical settings.

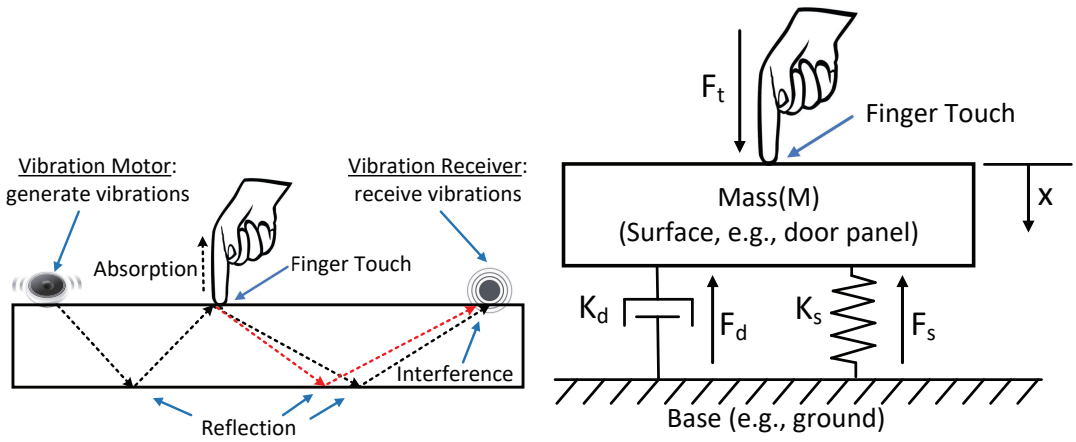
Chapter 3

Towards Finger-input Authentication on Ubiquitous Surfaces via Physical Vibration

3.1 Physical Vibration Propagation

Physical vibration is a mechanical phenomenon, which creates a mechanical wave transferring the initial energy through a medium. Similar to the transmission of wireless signals, when a vibration signal travels through a medium, it experiences attenuation along the propagation path and reflection/diffraction when the signal hits the boundary of two different media (e.g., the contacting area between a finger and a medium). Figure 3.1(a) illustrates the reflection and diffraction of a vibration signal propagating in a solid surface when a finger touches the area in between the vibration signal generator and receiver. When the vibration signal hits the contacting area of the finger, part of the signal reflects back to the surface and the rest of it propagates into the finger (i.e., absorption) and bounces back to the surface along a different propagation path. The vibration signal is affected by the touching location of the finger and traverses different paths before reaching the receiver (i.e., vibration sensor). Thus, the touching location information is embedded in the various interference effects captured at the receiver.

Furthermore, when a finger touches the surface of an object (e.g., a table), the flexibility of the object is affected not only by the touching location but also the strength of touch. A recent study [76] utilizes these properties to enable a commodity phone to recognize the force applied to its phone body and screen. To mathematically model the vibration effect on the object under an external force caused by the finger touch, we consider a spring-mass-damper system as shown in Figure 3.1(b). A free body diagram with the mass M represents the vibrating surface, while the external force F_t is caused by the finger touch. Moreover, the vertical shaft has an effective spring constant of K_s



(a) Vibration signal propagation characteristics (b) Vibration model under finger touch

Figure 3.1: Illustration of the propagation characteristics of vibration signals on a solid surface.

and a damping coefficient of K_d. When the surface has a vertical displacement of x, we have

$$F_t = K_d(\frac{d}{dt})x + K_s x + M(\frac{d}{dt})^2 x. \tag{3.1}$$

To satisfy the equilibrium condition, the vertical displacement x is dependent on the external force F_t. This indicates that the finger touching force could be captured by analyzing the received vibration signals and utilized as a biometric-associated feature in VibWrite. Note that the above analysis also works on vertical planar surface (e.g., door panel) as the equilibrium condition could be analyzed along the direction perpendicular to the surface.

In addition, Dong *et al.* [77] experimentally demonstrate that the vibration energy absorbed into the human finger-hand-arm system is different under different vibration frequencies. In our empirical study we find that the frequency response of the same user finger-press presents higher correlation than that of different users when they touch the same location on a surface. This important observation suggests that the vibration propagation properties are strongly influenced by unique human physical traits such as contacting area, touching force and etc., which can assist ubiquitous user authentication together with passcode on any surface beyond touch screens.

3.2 Approach Overview

In this section, we present the attack model and system overview of VibWrite.

3.2.1 Attack Model

We consider the following attacks that are harmful to the proposed ubiquitous authentication functionalities.

Blind Attack. An adversary randomly touches on the authentication surface equipped with the VibWrite system, hoping the random touching events can result in similar impacts to the vibration signals as the legitimate user does and passes the authentication.

Credential-aware Attack. An adversary has the prior knowledge of the legitimate user's credentials, including the PIN number, lock pattern or personal gesture, but does not possess the knowledge of the VibWrite setting details such as the grid size, gesture region, and the authentication surface.

Knowledgeable Observer Attack. An adversary is capable of both observing the legitimate user's hand movements when he is passing the authentication system via shoulder surfing or video taping as well as knowing the user's credentials and VibWrite setting details. The adversary tries to imitate the legitimate user's hand or finger movements based on his understanding of the user's credentials to pass the authentication.

Side-channel Attack. An adversary makes an effort to hack the VibWrite system directly in the hope of capturing the similar vibration signals of the legitimate user by placing a hidden vibration receiver on the authentication surface or employing a microphone in a nearby location.

3.2.2 System Overview

The basic idea underlying VibWrite is to analyze unique features from the received vibration signals to enable authentication on ubiquitous object surfaces such as entrances (e.g., apartment building or car doors) and smart home appliances (e.g., hot stove and dryer). In particular, VibWrite can be triggered when a person moves closer

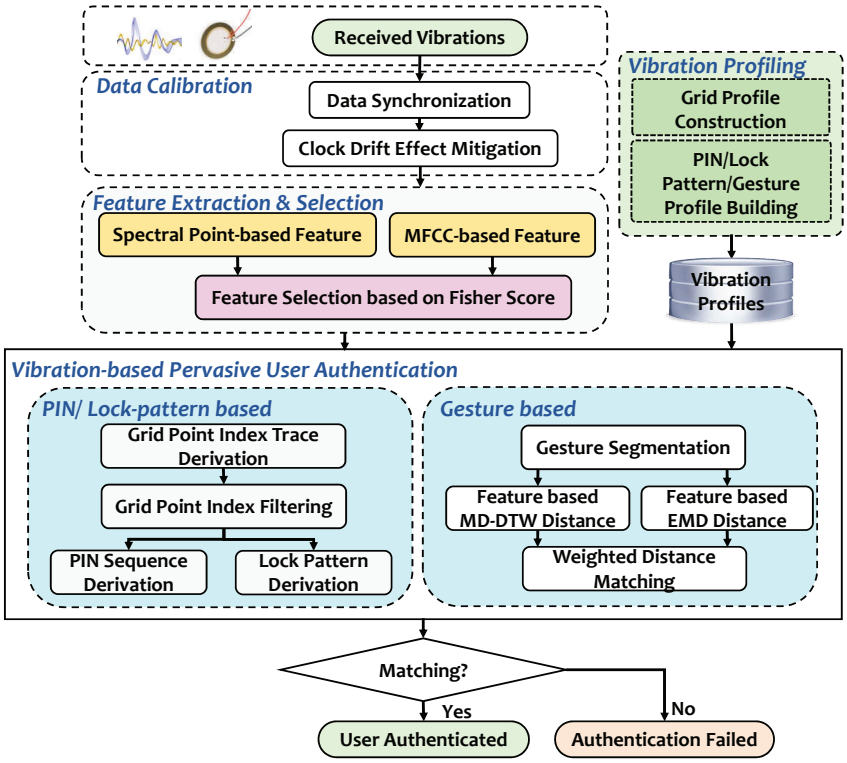


Figure 3.2: Overview of VibWrite architecture.

to the security access area (e.g., a door panel), which can be easily achieved using low power proximity sensors or motion sensors [78, 79]. As illustrated in Figure 3.2, the vibration motor then generates low annoyance vibrations and VibWrite starts taking inputs of vibrational signals from the vibration receiver. The system first performs *Data Calibration* (Section 3.3.2) including data synchronization and clock drift effect mitigation to ensure the received vibration signals always synchronized and eliminate the effects caused by the clock drift (i.e., inconsistent sampling frequency).

VibWrite then extracts and selects vibration features (Section 3.3) in the frequency domain from the synchronized vibration signals within a sliding window. We find that *Spectral Point-based Feature* (i.e., frequency amplitude of each spectral point) and *MFCC-based Feature* (Mel-frequency cepstral coefficient [80]) reflect the intrinsic physical traits embedded in the user's finger inputs. The system further performs *feature selection* based on the Fisher Score [71] on top of the Spectral Point-based and

MFCC-based features by selecting a subset of features exhibiting more discriminative power among different touching locations as well as maintaining feature consistency within each touching location.

The extracted vibration features are used by two phases in VibWrite: *profiling* and *authentication*. In both PIN number based and lock pattern based authentications, a grid is drawn on the touching surface. In the profiling phase, the features are extracted and captured while a user first enrolls in the system and presses his finger at different grid points on the touching surface. These features are labeled and saved to build the user's profile in *Grid Profile Construction*.

During the authentication phase, the received vibration signals are utilized to extract vibration features. The extracted features then serve as inputs to *Grid Point Index Trace Derivation* via a classifier based on Supporting Vector Machine (SVM) trained by the grid profiles. The classifier compares the extracted features with the stored ones in the profile to filter out the signal segments before and after the finger inputs and derive grid point trace containing finger touching inputs. The derived grid point trace would then be put into *Grid Point Index Filtering* (Section 3.4.2) to eliminate the incorrectly classified grid point indices and obtain the ones corresponding to the finger presses in the grid point index trace. Next, the filtered grid point trace would be recovered to the PIN sequence/lock pattern via *PIN Sequence Derivation* or *Lock Pattern Derivation* (Section 3.4.3). The recovered PIN number/lock pattern is then compared with the local stored PIN/lock pattern information for the final authentication.

Independently, VibWrite also enables the user to perform simple gestures (e.g., drawing a circle on the surface) for authentication without the restrictions of pressing/passing the grid points on the authentication surface. Different from the fixed grids in PIN/lock pattern based authentication, using gestures provides more flexibility for authentication. However, even for the same user, the same finger gesture could be slightly different at different authentication times due to the lack of consistency. Thus, the mechanism for gesture-based authentication in VibWrite needs to capture the intrinsic gesture behavior to deal with gesture inconsistency while preserving individual

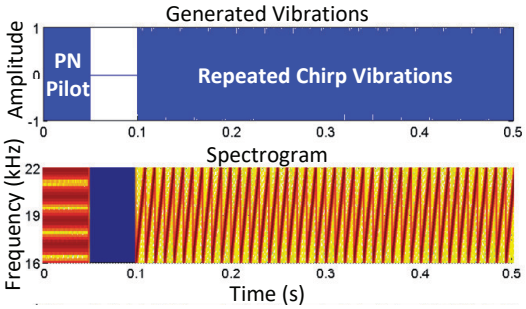


Figure 3.3: Example of generated vibrations between $16kHz$ and $22kHz$.

diversity. In particular, during the gesture-based authentication, VibWrite first identifies the signal segment containing the gesture operation via *Gesture Segmentation*. In the profiling phase, the extracted feature sequence (i.e., Spectral Point-based and MFCC-based features) from the gesture segments are saved to build the specific user's profile. To measure the similarity of generated features in the authentication phase to the gesture profiles, VibWrite addresses the gesture inconsistency problem by considering both time warped feature sequences and the distribution of the features. This is achieved by calculating both MD-DTW (Multi-Dimensional Dynamic Time Warping) Distance [81] and EMD (Earth Mover Distance) [82] of the extracted feature sequences to the profiles. The weighted distance combination in *Weighted Distance Matching* obtains the combined distance from the two techniques. Finally, VibWrite makes decision as user authenticated or access denied by checking a threshold to the calculated distances between input gestures and the stored profiles.

3.3 Vibration Signal Design and Feature Extraction & Selection

In this section, we first describe the details of vibration signal design and calibration. We then present how to extract and select unique features for the authentication process in VibWrite.

3.3.1 Vibration Signal Design

To facilitate finger-input based user authentication via physical vibration, the vibration signals used in our system need to contain a broad range of frequencies to increase

the diversity of vibration features in the frequency domain. Specifically, we generate repeated chirp vibration signals to linearly sweep frequency from $16kHz$ to $22kHz$, which are hardly audible to most human ears [83]. Additionally, such frequency range is much higher than the frequency range of ambient noise and the vibrations caused by human body (e.g., breathing and heart beating). This makes our system less possible to be interfered by these unrelated noises. Figure 3.3 illustrates an example of the generated vibration signal and its corresponding spectrogram.In particular, there is a short pseudo-noise (PN) sequence preamble played before the repeated chirp vibrations, which is used for the signal synchronization. We leave the details in Section 3.3.2. After transmitting PN pilot, with a $50ms$ pause, the vibration motor repeatedly transmits the chirp vibration signal to keep its continuous sensing capability while performing authentication. The length of each chirp vibration signal is set to T=$10ms$, which provides high time resolution to enable continuously finger-input sensing.

3.3.2 Vibration Signal Calibration

Vibration Signal Synchronization. The timing of the VibWrite's vibration motor and receiver needs to be synchronized, so that we could guarantee that each sliding window being used to extract vibration features contains the same parts of the chirp vibration signals without time delay. Therefore, they can be used for further comparison of their extracted features and capture the difference in each window when the finger touches different positions on the surface. In order to avoid the uncertainty, we add a pseudo-noise (PN) sequence preamble (i.e., 2400 samples) [84], which has ideal autocorrelation properties, at the beginning of the generated chirp vibration signals as illustrated in Figure 3.3. We then synchronize the received vibrations using cross-correlation between the PN sequence of the received vibration signal and the known generated PN sequence.

Clock Drift Effect Mitigation. When the vibration receiver senses the vibration, the analog voltage signals created by the sensor will be converted into the digitized signals via an Analog to Digital Converter (ADC). The ADC can be configured at a wide range of rates, and it is usually set to sample the analog signals at a fixed

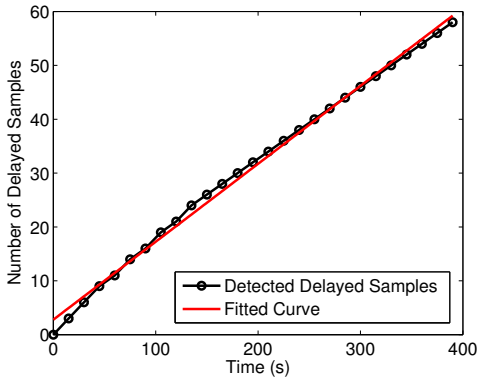


Figure 3.4: Illustration of clock drift effect mitigation.

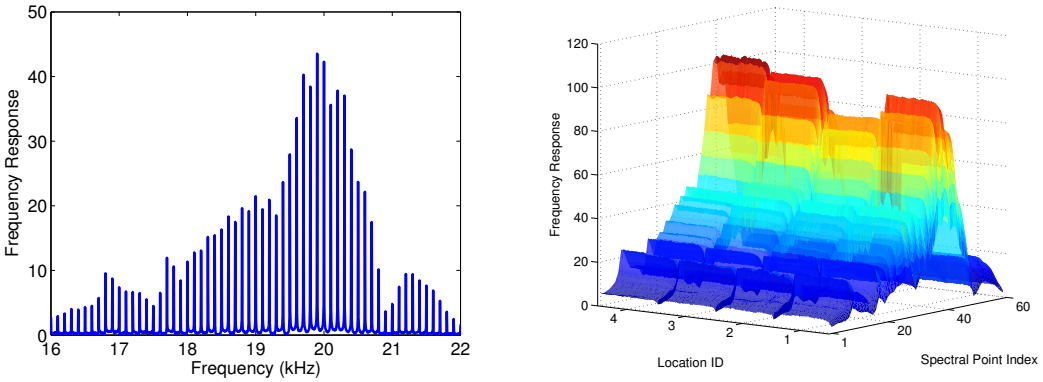
frequency driven by different application requirements. For instance, a few options (e.g., $32kHz$, $44.1kHz$ and $48kHz$) can be set in most smartphones' audio ADCs in terms of the required audio recording quality. However, we experimentally find that the sampling rate may be not a fixed value over time due to imperfect clock, and there exists a small gap between the real sampling rate and the configured sampling rate. To eliminate the effect caused by the clock drift, we estimate the sampling rate offset during a short calibration phase at the beginning. During the calibration, the vibration motor periodically sends a short vibration chirp with a fixed time interval (e.g., 2s). The time intervals between these chirps should be fixed value as well if there is no clock drift. We use cross-correlation to measure the sample delays of the received vibration chirps over time, which is illustrated in Figure 3.4. We observe that the number of the delayed samples increases linearly over time, indicating that the real sampling rate is slightly larger than the configured sampling rate but remains a relative fixed value. We then use a least-squares based approach to fit a quadratic curve to the measured delayed samples, and obtain the slop k to shift the starting point S_p of each received vibration chirp to $S_p = S_p - \lfloor kt \rfloor$, where t is the time interval between the current vibration chirp and the first received vibration chirp.

3.3.3 Spectral Point-based Feature Extraction

In order to extract unique vibration features from the received vibrations to discriminate the finger touches on different surface locations and distinguish different users touching a same surface location, we first analyze the received vibration signals in the frequency domain using a $200ms$ sliding window. Figure 3.5(a) presents an example of the Fast Fourier Transform (FFT) of a time series of the received vibration signals, ranging from $16kHz$ to $22kHz$, in a sliding window. The transmitted chirp vibration signal has fundamental frequencies that are all multiples of the frequency $1/T\ Hz$, where T is the time duration of each chirp vibration signal (e.g., $T = 0.01s$ in VibWrite). We find that the amplitudes of some designated frequency components in the signals (i.e., peak values in Figure 3.5(a)), called *spectral points*, are most sensitive to the minute changes caused by finger touching or swiping. These spectral points are more sensitive to the finger touches and could be utilized to differentiate different surface locations finger presses or finger moving along. For example, in our preliminary experiments, the vibration signals are collected when a user's finger presses at four different locations of a solid surface (i.e., wooden table) equipped with our vibration motor and receiver. We observe obvious distinguishable patterns of the frequency amplitude at these 60 spectral points (i.e., $\frac{22000-16000}{100} = 60$) between different locations, which are shown in Figure 3.5(b). Furthermore, the spectral points in the frequency domain may not be exactly spaced at $100Hz$ due to imperfect sampling module. We thus design a threshold-based strategy (i.e., minimum distance between two neighboring peaks and minimum height of each detected peak) to find peaks of the frequency response to extract each spectral point feature.

3.3.4 MFCC-based Feature Extraction

The Mel-frequency cepstral coefficient (MFCC) is widely used to represent the short-term power spectrum of acoustic or vibration signals [80] and can represent the dynamic features of the signals with both linear and nonlinear properties. While the MFCCs are able to distinguish people's sound differences in speech and voice recognition, we



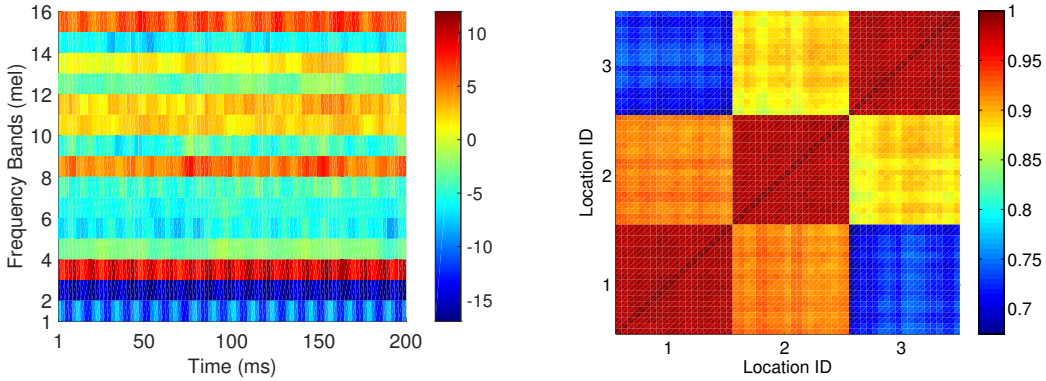
(a) Spectral points at every $100Hz$ interval

(b) Distinguishable spectral points when a finger presses 4 different locations

Figure 3.5: Illustration of the frequency response of the received vibrations in a $0.2s$ time window. And the frequency response is depicted at spectral points when a finger presses 4 different locations of a desk.

find that they can also characterize the vibration signals transmitting via the medium of a solid surface on which the user's finger touches, because the user's behavioral and physiological characteristics (e.g. touch area and pressure) and the touching position can cause different changes to the vibration propagation. We thus extract the MFCC-based features to characterize the different vibration signatures when the user touches or writes at different positions on the surface. In particular, we calculate the MFCCs of the received vibration signals in each sliding window. The number of filterbank channels is set to 32, and 16-th order cepstral coefficients are computed in each $20ms$ Hanning window, shifting $2ms$ each time.

Figure 3.6(a) shows the MFCCs extracted from the received vibration signals in a $0.2s$ sliding window when the user presses on a solid surface. We observe that the extracted MFCCs have a periodical pattern, which is caused by the cycle of the repeated vibration chirp signals. Figure 3.6(b) shows Pearson correlation coefficient [85] of the MFCC-based features when the user's finger touches at three different locations. In this experiment, twenty consecutive sliding time windows (i.e., instances) are used to extract MFCCs for each finger-touching location to compare the similarity between different finger touches. We observe that the MFCC features of the same finger-touching location present higher correlation than that of different locations, which confirms the effectiveness of utilizing the MFCC features to characterize the user's finger-touching



(a) (b)

Figure 3.6: MFCC feature illustration: (a) Example of the extracted MFCC features and (b) Pearson Correlation between MFCC features when a finger presses three different locations on a desk surface.

on the surface.

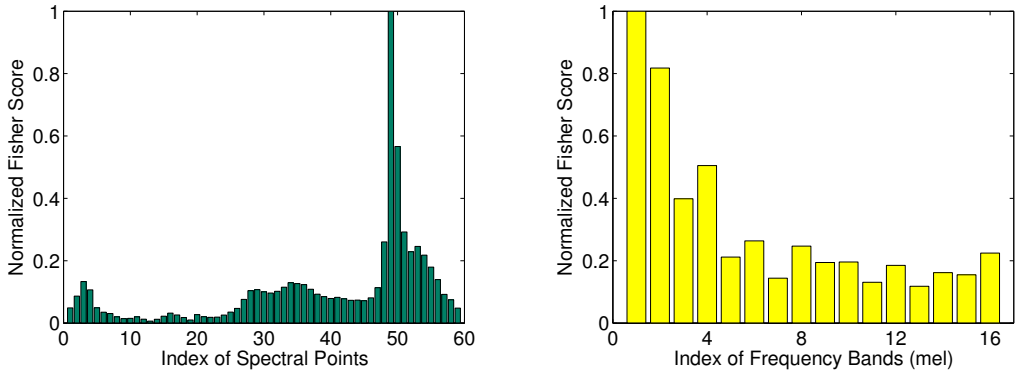
3.3.5 Feature Selection based on Fisher Score

From our experiments, we observe that not all extracted features including both spectral points and MFCC are unique enough to discriminate different touching locations and distinguish different users touching the same location. The discrimination power is dependent on the extracted features at specific frequencies or Mel-frequency bands. We therefore propose to select features based on Fisher Score [71] to find a subset of features which are more distinct between classes (i.e., touching locations per user) and consistent within a class. The fisher score of the r-th feature candidate is defined as follows:

$$F_r = \frac{\sum_{i=1}^{c} n_i(\mu_i - \mu)^2}{\sum_{i=1}^{c} n_i \delta_i^2}, \tag{3.2}$$

where n_i is the number of instances in class i. And μ_i and δ_i^2 denote the mean and variance of class i, $i = 1, ..., c$, corresponding to the r-th feature candidate. μ denotes the mean of r-th feature candidates in the whole data sets.

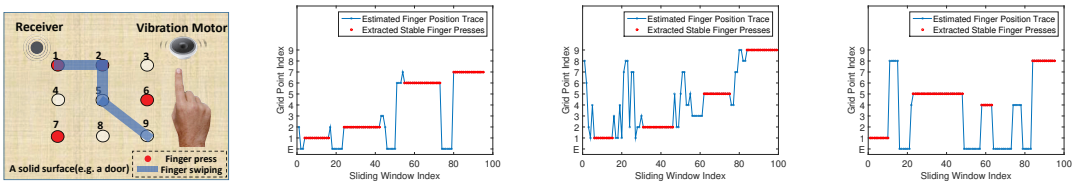
To analyze the feature difference between different frequency bands, we consider each spectral point or MFCCs at each frequency band as an individual feature candidate. Figure 3.7 shows the normalized fisher scores of both the spectral point based and MFCC based features that we use to perform user authentication. In VibWrite, we



(a) Fisher score of spectral point based features

(b) Fisher score of MFCC based features

Figure 3.7: Fisher score of the feature candidates (a) spectral point based and (b) MFCC based.



(a) User presses a PIN sequence "1267" and swipes a lock pattern "$1-2-5-9$" on a 3×3 grid

(b) Estimated finger position trace in terms of grid point index when the user enters the PIN sequence "1267"

(c) Estimated finger position trace in terms of grid point index when the user swipes the lock pattern "$1-2-5-9$"

(d) Example of an attacker entering the legitimate user's PIN sequence "1267" on the same grid of the same desk surface.

Figure 3.8: Example of PIN sequence/lock pattern derivation in sliding windows when entering a PIN sequence/lock pattern on a solid surface.

empirically choose top 30 spectral point based features, and top 8 MFCC based features which are more sensitive to the finger pressing and swiping.

3.4 Authentication Using PIN Numbers and Lock Patterns

The VibWrite system allows users to perform PIN number based authentication by touching grid points on a solid surface or conduct lock pattern based authentication by swiping finger through the grid points. Depending on the type of applications, the solid surface could be a range of options including an apartment door, a car door, an executive's office desk or a smart appliance. VibWrite first converts the received vibration signals to a time series of grid point indices, then filters out the incorrectly

classified grid point indices and finally determines the PIN sequence/lock pattern based on the derived grid point indices.

3.4.1 Deriving Grid Point Index Traces

The system takes the received vibration signals as input when the user enters PIN sequence/lock pattern. In particular, we apply a sliding window to the vibration signals and derive vibration features (e.g. spectrum-based feature and MFCC-based feature) in every sliding window. We then apply a machine learning-based grid point classifier based on the Support Vector Machine (SVM) using LIBSVM [86] to estimate the finger-press positions in terms of the grid point index for each sliding window, by leveraging the user's personal grid profile. The resulted grid point index trace is actually an estimated finger-press position trace which reflects the finger position changes among the grid point indices in the entire PIN sequence/lock pattern input duration. Note that when we derive grid point index trace, it involves user's behavior and physical characteristics. It is highly difficult for an unauthorized user to obtain correct grid point index at this step because the system needs to compare with the authorized user's profile, which integrates both PIN/Lock pattern and the user's behavior characteristics. Based on the derived grid point index trace, we can recognize the user's PIN sequence/lock pattern input and verify their identities.

Figure 3.8 shows an example of the user's PIN sequence/lock pattern based authentication on a solid surface (e.g. an apartment door) with a 3×3 grid. The predesigned grid is drawn in-between the receiver and vibration motor as shown in Figure 3.8(a), and the distance between the grid points is $3cm$. The user first builds a personal grid profile, which is discussed in Section 3.4.4. The user then presses the grid points "1267" sequentially to input a PIN sequence and swipes the finger through the grid points "1-2-5-9" to input a lock pattern as shown in Figure 3.8(a). The vibration features during the PIN sequence/ lock pattern input are extracted in each sliding window and are inputted to the SVM-based classifier. The estimated finger position trace (i.e., grid point index trace) for the PIN sequence input "1267" is shown in Figure 3.8(b). We observe that when the user presses on a number with the finger staying on the virtual

key, the consecutive same grid points corresponding to the key can be obtained, and when the user moves the finger in the air to the next key, the vibration signals are classified as "E" representing "Empty" based on the vibration profile collected when no finger presses the surface.

Figure 3.8(c) shows the estimated finger position trace of the lock pattern "1-2-5-9". We observe that when the finger swipes near a virtual key, the vibration signals will be classified to the corresponding grid point index. In particular, the consecutive same grid points can be obtained for the duration beginning from the finger moving close to, pressing on, to just swiping away from the virtual key. Thus the derived grid point index trace can reflect the user's finger positions on the grid and can be utilized to further derive the user's PIN sequence/lock pattern inputs.

3.4.2 Grid Point Index Filtering

However, the derived grid point index traces contain incorrectly classified grid point indices, which are due to the unstable vibration features caused by the varying finger touching area and force when the finger is just detaching or pressing on the surface (e.g., the noises in Figure 3.8(b)), or are because the swiping finger is far from any of the predesigned profiled virtual keys (e.g., the noisy indices in Figure 3.8(c)). These incorrectly classified grid point indices should be excluded when deriving the passcode patterns.

We develop a grid point index filter to determine the segments that have consecutive same grid point indices. Intuitively, these segments are corresponding to the time periods when the user's finger is pressing on or swiping near a grid point, which means they are more reliable results for identifying the PIN sequence/lock pattern. The grid point index filter consists of three steps: 1) calculating the difference between every two consecutive grid point indices in the trace and the firm presses will generate consecutive "0" for the differential grid point index; 2) searching for the starting and ending points of the consecutive differential grid point indices (i.e., 0s) to extract *finger-press segment*, indicating the finger positions of the firm finger presses right on or near virtual keys; 3) removing the grid point indices from the trace that are out of the finger-press segments.

The red dots in Figure 3.8(b) and Figure 3.8(c) are filtered grid point indices for the PIN sequence and lock pattern derivation, respectively.

3.4.3 PIN Sequence/Lock-pattern Derivation

Next, we further confirm each finger-press segment based on their time length and remove the incorrect finger location estimations to derive the PIN sequence/lock pattern. The intuition is that when users enter their PIN sequences, the finger press for each PIN number lasts for a certain amount of time. And when users draw their lock patterns, the duration beginning from the finger swiping close, right pressing on, to finger swiping away from each virtual key should last for an amount of time. The grid point index segments shorter than this amount of time are highly possible to be incorrect finger location estimations. We empirically determine the threshold of minimum finger-press duration (i.e., $300ms$) to remove the finger-press segments with shorter time duration. Finally, given the length of the user's PIN sequence/lock pattern, the system finds the same number of the longest finger-press segments as the valid finger-press segments and derives the PIN sequence/lock pattern by mapping the segments' grid point indices to the virtual keys.

3.4.4 Grid Profile Construction

We notice that the users can generate individually unique vibration features even by pressing at the same position of a solid surface due to the individual's different behavioral and physiological characteristics (i.e., touching area and pressure on the surface). The user's such unique vibration features can provide another level of security to our user authentication in addition to the secrecy of passcodes.Our PIN/Lock-pattern based authentication requires constructing the user's profile corresponding to every grid point, which enables successful identification of the input virtual keys during authentication. Specifically, the VibWrite system records a short time period (e.g., 1 to 5 seconds per grid point) of received vibration signals when the user presses at each grid point. The recorded vibration signals are used to derive the vibration features in sliding windows. The feature in each sliding window is labeled with corresponding grid point index. In

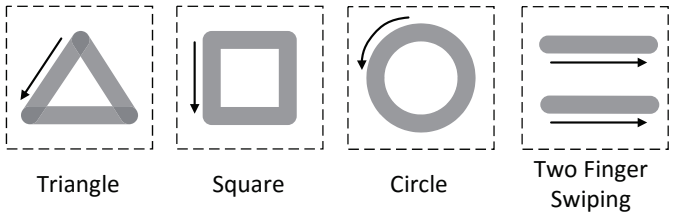


Figure 3.9: Illustration of the four pre-defined finger gestures for gesture-based authentication.

addition, we also build a profile when no finger touches the surface and label it as "E" (i.e.,"empty") to discriminate whether finger presses on the surface.

To illustrate the security provided by the user's unique vibration features in addition to the passcodes for PIN number/lock pattern based authentication. We ask an attacker to enter the legitimate user's same PIN number "1267" via VibWrite on the same grid and the same surface as shown in Figure 3.8(a). The VibWrite processes the attacker's vibration signals based on the legitimate user's grid profile and the results are shown in Figure 3.8(d). We observe that nearly all the vibration features of the attacker are incorrectly classified and thus cannot pass the authentication, which verifies the effectiveness of the individual physical characteristics contained in the user's grid profile.

3.5 Authentication Using Gestures

Different from PIN/lock pattern based authentications, using gestures provides more flexibility for authentication. In particular, VibWrite defines four simple finger gestures as shown in Figure 3.9: swiping a single finger along three patterns including a triangle, square and circle, and swiping two fingers horizontally.

3.5.1 Gesture Segmentation

To facilitate the gesture-based authentication, our system needs to first detect the occurrence of the user's gesture input from the received vibration signals and remove the vibration signals with no gestures (i.e., no touch on the surface). Specifically, VibWrite

extracts vibration features from spectral points and MFCC and then calculates vibration feature differences between the received vibration signals and those in the profile when no finger touches on the surface. The intuition is that when the user inputs a gesture, the finger swipes on the surface, causing the vibration features to differ largely from those when there is no finger touching. Figure 3.10 shows an example of calculated vibration feature differences when the user inputs square gestures on the surface for five times. For all the five gesture inputs, we observe the vibration feature difference grows higher (e.g. over 300) when the finger swipes on the surface and falls back to lower values (e.g., around 200) when the finger releases from the surface. We thus normalize the vibration feature differences and segment each gesture via a threshold.

3.5.2 Distance Calculation of Feature Sequence

User authentication using such simple gestures is much harder due to lack of unique secrecy to discriminate different users. Moreover, the speed, duration, and trajectory of the same user's gestures could be different from time to time, which causes gesture inconsistency and makes the generated vibration signals present different lengths and results in varying density of locations within the swiped pattern. In addition to feature extraction containing user's unique physical traits, we resort to two techniques to complete the authentication process in high accuracy to cope with these challenges: the Dynamic Time Warping (DTW) [81] is exploited to deal with gesture inconsistency, and the earth mover's distance (EMD) [82] technique is employed to preserve individual diversity because the feature distribution of the same user should have a higher similarity than that from different users.

Specifically, we first derive a time series of vibration features based on the vibration signals in segmented gestures using a sliding window. The DTW technique stretches and compresses required parts to allow a proper comparison between two data sequences. Therefore, it is useful to compare the vibration feature traces extracted from two segmented gestures regardless of different swiping speeds. In our system, vibration features are in a format that reports both frequency amplitude at multiple spectral

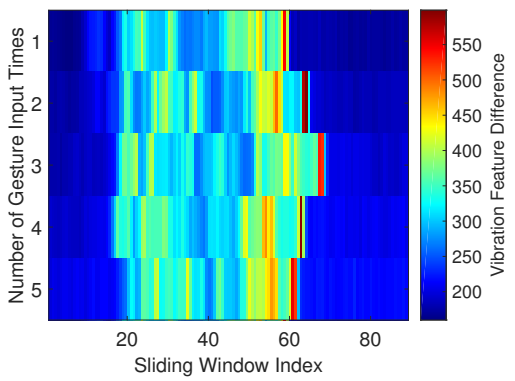


Figure 3.10: Illustration of gesture segmentation when a user inputs gestures for five times.

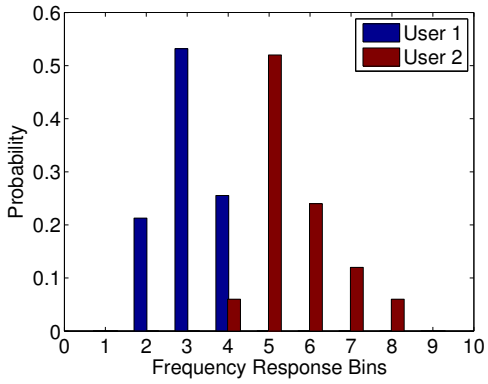


Figure 3.11: Histogram of frequency response at a spectral point for two users swiping a same gesture.

points and MFCC coefficients, which is discussed in Section 3.3. To perform multidimensional sequence alignment, our system applies Multi-Dimensional Dynamic Time Warping (MD-DTW) [81], in which the vector norm is utilized to calculate the distance matrix according to:

$$d(v_i, v'_j) = \sum_{p=1}^{P} (v_i(p) - v'_j(p))^2, \tag{3.3}$$

where $V = v_1, v_2, ..., v_T$ and V'=$v'_1, v'_2, ..., v'_T$ are two vibration feature traces for gesture discrimination, and P is the number of dimensions of the sequence data (i.e., the number of extracted features within each window). A least cost path is found through this matrix and the MD-DTW distance is the sum of the matrix elements along the path.

Besides time warped feature sequence, we find that the histogram of the spectral point based features preserve individual diversity and can be used to distinguish different users when even the same gesture is swiped. Figure 3.11 shows the feasibility study results where two users swipe their fingers following an exactly same circle gesture pattern on a desk surface. The histogram of frequency response (quantized to 10 bins) at a specific spectral point during their swiping presents distinct distributions that can clearly distinguish these two users. We thus take the advantage of the EMD-based distribution difference to preserve the individual diversity during gesture based authentication. Specifically, we normalize the EMD distance and MD-DTW distance to be integrated for final authentication. If the integrated distance to the gesture profiles

is larger than a threshold, VibWrite regards the swiped gesture as an unknown gesture and fails the authentication. Otherwise, we consider the swiped gesture is from the user whose profile results in the minimum integrated MD-DTW and EMD distance.

3.5.3 Gesture Profile Construction

Unlike grid point profile construction, VibWrite does not need to construct profiles for each grid point for the gesture-based authentication. Instead, when constructing the gesture profile for a particular user, VibWrite collects the vibration signals while the user swipes a finger following a predefined gesture. In particular, we use the sequence of the vibration features extracted from the segmented signals for building individual gesture profile. Though the profile only contains simple gestures, such profile contains the user's unique behavior and physiological characteristics and is sufficient to perform user authentication. We also build a profile with the vibration signals when there is no finger touching on the surface to determine the presence of finger touching or not for gesture segmentation.

3.6 Performance Evaluation

In this section, we first describe the experimental setup and methodology. We then present the performance of VibWrite in terms of authenticating the legitimate user and its robustness under various attacking scenarios.

3.6.1 Prototyping and Experimental Setup

We evaluate the performance of user authentication using PIN and lock patterns on a 3×3 square-shaped grid. In practice, the grid patterns could be flexibly extended as needed. The grid is drawn on a solid surface in a typical office environment. The distance between every two adjacent grid points is $3cm$. We test with two different surfaces as shown in Figure 3.12: one with the testing region resided below the vibration motor and receiver on a wooden table (e.g., the executive's desk in a company), and the other with the testing region resided in between the motor and receiver on a door panel

(e.g., an apartment door). For the user authentication using gestures, we remove the restriction of pressing/passing the grid points on the authentication surface, and aim to utilize the simplest finger gestures as shown in Figure 3.9. We want to demonstrate that even the simplest finger gestures carry the unique behavioral and physiological characteristics reflected by the physical vibrations. The gesture patterns are drawn on the table within a $6cm \times 6cm$ region between the vibration motor and receiver to guide user's swiping.

The vibration generator is implemented with a Linear Resonant Actuator (LRA) based motor, which has a wide frequency response. The frequency and amplitude of the generated vibration can be regulated by the frequency and peak-to-peak voltage of an input analog signal. The low-cost vibration receiver is implemented with a vibration receiver (i.e., piezoelectric sensor) and a low-power consumption amplifier, which can be easily plugged into the standard audio jack of any audio recording device (e.g., mobile phone) to sense vibration signals. The sampling rate of the vibration receiver is determined by the audio recording device, which is typically $48kHz$. The size of vibration motor and receiver is very small, which makes them easily to be attached to any solid surface. Compared to other authentication systems based on cameras, touch screens, or biometric readers, in VibWrite we seek to explore using low-cost sensor settings (i.e., vibration motor and receiver) for the potential of wide-deployment such as in apartment buildings, hotel rooms, smart homes, office desks, etc. Besides the vibration motor and receiver, our system needs additional supporting hardware including, but not limited to, amplifier, ADC, micro-controller and storage device to perform necessary data process, feature extraction and profile matching. With these required components, we roughly estimate the cost of an end-to-end system could be maintained around tens of dollars (e.g., \$20 $\sim$ \$50). As a comparison, some existing authentication systems (e.g., face recognition based and fingerprint based [87, 88, 89]) may usually cost hundreds of dollars.

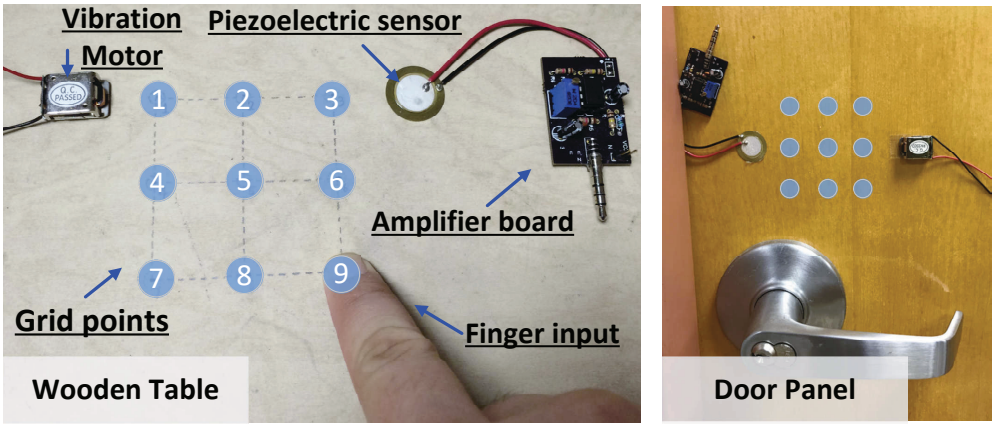


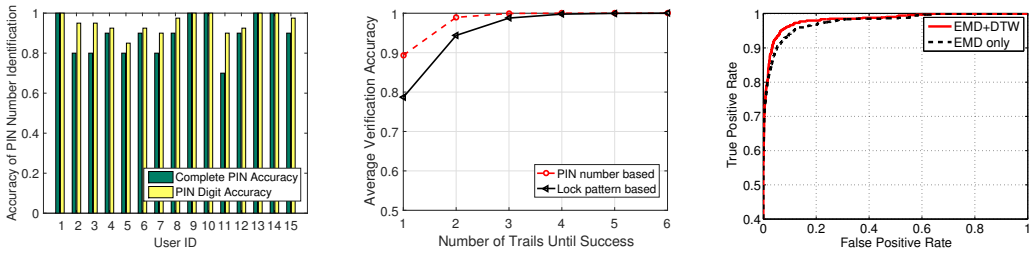
Figure 3.12: Experimental setup of VibWrite on a wooden table and door panel.

3.6.2 Evaluation Scenarios & Data Collection

Legitimate User Verification

We recruit 15 participants to evaluate the performance of VibWrite under three types of authentication. [1] Our data is collected across three-month period, and 15 participants were involved across different days. Additionally, before the data collection, we allow users to practice multiple rounds of authentication inputs on the authenticating surface to get familiar with the VibWrite system. 1) For PIN number based authentication, each user is asked to sequentially press the 9 grid points for $5s$ to create his/her grid profiles. During verification, each user presses 10 random 4-digit PIN sequences as their passcodes. 2) For lock pattern based authentication, our system uses the same grid point profiles. During testing, each user swipes his/her finger through 10 lock patterns to verify the system's authentication performance. 3) For gesture based authentication, each user chooses one of the four gestures as shown in Figure 3.9 as their preferred gestures and swipes the finger gesture 10 times. In total, we collected 450 genuine input passcodes (i.e., PIN sequences, lock patterns and gestures) for each motor/receiver placement to evaluate legitimate user access authentication. We further collected attack data to evaluate the VibWrite performance under attack scenarios.

[1] The study has been approved by our institute IRB.



(a) PIN number based authentication (b) Multiple trails until success (c) Gesture based authentication

Figure 3.13: Performance of verifying legitimate users when the testing region is below the vibration motor and receiver.

Various Attack Scenarios

We evaluate the robustness of VibWrite under various types of attack. Specifically, we choose one user as a legitimate user and the rest users as attackers to launch the attacks.

Blind Attack. The attacker randomly guesses the legitimate user's PIN, lock pattern and gesture and uses his/her finger to press and swipe on the solid surface for 10 times. In total, we collected 420 blind attack inputs.

Credential-aware Attack. The attacker gets to know the legitimate user's PIN/lock pattern/gesture. But he has not observed how the legitimate user presses his/her PIN numbers or swipes his/her lock patterns and gestures on the authentication surface. The attacker performs the same PIN/lock pattern/gesture as the legitimate user did without knowing the legitimate user's detailed behavior. Each attacker inputs the PIN/lock pattern/gesture 10 times. In total, we collected 420 inputs.

Knowledgeable Observer Attack. The attacker not only knows the legitimate user's PIN/lock pattern/gesture but also observes how the legitimate user inputs them on the authentication surface. Each attacker practices 5 times and then inputs the PIN/lock pattern/gesture 10 times, trying to pass the authentication. Again, 420 inputs are collected.

Side-channel Attack. In addition, we perform the side-channel attack by placing additional vibration receivers on the authentication surface. In particular, two receivers are employed: one is placed adjacent to the original receiver, whereas the other is placed

at the other side of the surface opposite to the original receiver.

3.6.3 Evaluation Metrics

Verification Accuracy/Attack Success Rate of PIN Number-based Authentication. The verification accuracy/attack success rate shows the percentage of correctly verified PIN numbers entered by the legitimate user or attacker respectively during the user authentication process. Specifically, it includes the complete PIN sequence verification accuracy and the PIN digit verification accuracy. The complete PIN sequence verification accuracy measures the rate of the user's input PINs being completely recognized (i.e., all numbers in the PIN sequence are correctly recognized), while the PIN digit identification accuracy shows the rate of successfully recognizing each single PIN digit.

Verification Accuracy/Attack Success Rate of Lock Patten-based Authentication. The verification accuracy/attack success rate shows the percentage of correctly verified lock patterns input by the legitimate user or attacker respectively during the user authentication phase. Similarly, it includes the complete lock pattern verification accuracy and lock pattern segment verification accuracy.

ROC Curve of Gesture-based Authentication. ROC curve is a plot of true positive rate (TPR) over false positive rate (FPR). The TPR denotes the rate of the legitimate users passing the authentication while FPR denotes the rate of the attackers successfully passing the system. Through varying the feature distance threshold in gesture-based authentication, we can achieve varied TPR and FPR and obtain ROC curves to evaluate the system performance.

3.6.4 System Performance of Verifying Legitimate Users

PIN Number-based Verification. Figure 3.13(a) shows the identification accuracy of each PIN digit and the complete PIN sequence of 15 legitimate users. Our PIN number based authentication can achieve a high verification accuracy. Specifically, the users can obtain over 95% verification accuracy of recognizing each PIN digit and the mean verification accuracy of the complete PIN sequence reaches 90%. Moreover, the

verification accuracy of each PIN digit is higher than that of PIN sequence, since the complete PIN verification accuracy result requires that all the PIN numbers in the PIN sequence are correctly identified. The results demonstrate our system is effective in verifying all the legitimate users.

Lock Pattern-based Verification. Figure 3.13(b) shows the average authentication accuracy of the lock-pattern based verification with different number of trials. Specifically, the average verification accuracy of the complete lock pattern reaches 79% and 95% with a single trial or two trials respectively, which requires all the segments of the lock pattern to be correctly identified. In addition, the accuracy of the lock pattern identification is slightly lower than that of the PIN sequence based authentication, which indicates that swiping a finger continuously on the surface generates more errors than pressing the finger separately on each grid point. The above verification results show that our VibWrite can achieve a good performance to authenticate users by lock patterns.

Gesture-based Verification. Figure 3.13(c) illustrates the effectiveness of legitimate user verification in gesture-based authentication with ROC curves. 15 legitimate users perform their preferred simple gestures (i.e., one of our four predefined gestures as shown in Figure 3.9) ten times. With only one training instance (i.e., one time swiping) for each user, we observe that given a requirement of a 90% true positive rate, we can achieve as low as a 5% false positive rate on average, which indicates only around 5% of gesture trials have gained unauthorized access. We also observe that the using both DTW and EMD techniques can provide slightly better performance than that of only using EMD technique, since it considers the similarity in both time warped feature sequences and the features' distributions. The obtained high verification accuracy and the low-training efforts demonstrate that VibWrite is capable to distinguish different users even though they perform the same simple gesture due to their distinct behavioral biometrics (i.e., finger tip size and structures).

Multiple Authentication Trials and Fall-back Strategy. Figure 3.13(b) shows the average verification rate under different number of trials. We observe that our system can achieve over 99% verification rate with both of the PIN number and lock

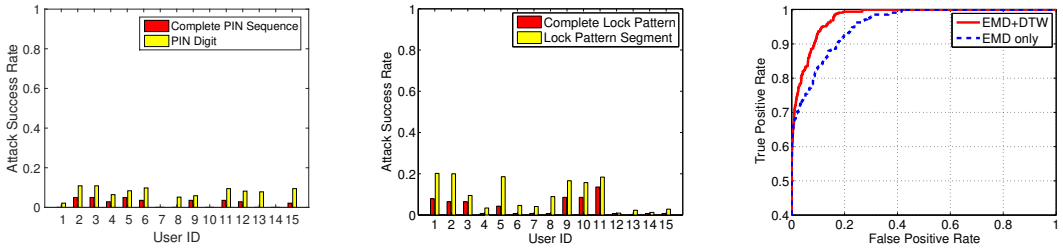
pattern inputs when users enter three trials. For the first-time user input, our system can achieve around 89% and 79% accuracies when users enter their PIN numbers or lock patterns, respectively. Additionally, our system can integrate with any fall-back strategy to let the legitimate user bypass the system, e.g., the legitimate user can always use a physical key to enter his vehicle/apartment.

3.6.5 Attacks on Legitimate User's Credentials

Under blind attacks, both our PIN number and lock pattern based authentications can achieve close to zero attack success rate. The results are intuitive because the attackers' random PIN guesses or lock pattern guesses are nearly impossible to pass the legitimate user's system within limited trials. Similarly, for gesture-based authentication, the TPR in the obtained ROC curve is close to 100% when the FPR is close to 0%, which shows that the attackers' random gestures cannot successfully access the system.

Under credential-aware attacks, our system also achieves high accuracy (i.e., close to 0% attack success rate) for all three types of authentications. Since the attackers do not possess the knowledge of the VibWrite setting details (e.g., grid size, gesture region and the authentication surface), the attackers' finger-inputs are hard to generate the similar impacts on the vibration propagation as the legitimate users do. Knowledgeable observer attack is the most extreme attack, where the attacker is capable of knowing the user's credentials and observing the legitimate user's finger inputs. Additionally, the attacker has the knowledge of the VibWrite setting details and can perform the finger inputs on the same authentication surface. Thus in the rest of this dissertation, we present the performance evaluation results of our system under this more challenging knowledgeable observer attack.

PIN Number-based Authentication. Figure 3.14(a) shows the performance of our VibWrite in PIN number based authentication under knowledgeable observer attack, where 1 of 15 users alternatively behaves as victim and other 14 users play as attackers. We find that the VibWrite system is very effective in defending against attackers even though they have the knowledge of the legitimate user's PIN and use



(a) PIN number based authentication (b) Lock pattern based authentication (c) Gesture based authentication

Figure 3.14: Performance of user authentication under knowledgeable observer attacks when the testing region is below the vibration motor and receiver.

the same VibWrite setting (e.g., grid size and authentication surface). In particular, the attackers can only break an average of around 7% single PIN digits. Furthermore, even if the attackers can successfully verify several PIN digits, it is even harder for them to break the complete PIN sequences of the legitimate user. In particular, the attackers can only achieve an average of 2% attack success rate in verifying complete PIN sequences.

Lock Pattern-based Authentication. Similarly, we ask the 15 users to alternatively play one victim and fourteen attackers, who swipe 10 lock patterns after practice based on the knowledgeable observation. Figure 3.14(b) depicts the attack success rate of lock-pattern based authentication on each legitimate user under the knowledgeable observer attack. The results show that the attackers are hard to pass the system even though they imitate the legitimate user's behavior to swipe the same lock patterns on the same grid of the same authentication surface after practice. Specifically, for the user $4, 6$-8 and 12-15, all the fourteen attackers can hardly pass the legitimate user's complete lock patterns in 10 trials though they can successfully swipe around 5% accurate segments of the lock patterns. The average attack success rates of the lock pattern segment and the complete lock pattern are around 5% and 11% respectively. Moreover, we find the performance of the lock pattern based authentication under knowledgeable observer attack is comparably good to that of the PIN number based authentication.

Gesture-based Authentication. We evaluate the performance of VibWrite in gesture-based authentication under knowledgeable observer attacks, where attackers try to mimic the legitimate user's swiping gestures. In order to test the worst case in

VibWrite, we only rely on one single training data for the legitimate user. Figure 3.14(c) shows the ROC curve, where we can achieve as low as a 3% false positive on average given a requirement of a 80% true positive rate. Even for only using EMD technique, we can still achieve as low as a 8% false positive rate on average given a requirement of a 80% true positive rate. The results indicate that, even for the most challenging knowledgeable observer attack, VibWrite is still effective in defending against attackers and successfully authenticate legitimate users in the meanwhile.

3.6.6 Side-channel Attacks

Attacks via a Vibration Receiver. One may suspect that attackers can place hidden vibration receivers on the authentication surface to recover the vibration signals and obtain the unique features of the legitimate user. In reality, the hidden receiver cannot be placed at the exact same location as the VibWrite's receiver. Thus, our $Hidden1$ and $Hidden2$ are placed at two representative locations that an adversary may choose to launch a side-channel attack. Particularly, $Hidden1$ is placed adjacent to the original receiver, whereas $Hidden2$ is placed at the other side of the authentication surface (around $3cm$ thickness) opposite to the original receiver. Figure 3.15 shows the mean and standard deviation of the Pearson Correlation coefficients [85] between the signals received by the original receiver and two hidden receivers after the designed vibration chirps are generated 20 times. We observe that $Hidden1$ and $Hidden2$ can only achieve a very low correlation coefficient less than 0.2. This indicates that the vibration signals received by hidden receivers present very different patterns comparing to that received by the original receiver even when the hidden receivers are placed very close to the original receiver, making the attacks via a hidden vibration receiver ineffective.

Attacks via a Nearby Microphone. Furthermore, a nearby microphone can record the acoustic sounds emitted by the vibration motor, however, the additional transmission path (i.e., air between the vibration motor and microphone) can largely change the vibration patterns, making it also difficult to recover the similar vibration signals received by VibWrite's vibration receiver. Additionally, a few new studies

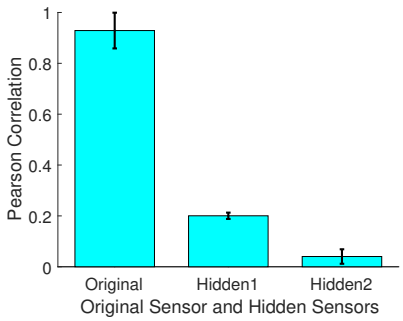


Figure 3.15: Similarity between the vibrations received by VibWrite's original receiver and hidden receivers.

demonstrate that physical vibrations can be recovered to a certain extent by using wireless signals [90] and high-speed cameras [91]. However, these solutions can only recover relatively low-quality audio/vibrational signals due to the limits of the hardware sensing ability in both vibration amplitude and frequency. Thus, they are mainly used for eavesdropping human speech sounds whose frequency typically falls below $1KHz$.

3.6.7 Impact of Training Data Size

PIN Number/Lock Pattern based Verification. Our system can achieve around 90% accuracy in identifying each PIN digit/lock-pattern segment with the grid point training time over 0.4 seconds while the identification of complete PIN sequences or complete lock pattern achieve over 80% accuracy with the grid point training time over 0.6 seconds as shown in Figure 3.16. Moreover, the PIN sequence/lock pattern based authentication can achieve higher accuracy with longer training time and the accuracy reaches stable when the training size is over around 2 seconds.

Gesture-based Verification. From the results as shown in Figure 3.13(c) and Figure 3.14(c), we observe that our gesture-based verification can obtain very high authentication accuracy with the training profile only containing one single gesture training instance. The results also indicate that our gesture-based authentication system could work with a very small training data size.

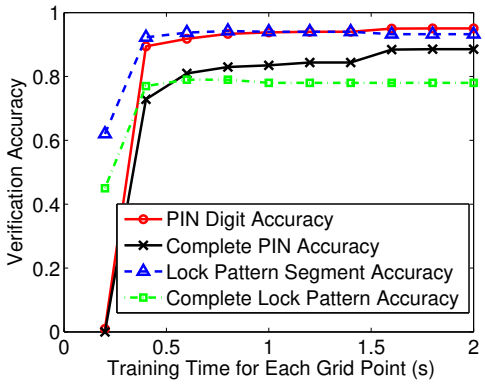


Figure 3.16: Performance of both PIN number based and lock pattern based authentications with different grid point training time periods.

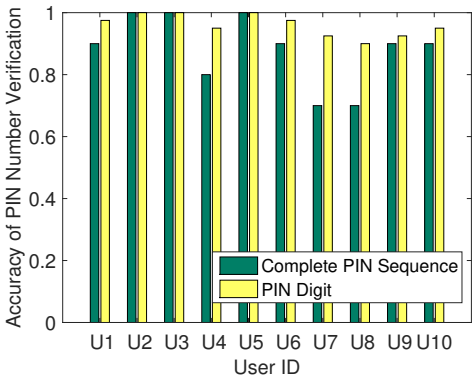


Figure 3.17: Performance of PIN number based authentication in verifying legitimate user when the testing region is on a door panel.

3.6.8 Impact of Surface and Vibration Motor/Receiver Placement

We change the positions of the vibration motor and the piezoelectric sensor to the center of each side and evaluate the PIN sequence verification accuracy on the grid of the door panel surface. Ten users are first asked to construct their individual grid profile and then input their PIN sequences with this new experimental setup for verification. The results in Figure 3.17 show that our PIN number based authentication can achieve comparably high verification accuracy for this setup. In particular, the accuracies of verifying the complete PIN sequence and PIN digit are 88% and 94% respectively. The similar results can also be observed for lock pattern based and gesture based authentication. Thus our system is robust for different vibration generator/receiver placements.

3.7 Discussion

Serving as a concrete starting point of vibration-based authentication system, VibWrite is a low-cost and easy-to-deploy solution that has a high potential to work at various places such as apartment buildings, hotel rooms, smart homes, etc. We admit that the current system is still not ready for the industrial deployment in terms of its authentication/false-accept rates, thus a large space is left for us to further improve the

system. In this section, we introduce a few limitations of the current VibWrite system and the potential for future improvements.

Accuracy, and Further Improvement. The current system achieves around 89% and 79% authentication rates with a single trial when users enter their PIN numbers or lock patterns, respectively. The accuracy number is comparable to a few recent low-cost authentication/verification solutions (e.g., [92, 93, 94, 95]), which use either gait patterns captured by existing Wi-Fi/smartphone or passive sensing of embedded sensors on smartphones. Specifically, the gait pattern based solution could achieve around 80% detection rate of unauthorized users when leveraging accelerometers on smartphones [92] and 79% user recognition accuracy when using off-the-shelf Wi-Fi [93]. Multi-sensor (i.e., gyroscope, magnetometer and accelerometer) based smartphone authentication can achieve around 70% and 90% accuracy in the studies [95] and [94], respectively. However, the current VibWrite system is still far from practical deployment as a legitimate user may need to try a few times to pass the system. To improve the system performance, we target to explore the following aspects in our future work including deploying multiple sensor pairs, refining the hardware, and improving the authentication algorithms. Specifically, more than one pair of vibration transmitters and receivers can be employed to help increase the dimension of the surface sensing features, which can better represent each individual's behavioral and physiological characteristics. In addition, empirically we noticed that the uniqueness of the features is affected by the stableness of the hardware components as the weak analog signals extracted by the piezoelectric sensor can be easily distorted when passing through electronic components (e.g., amplifier and ADC). We thus could build a higher standard hardware signal processing component (e.g., ultra-low-noise signal amplifier) to enhance the system. Meanwhile, the improvement of the vibration motor in terms of its power level, stableness and frequency response could become another venue to explore.

Coping with Additional Physical Attacks. In addition to the side channel attacks via a hidden vibration receiver or a nearby microphone, other types of physical attacks might be launched when the system is deployed in practice. We discuss a couple of representative ones below and show how VibWrite could be extended in coping up

with such attacks. Given that the proposed system is highly dependent on the attached surface, such surface dependency might be employed by an adversary to launch a denial-of-service (DoS) attack (e.g., adhering tiny objects or a hidden vibration motor to the surface) to prevent the legitimate user from passing the system. To combat the DoS attack, VibWrite can develop a simple mechanism to perform the surface sanity check periodically by comparing the received vibration signals with the *empty surface* training profile. If the surface dissimilarity is detected, the authentication surface will be examined. The most extreme case is when an adversary gets access to the cable connecting the vibration motor/sensor and cut it to make the system not function at all. On one hand, to deal with such a physical attack, the vibration motor and receiver could be placed at the opposite side of the authenticating surface hidden from the users and even placed inside some enclosed cases hard to access without authorization. On the other hand, the adversary does not gain much benefit in this attack as he still cannot pass the authentication system. We leave the detailed study of these adversarial cases as an avenue for our future work.

System Maintenance. As a starting point, our system is evaluated in a relatively stable indoor environment. However, in practical deployment, there are many environmental factors that need to be taken into consideration and may affect the system performance. For instance, if the surface (e.g., car door panel) is exposed to an outdoor environment, the surface's vibration response may be changed across different days affected by temperature, humidity, wind, wetness, dirt, etc. Additionally, the temporary presence of additional objects placed on the surface (e.g., a book placed on the desk) could alter the received vibrations slightly different from the trained one. The noticeable effect caused by these factors might be reduced through further filtering or directional sensing techniques. More robust machine learning methods grounded on deep learning [96] can also be built in our future work to deal with various environmental-related elements. In addition, future work should continue the evaluation with more/diverse population samples, longer time periods and more influential factors to improve the system robustness.

3.8 Conclusion

In this work, we propose VibWrite, which implements the idea of low-cost low-power tangible user authentication beyond touch screens to any solid surface to support smart access applications (e.g., apartment entrances, vehicle doors, or smart appliances). Utilizing low-cost physical vibration, VibWrite performs ubiquitous user authentication via finger-input by integrating passcode, behavioral and physiological characteristics, and surface dependency together to provide enhanced security. VibWrite is built upon a vibration-based touch sensing technique that enables touching and writing on any solid surface through analyzing unique vibration signal features (e.g., frequency response and cepstral coefficient) in the frequency domain. It is easy to deploy and flexibly provides users with three independent forms of secrets (including PIN number, lock pattern, and simple gesture) to gain security access by developing new techniques of virtual grid point derivation, featured-based dynamic time warping (DTW) and distribution analysis based on earth mover's distance (EMD). We perform extensive experiments with participants input their passcodes by using three forms of secrets. We also study the robustness of Vibwrite under various attacks trying to impersonate the legitimate user or launching side-channel attacks to hack the VibWrite system directly. Our results indicate that VibWrite is resilient to side-channel attacks. And it can verify legitimate user with high accuracy under minimum training efforts while successfully deny the access requests from unauthorized users with a low false positive rate.

Chapter 4

Snooping Keystrokes with mm-level Audio Ranging on a Single Phone

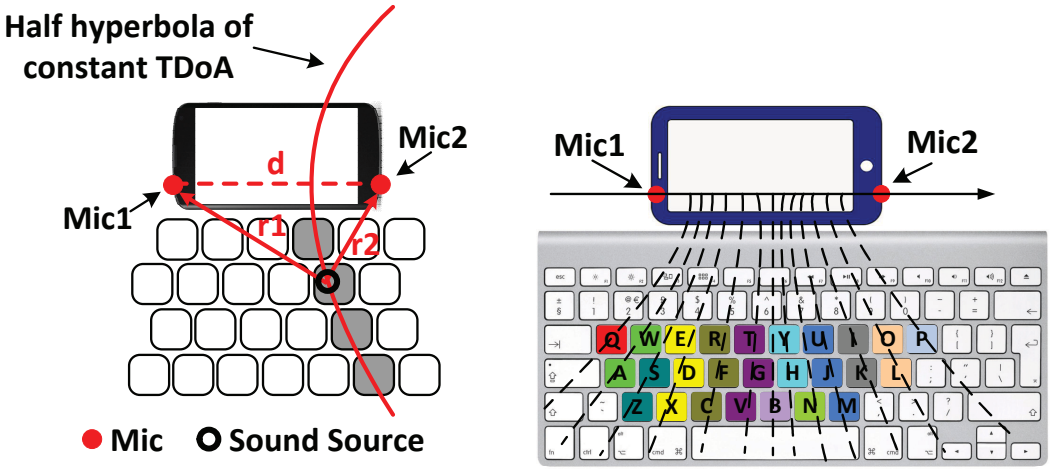
4.1 Attack Model & Limits of TDoA with a Single Phone

In this section, we introduce the attack scenarios and the rationale for their selection. We also analyze key factors affecting the accuracy of keystroke snooping when using a single phone and define basic concepts.

4.1.1 Attack Model

We consider a scenario where an adversary seeks to identify each entered character in a sequence of keystrokes from the acoustic signal generated by depressing a key ("typing sound"). We assume that adversaries have the access to stereo audio recordings from a single mobile device (e.g., smartphone or tablet) that is placed close to a victim's keyboard. Two representative scenarios where this is plausible are: (1) the adversary inconspicuously leaves a prepared recording device next to the victim's keyboard, perhaps in a confined setting such as library seats where physical proximity is not suspicious; (2) the adversary gains software access to the microphones of the victim's phone, perhaps through a malicious app, and waits until the victim places the phone next to a keyboard. We note that it is not uncommon for users to place their phone on the desk while working on a laptop keyboard. Moreover, tablets and large phones are frequently used with external Bluetooth keyboards, where the device is placed directly behind the keyboard. We believe that this latter scenario is particularly likely and use it as the primary example in this work.

We do not assume any particular structure in the typed information. This means



(a) Geometrical TDoA measured by a pair of microphones.

(b) Theoretical key groups and corresponding half hyperbolas.

Figure 4.1: Illustration of the geometrical TDoA on a single-phone and theoretical key groups.

that adversary seeks to identify not only text input matching a known linguistic model but also seeks to identify random input strings such as strong passwords. We also explicitly do not assume that the adversary has access to labeled training data (i.e., audio recordings for each key, where it is known which key was pressed). Such training data is specific to a particular phone-keyboard combination, the exact placement, and the exact acoustic environment. It would therefore be challenging to obtain in many adversarial settings.

4.1.2 Basic Concepts of Single Phone TDoA Localization

The selection of the attack model is rooted in an understanding of the fundamental limits of acoustic localization. To avoid the need for labeled training data we disregard fingerprinting techniques and focus on time-of-flight measurements, which are attractive given the relatively low propagation speeds of audio signals. Since we do not know when a particular keystroke sound is emitted, we rely on measuring the time-difference-of-arrival (TDoA) of this sound across the two microphones of the device.

A TDoA measurement reveals information about the direction of the incoming

sound. Determining an exact position of origin for the sound normally requires triangulation, that is at least two direction measurements from different locations. In the keyboard snooping scenario, however, there is a discrete and relatively small set of candidate positions from which the sound can emanate: the center of each key. If the relative phone position and keyboard geometry is known, it is therefore possible to locate the sound even with a single TDoA measurement by finding the best match between the direction estimate and the expected direction for each key.

This process, however, requires mm-level accuracy, which is an order of magnitude beyond the cm-level accuracies that have been previously demonstrated in audio localization. Operating at this level of accuracy involves estimating precise hyberbolas instead of coarse direction estimates. Consider our primary scenario as illustrated in Figure 4.1(a). Let's denote the distance between two microphones as d, and the distance between the sound source (i.e., the keystroke made on the keyboard) to that of two microphones as $r1$ and $r2$, respectively. Suppose Δt is the TDoA measured at two microphones, the derived distance difference Δr from the pressed key to two microphones can be represented as:

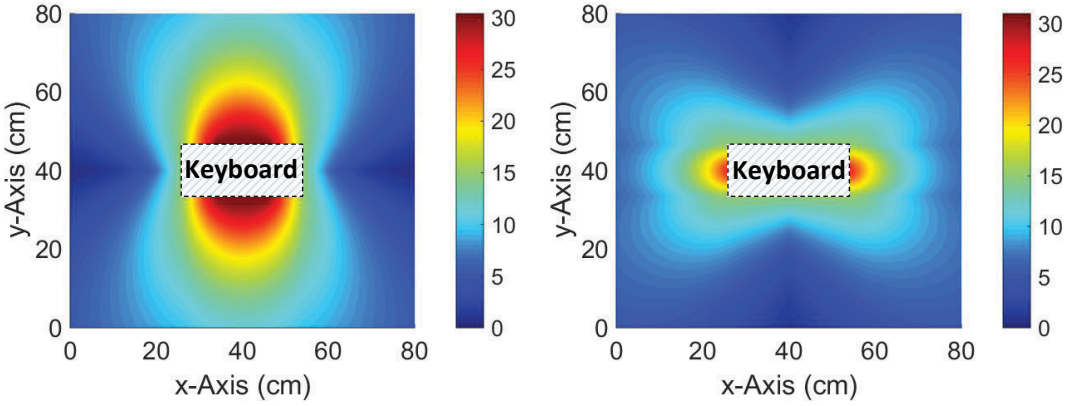
$$\Delta r = r1 - r2 = \Delta t \cdot s_0, \tag{4.1}$$

where s_0 is the velocity of sound. All possible locations that satisfy Δr lie on a hyperbola as illustrated by the red curve in Figure 4.1(a). This hyperbola typically crosses several keys on the keyboard as indicated by the darker keys in the figure (and one key may also be crossed by more than one hyperbola). Narrowing this to a single key therefore involves determining the closest key center to this hyperbola. As can be seen in the figure, shifting the hyperbola by only a few mm would bring it closer to the center of a different key. For this reason, the process require mm-level precision and accuracy.

4.1.3 Factors Affecting Accuracy

The achievable accuracy and precision with TDoA measurements depend on several key factors.

Sampling Rate. When recording the keystroke sound, the sound is digitized by



(a) Horizontally placed. (b) Vertically placed.

Figure 4.2: Good phone locations for keystroke snooping. Warmer color indicates locations from which a higher range of TDOA values can be observed over different keys.

an Analog to Digital Converter (ADC) with a fixed sampling rate before it becomes accessible to applications. This therefore limits the resolution with which the time difference of arrival can be measured by application software. While signal processing techniques exist that promise sub-sample accuracy, this time resolution serves as a useful guideline.

Current state-of-the-art audio hardware on mobile devices supports up to $192kHz$ sampling rate but drivers and or operating system usually still limit this to $48kHz$. At a speed of sound of 343m/s, this results in a resolution for the distance difference Δr of $\approx 1.8mm$ and $\approx 7.15mm$ for the two sampling rates, respectively.

Distance between Two Mics. The number of distinguishable hyberbolas also depends on the range of possible TDoA measurements. The range is bounded by the distance between two microphones on the phone. It can be calculated based on the triangle inequality theorem. As can be inferred from Figure 4.1(a), the range of Δr is $[-d, +d]$. The TDoA value Δt then falls into the range of $[-d/s_0, +d/s_0]$, which corresponds to the number of distinguishable hyberbolas N at the sample level as expressed below:

$$N = \lceil \frac{2d \cdot f_s}{s_0} \rceil. \tag{4.2}$$

For example, the distance between the two microphones of the Samsung Galaxy Note 3

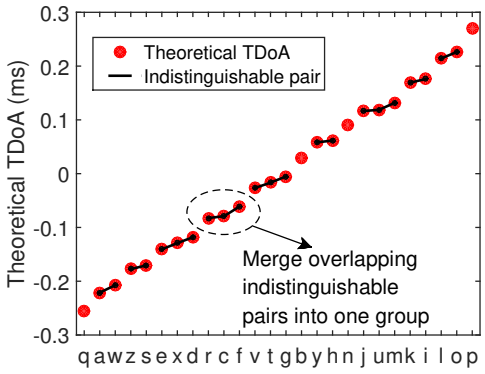


Figure 4.3: Illustration of sorted TDoAs for theoretical key groups construction.

smartphone is $d = 15.3cm$. With a sampling rate $f_s = 48kHz$, this yields 42 hyperbolas that can be discriminated at the sample level.

Placement of the Mobile Device. In practice, only a subset of these N hyperbolas may actually cross the keyboard and be useful for distinguishing keystrokes. The size of this subset depends on the size of the keyboard and the relative location of the recording device. Figure 4.2 shows the size of this subset depending on the phone position around the keyboard, for two different phone orientations. Warm colors indicate good phone positions relative to the keyboard. Horizontal phone orientation means that a line connecting the two microphones would be parallel to the long side of the keyboard (as also the case in Figure 4.1(a)). Vertical orientation means that the phone is rotated 90 degrees to the left, so that the line is parallel to the short side of the keyboard. This analysis assumes a sampling rate of $48kHz$, a keyboard size of $28cm \times 13cm$ (as for the Apple wireless keyboard MC184LL/A) and microphone distance of $d = 15.3cm$ (as on the Samsung Galaxy Note 3). The results show that a vertically placed phone on the side needs to be in very close proximity but a horizontally placed phone behind the keyboard offers a bit more flexibility. Such placements are consistent, however, with the attack scenarios that we have identified earlier. In our primary scenario (e.g., a Samsung Galaxy Note 3 is placed behind an apple keyboard as illustrated in Figure 4.1(a)), in particular, this leaves us with 31 hyperbolas crossing the 26 alphabetic keys.

4.1.4 Theoretical TDoA and Key Groups

Given known keyboard geometry and phone placement, keystroke snooping can be simplified as a matching process between measured TDoA values and expected TDoA values for each key. By solving for Δt in Equation (4.1), it is possible to compute an expected TDoA value for each key, which we also refer to as the theoretical TDoA value for a key.

In addition to the limiting measurement factors discussed earlier, measurements will be affected by noise. This further limits the distinguishability of keys and leads us to introduce the notion of theoretical key groups, which are groups of keys whose expected TDoA values are so close that we would expect them to be difficult to distinguish. For instance, the 26 alphabetic keys in our primary scenario are grouped into 13 theoretical key groups, each illustrated through a separate color in Figure 4.1(b).

These key groups are established as follows. We first sort the keys based on their theoretical TDoAs. We then link any pair of keys whose difference in theoretical TDoA is less than a threshold τ. Based on our experiments with different keyboards and sampling rates, we empirically determine τ as $\frac{1}{480}ms$ (which corresponds to $1, 2, 4$ TDoA samples corresponding to $48kHz$, $96kHz$ and $192kHz$). Each connected set of keys is then considered as one theoretical key group, as illustrated in Figure 4.3.

We will explain how to use these concepts and how to achieve accuracy below the level of a theoretical key group next.

4.2 System Overview

To accurately recover keystrokes using a single mobile device, we design an approach that leverages TDoA measurements and fine-grained acoustic signatures of keystrokes. In this section, we discuss the challenges and architecture of our system design.

4.2.1 Challenges

To achieve the goal of accurately recognizing keystrokes by utilizing a single mobile device without relying on training and contextual information, the design and implementation of our system involve a number of challenges:

Sensing with Single Mobile Device. Using one single mobile device to recover keystrokes is challenging as most commercial mobile devices only support stereo recording with two microphones, while general acoustic TDoA localization approaches require at least three microphones to create multiple half-hyperbolas to locate a sound source. Moreover, the distance between two microphones in a phone is highly constrained, which limits the range of possible TDoA values. Although some mobile devices have three microphones, for example iPhone 5s and Samsung Galaxy Note 3, neither Apple nor Google provides API to record 3-channel audio with three microphones. Therefore, our system must be designed in a way that it can accurately identify keystrokes based on the stereo recording of two microphones.

Imperfect measurement of TDoA. Different from some recent TDoA localization studies [32, 34, 33, 35] that utilize customized acoustic signals, such as a high frequency narrow band signal, our work locates more challenging sound sources, i.e., keystrokes, which cannot be controlled and contain rich frequency components. Meanwhile, the range of possible TDoA values is limited by the distance between two microphones and is affected by the placement of the mobile device. Also, the sampling frequency limits the resolution with which the time difference of arrival can be measured. Moreover, the measured TDoA may also be affected by multipath effects and environmental noises. These factors result in imperfect measurements of TDoA which make it hard to uniquely locate each keystroke.

Training-free Keystroke Recognition. Without the cooperation of the targeted user, developing training-free keystroke recognition is critical when performing keystroke snooping, especially when an adversary seeks to derive the user's sensitive typing information. Our system aims to recognize keystrokes without training efforts that involve target users (e.g., requiring the target user to type each key repeatedly to

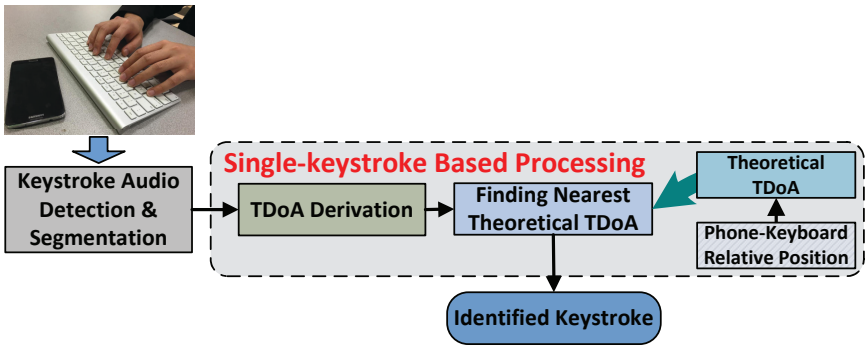


Figure 4.4: System architecture: single-keystroke based processing.

label the data beforehand).

Recovering Keystrokes without Linguistic Model. Users may type not only sentences following English language constraints (e.g., emails and articles), but also random letters or numbers (e.g., passwords and credit card numbers). Our developed method should have the ability to recover sensitive information consisting of random combination of letters and numbers. This requires our system to recognize keystrokes without relying on linguistic models or dictionaries.

4.2.2 System Architecture

The basic idea of our system is to perform keystroke snooping leveraging the dual-microphone on a single smartphone through studying the fine-grained acoustic signatures inherent from key typing sounds. In particular, we consider two processing approaches, namely *Single-keystroke Based Processing* and *Set-keystroke Based Processing*. These two approaches seek to cover various practical scenarios that have different requirements on the accuracy and response time. The *Single-keystroke Based Processing* can be applied to even a small set of recovered keystrokes, since it can process each keystroke individually. The *Set-keystroke Based Processing* exploits a larger set of keystroke samples to improve the recognition accuracy. It reduces the effect of imperfect measurement of TDoA by combining multiple keystroke samples from the same key. It can identify strokes of the same key by extracting the acoustic cepstral features of keystrokes as well as by using coarse TDoA matching.

In our proposed system, we assume the relative position of the mobile device to

the keyboard is known. This information can be obtained if an adversary intentionally places the mobile device close to the keyboard, or when the external keyboard is attached to the mobile device (e.g., Microsoft Surface). There are also other means to obtain such information. For example, the adversary may take a picture of the setting of the keyboard and mobile device. The adversary may also estimate the setting using a bunch of collected keystrokes of multiple keys. It is important to note that such process does not need the participation of the target user as in the traditional training phase. Additionally, we discuss how to derive such information when the relative position the mobile device is unknown in Section 4.4.3.

Single-keystroke Based Processing. A quick way to recover each individual keystroke is to leverage the theoretical calculated TDoAs based on the relative position between the mobile device and keyboard. The Single-keystroke Based Processing method compares the measured TDoA derived from the input keystroke to the computed theoretical values and determine which key has been pressed. The main steps of this method are depicted in Figure 4.4 and described as follows: For each captured keystroke sound, this method first perform *Keystroke Audio Detection & Segmentation* to extract the audio signals corresponding to the press and release phases of the keystroke. It then derives the TDoA based on the extracted keystroke acoustic signal using signal processing techniques. Next, it determines which key has been pressed by finding the top-w keys that have the theoretical TDoAs closest to that of the input keystroke (i.e., *Finding the Nearest Theoretical TDoA*).

Set-keystroke Based Processing. This method aims to reduce the impact of the imperfect TDoA measurement by examining a set of input keystrokes and study the statistics of the fine-grained acoustic features in addition to pure TDoA computation. Figure 4.5 illustrates the steps of the Set-keystroke Processing approach. This method first takes as input a set of different keystroke sounds recorded by a nearby mobile device. It then extracts the audio signals corresponding to the keystrokes and derives the TDoAs. Next, it performs *Pre-grouping of Keystrokes Using TDoA* to categorize the input keystrokes to multiple key groups based on the pre-calculated theoretical key groups

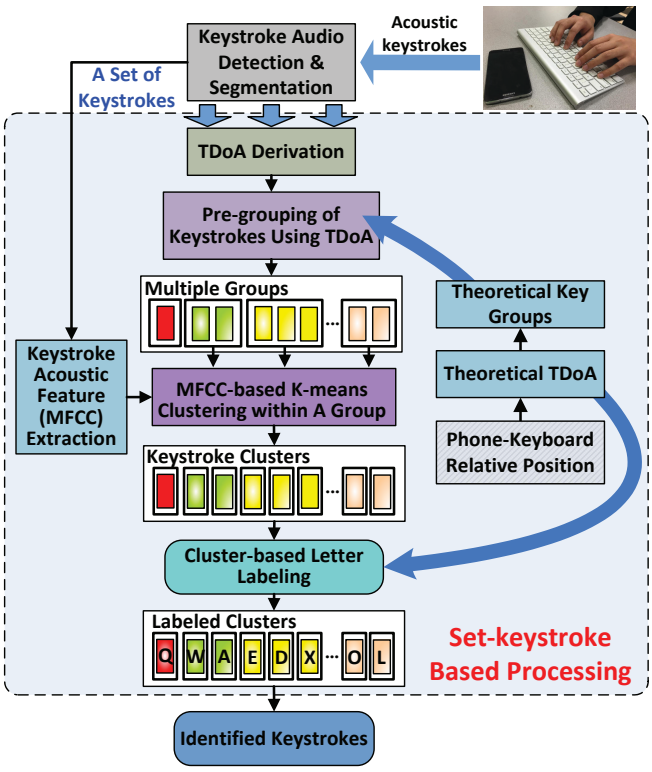


Figure 4.5: System architecture: set-keystroke based processing.

in Section 4.1. To overcome the limited accuracy, this method then extracts the cepstral features (e.g., Mel Frequency Cepstral Coefficients (MFCCs)) from the keystroke sounds through the *Keystroke Acoustic Features (MFCCs) Extraction* component. The MFCC features are utilized to further cluster the keystrokes in the same key group so that each cluster only contains strokes of the same key. This allows calculation of mean TDoA values over several strokes of the key, which helps reduce measurement noise. Finally, the system performs key labeling of each cluster to recover each keystroke by examining the distance difference between the mean TDoA of each cluster to that of the theoretical TDoAs. We discuss the details of this approach in Section 4.3.

4.3 Set-keystroke Based Processing

4.3.1 Pre-grouping Keystrokes Into Theoretical Key Groups

After a set of keystrokes are collected, the Set-keystroke Based Processing approach first obtains the TDoA of each keystroke based on the techniques described in Section 4.4. It then utilizes these derived TDoA values to assign each keystroke into a theoretical key group based on the discussion in Section 4.1. We denote each key as K_i^n, where i is the key ID and n is the corresponding theoretical key group ID (e.g., K_1^1 is the key "Q" and K_{19}^{12} is the key "L"). We further denote the theoretical TDoAs of keys with the theoretical key group ID n as $D_n = \{\Delta t_i^n\}$, where i is the key ID and Δt_i^n is the theoretical TDoA of the key K_i^n. We then put each input keystroke into one of the theoretical key group by comparing its measured TDoA Δt with the theoretical TDoAs Δt_i^n. The input keystroke will be assigned to the theoretical key group n, if the differential TDoA between Δt and Δt_i^n is the minimum as shown below:

$$G = \arg\min_n |\Delta t - \forall \Delta t_i^n \in D_n|. \tag{4.3}$$

At the end, each input keystroke is assigned with a theoretical key group ID.

4.3.2 MFCC Based K-means Clustering

We next explore the acoustic features of keystroke sound to further separate the keystrokes within the same key group.

MFCC Feature Extraction. In our experiments, we find that the Mel-Frequency Cepstral Coefficients (MFCCs) [80, 97] of keystroke sounds capture acoustic signatures of different keys within the same theoretical key group. MFCC utilizes the magnitude of the Fourier Transform of the time-domain speech frames to analyze acoustic signals. The rationale of using MFCC to distinguish different keystrokes in the same theoretical key group is that physical uniqueness of each key component results in slightly different keystroke sounds for different keys. In addition, the keystroke sounds of keys at different locations experience different multipath effects. In particular, we extract the MFCC features from the entire duration of a keystroke sound. The number of filterbank

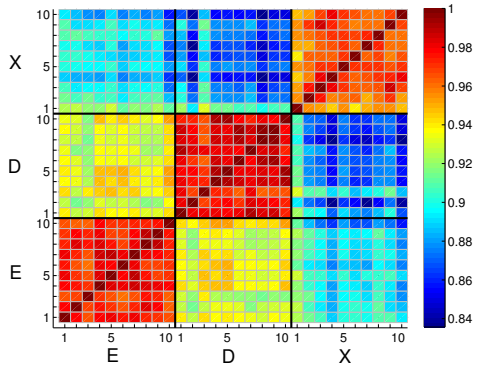


Figure 4.6: Pearson Correlation between MFCC features of three keys within a group: same key shows higher correlation, while different keys present lower correlation.

channels is set to 32, and 16^{th}-order cepstral coefficients are computed in each $10ms$ Hanning window, shifting $2.5ms$ each time. To exclude the frequency range of ambient noise, we only consider acoustic signal from $400Hz$ to $14kHz$ for MFCC extraction.

To illustrate the effectiveness of using the MFCC features to distinguish different keystrokes within a key group, we repeatedly type "E","D", and "X" keys (which are within the same theoretical key group) 10 times respectively and examine the correlation between the MFCC features extracted from the keystrokes. Figure 4.6 shows the Pearson correlation coefficient [85] between any two MFCC features derived from those keystrokes. We observe that the MFCC features of the same key present higher correlation than that of different keys. It thus appears promising to use MFCC features to distinguish keystrokes within a group. We note only one channel of the keystroke sound is used to extract MFCC features. If dual-microphones have different characteristics, we could combine parallel features to improve the clustering performance [98].

In-group K-means based Clustering. To reduce the effect of the imperfect measurement of TDoA and minimize the impact of environmental noise, we further cluster keystrokes within a group into different clusters based on the MFCC features (if the corresponding theoretical key group contains multiple keys). In particular, we use the *cityblock* distance to measure the distance between MFCC features of different keystrokes using K-means clustering [99]. In order to obtain the optimal clustering results, we minimize the variances of the MFCC features of keystrokes in each cluster

by satisfying:

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{n=1}^{N_k} |m_i^k - \mu_k|^2, \tag{4.4}$$

where N_k represents the number of keystrokes in the k^{th} cluster, m_n^k denotes the MFCC features of the n^{th} keystroke in the k^{th} cluster, and μ_k is the mean value of the MFCC features in the k^{th} cluster.

4.3.3 Cluster Based Letter Labeling

Finally we label each cluster. We leverage the statistical information of TDoAs in each cluster to determine which key the cluster belongs to. In practice, the TDoA measured from multiple keystrokes for the same key may have slightly different values as the touch point may change slightly each time. In our experiments, we find that keystroke sounds emitted from different keys within group have different distributions of TDoAs which result in slightly different mean TDoA. Moreover, the mean TDoA of the keystroke sounds emitted by the same key is very close to the corresponding theoretical TDoA. Thus, we compare the mean values of the measured TDoAs of each cluster to the theoretical TDoAs. The keystrokes in the cluster will be labeled as the key whose theoretical TDoA has the minimum distance to the mean TDoA of that cluster.

4.4 Implementation

4.4.1 Keystroke Segmentation

A typical keystroke acoustic signal can be divided into three parts [36, 38]: *touch peak*, *hit peak* and *release peak*. These peaks correspond to touch, hit and release the key respectively. Figure 4.7 shows an example of these three peaks from two different keyboards (i.e., Apple wireless keyboard and Razer Black Widow keyboard).

In order to extract the acoustic sound of a keystroke, we first examine the energy levels of the acoustic signal to determine the starting point of the keystroke sound [38, 40, 37]. Particularly, we calculate the energy levels of a keystroke sound by accumulating

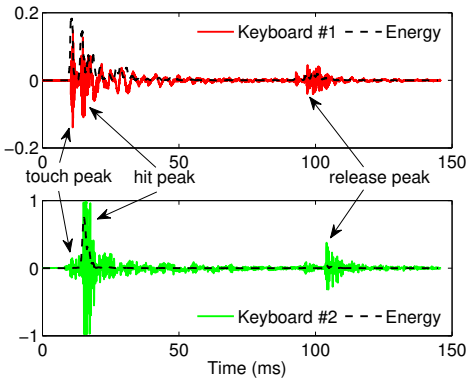


Figure 4.7: Keystroke acoustic signals emitted from two keyboards and corresponding short-time energy, keyboard-1 (Apple wireless keyboard) and keyboard-2 (Razer Blackwidow keyboard).

the square of the signal amplitude in a sliding time window as shown below:

$$A(t) = \sum_{n=t}^{t+W} r^2(n), \tag{4.5}$$

where W is the length of the time window and $r(n)$ is the amplitude of the sound signal within the time window. We empirically determine the length of the sliding window as $W = 2ms$ (i.e., 96 samples with $48kHz$ sampling rate). Figure 4.7 illustrates the energy levels of the keystroke signals from two keyboards.

We identify the starting point of the keystroke sound when the energy level exceeds a threshold. An empirical threshold of 0.05 is used in our work to determine the starting point p_s. We find the length of keystrokes is typically about 100 milliseconds. We thus extract the keystroke sound as the acoustic signal between $[p_s - 5ms, p_s + 100ms]$. Note that our system uses the entire keystroke sound to generate MFCC features, whereas the system only uses about first $20ms$ segment roughly corresponding to *touch peak* and *hit peak* to calculate the TDoA as these two peaks result in more accurate TDoA estimation than using the whole keystroke sound.

4.4.2 TDoA Derivation

Once we have input keystroke segment, we could find out how many delayed samples between two digital audio signals recorded at two microphones at a mobile device to obtain the time delay between two microphones when receiving keystroke sound. Suppose the acoustic signal of a keystroke is recorded at the two microphones as $r_1(n)$ and

$r_2(n)$ with length L respectively, where $n = 1, \ldots, L$. We use cross-correlation between the two recorded signals to derive the TDoA. Cross-correlation is a standard signal processing technique to measure the similarity between two signals and is calculated as:

$$cc(d) = \frac{\sum_n \left[(r_1(n) - \mu_{r_1}) \cdot (r_2(n-d) - \mu_{r_2})\right]}{\sqrt{\sum_n (r_1(n) - \mu_{r_1})^2 \cdot \sum_n (r_2(n-d) - \mu_{r_2})^2}}, \tag{4.6}$$

where μ_{r_1} and μ_{r_2} are the means of the corresponding signals. $cc(d)$ provides the similarity between $r_1(n)$ and shifted (delayed) copies of $r_2(n-d)$. If the Equation 4.6 is computed for all delays $d = 0, 1, \ldots, L-1$ then it results in a cross correlation series of the original $r_1(n)$ or $r_2(n)$. Then, the TDoA Δt (i.e., time delay) between $r_1(n)$ and $r_2(n)$ can be obtained as:

$$\Delta t = \frac{1}{f_s} \cdot (\arg\max_d cc(d) - L). \tag{4.7}$$

4.4.3 Relative Position Estimation

The relative position of the phone to the keyboard is needed in our method to calculate the ground truth of TDoA (i.e., theoretical TDoA values). This information could be obtained if the adversary intentionally places the phone at a pre-identified location or if the phone/tablet is used in a tablet stand. If the adversary plants a malware into the victim's smartphone, such information could be inferred based on the keyboard layout and the measured TDoA of keystrokes.

Keyboard layout can be obtained offline as long as the keyboard model is known. The keyboard model could be detected by capturing Bluetooth identifiers or through manual visual identification. With the keyboard layout, we then can define the coordinates of keys. For the sake of simplicity, we assume there are m keys K_i with known coordinate or location loc_i, where $i = 1, 2, ..., m$. Given the measured TDoA of a collection of keystrokes, we have the TDoA value of each key to that of two microphones with certain measurement error. We assume the measured TODA of each key to that of two microphones is $\hat{\Delta t}_i$, and the sorted one is $\hat{\Delta t}_j$, with $j = 1, 2, ..., m$.

With the above information, we can estimate the locations of two microphones M_1 and M_2, with the constraint $\|M_1 - M_2\| = d$, where d is the known distance between two

microphones. With the arbitrary locations of microphones and known location of the key loc_i, we can calculate the theoretical TDoA of each key to that of two microphones as Δt_i, with the sorted one Δt_j. The optimal location of the two microphones thus can be estimated as follow:

$$\arg\min_{M_1,M_2} \sum_{j=1}^{j=m} \|\hat{\Delta t_j} - \Delta t_j\|, \tag{4.8}$$

where $\|\hat{\Delta t_j} - \Delta t_j\|$ represents squared distance between $\hat{\Delta t_j}$ and Δt_j.

Note that we may not get the measured TDoA of each key in practice. Even so we can calculate the location of two microphones as long as we obtain several measured TDoA values. Moreover, the measured TDoA of different keystrokes for the same key may be different slightly. Empirical study shows that the difference is small (i.e., about one or two samples). We could then group similar TDoA values together and use the averaged value to represent the TDoA of one key.

4.5 System Evaluation

In this section, we first present the experimental methodology, and then evaluate the performance of both set-keystroke based and single-keystroke based approaches. We also discuss the impact of multipath propagation on the keystroke snooping.

4.5.1 Experimental Methodology

Keyboard & Phone

Keyboard. Although we do not study the sound intensity level of each key, we evaluate our system with three different kinds of keyboards (i.e., an Apple wireless keyboard MC184LL/A, a Microsoft surface keyboard and a mechanical keyboard Razer Black Widow Ultimate) that produce different keystroke sound intensity levels. In particular, the keystroke sound from the mechanical keyboard is much louder than that from the Apple keyboard. And the keystroke sound from the Microsoft surface keyboard is the weakest. These keyboards have different designs and dimensions resulting in different layout of keyboards and different characteristics of keystroke sounds. Specifically, the Apple wireless keyboard and Microsoft surface keyboard have comparable dimension

(i.e., $\sim 28mm \times 13mm$), whereas Razer keyboard is much larger (i.e., $\sim 47mm \times 17mm$). Moreover, the depth of key caps on the Apple and Microsoft keyboards (i.e., $\sim 2mm$) is much smaller than that of the Razer keyboard (i.e., $\sim 6mm$).

Mobile Phone. In our experiments, we utilize the Samsung Galaxy Note 3 as the mobile device to launch attacks. The operating system of the phone is Android 4.4.2. Although the Samsung Galaxy Note 3 has equipped with three microphones on the top, bottom and right bottom, the microphone on the right bottom edge is only used for noise cancelation. We thus use the top and bottom microphones to record the keystroke sound. The distance between these two microphones is about $15.3cm$.

Sampling Rate

The audio chipset on smartphone (i.e., Samsung Galaxy Note 3) is Qualcomm Snapdragon 800 MSM8974 [100], which supports $24bit$ nominal quantization at $192kHz$ sampling rate. Although the Android 4.4.2 system only supports up to $48kHz$ sampling rate, Smartphone Operating Systems are increasingly supporting higher sampling rate, for example recently released Android 5.0 claims it could support up to $96kHz$ sampling rate [101]. We thus envision that the software restriction on the sampling rate will be loosed and the smartphone could use $192kHz$ for audio recording in a near future. In the evaluation, we study the impact of different sampling rates on the performance of keystroke snooping. We simulate the high sampling rate (i.g., $96kHz$ and $192kHz$) by utilizing a pair of omni-directional microphones connected to a laptop through a USB adapter (i.e., Diamond Tube). We place the two microphones $15.3cm$ apart from each other to simulate the Samsung Galaxy Note 3 with $96kHz$ and $192kHz$ sampling rates.

Placement

We concentrate on the primary usage scenario, where the mobile device is placed behind the keyboard. We further study two more placement scenarios, where the mobile device is typically placed by the user when using the keyboards: in front of the keyboard and left side of the keyboard. These three placements are shown in Figure 4.8.

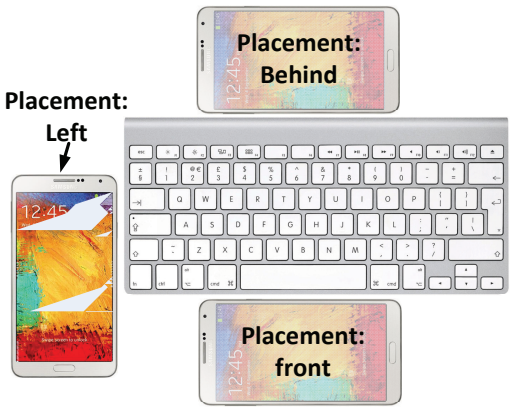


Figure 4.8: Three typical placements of the phone to the keyboard in the experiments.

Data Collection

We focus on experiment on the 26 alphabet letters, but our method also applies to the whole keyboard. Three participants are involved to randomly type the 26 keys *a*-*z* on keyboards in typical office environments (i.e., two laboratory rooms with ambient noise (e.g., HVAC noise)). For each experimental setup (i.e., a specific type of keyboard, placement, and sampling rate), 520 keystrokes are collected. In total there are $3,640$ keystrokes from three participants for our experimental evaluation.

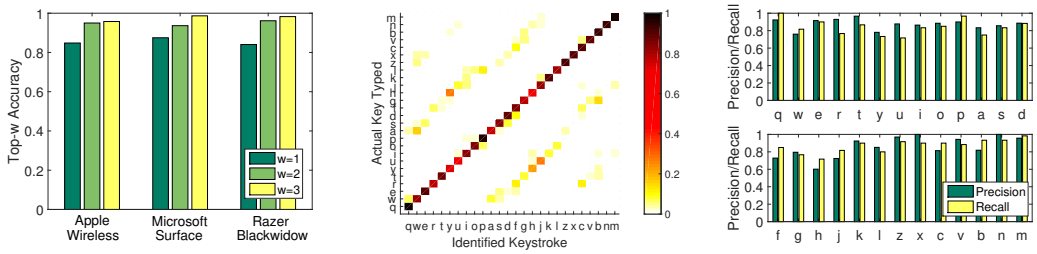
Metrics

We use the following three metrics to evaluate the performance of keystroke snooping:

Precision. Given N_k keystrokes of a key k, precision of recognizing the key k is defined as $P_k = \frac{N_k^T}{N_k^T + M_k^F}$, where N_k^T is number of keystrokes correctly recognized as the key k, M_k^F is the number of keystrokes corresponding to other keys mistakenly recognized as the key k.

Recall. Recall of the key k is defined as the percentage of the keystrokes that are correctly recognized as the key k among all keystrokes of the key k, which is $R_k = \frac{N_k^T}{N_k}$.

Top-w Accuracy. Given w identified key candidates, we want to know whether the pressed key is among these w candidates. The *top-w* accuracy measures overall performance of the keystroke recognition. Assuming the number of keys on keyboard



(a) Top-w accuracy with different keyboards.

(b) Confusion matrix.

(c) Precision and recall.

Figure 4.9: Performance of set-keystroke based processing using three keyboards and off-the-shelf phone (48kHz).

is K, the *top-w* accuracy is defined as $A = \frac{\sum_{k=1}^{K} P_k^{T,w}}{\sum_{k=1}^{K} N_k}$, where P_k^T is the number of the keystrokes that are correctly identified as one of the keys among the *top-w* candidates, N_k is the total number of keystrokes for key k.

4.5.2 Performance of Set-keystroke Based Processing

Overall Performance

We evaluate the overall performance of the set-keystroke based processing with the primary attack scenario (i.e, the phone is placed behind the keyboard). The sampling rate is set as $48kHz$. Figure 4.9(a) shows the overall accuracy for keystroke identification with three different keyboards. We find that the phone can capture different levels of keystroke sound intensity from all three keyboards when the mobile phone is placed close to the keyboard. We observe that all three keyboards have comparable high accuracies. In particular, the top-1 accuracy is about 85.5%, whereas the top-2 and top-3 accuracy increase to 94.9% and 97.6%, respectively. These results show that our training-free and context-free approach provides sufficient accuracy to snoop on passwords composed of random characters.

Figure 4.9(b) plots the confusion matrix for the keystroke recognition after combining the results from three keyboards. We find that there are only few keystrokes are mistakenly identified as incorrect keys. These mistakenly recognized keystrokes usually correspond to the neighboring keys that have the same TDoA value. For example, a few keystrokes of the key w are mistakenly recognized as the key a which is crossed

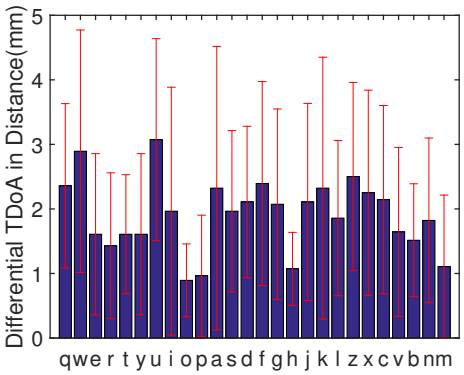


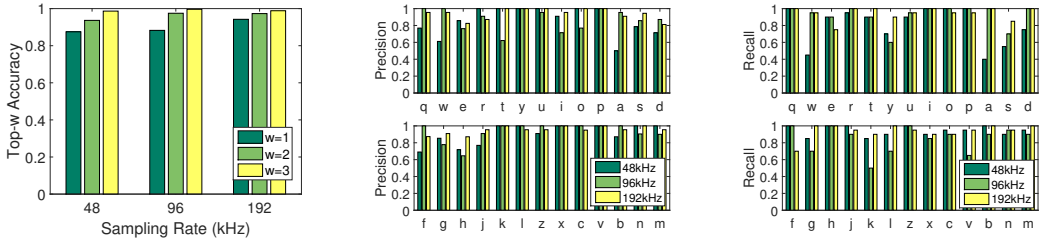
Figure 4.10: Differential between measured TDoA and theoretical TDoA with 192kHz sampling rate.

by the same hyperbola as that of the key w, as shown in Figure 4.1(b). Moreover, two neighboring keys may produce similar keystroke sounds resulting in high similarity of MFCC features. This could also lead to a few keystrokes mistakenly recognized as different keys.

The precision and recall of recognizing each alphabetic key is shown in Figure 4.9(c). It combines the results for all three keyboards. Overall, the average precision is about 87% and the average recall is about 85%. This result shows that our system could recognize each individual alphabetic letter without linguistic model. Thus, our system could recover passwords consisting of random combination of letters.

TDoA Ranging

We next study how accurately we can measure TDoAs with the phone's microphone capability of $192kHz$ sampling rate. We compare the measured distance difference (i.e., measured TDoA multiplies velocity of sound) of the keystroke sound to the true distance difference of the key at the two phone microphones. We use Apple wireless keyboard and each alphabet key is typed ten times in the experiment. Figure 4.10 illustrates mean and standard deviation of error for each key. We observe that the average ranging error is about $2mm$ indicating that mm-level accuracy could be achieved at $192kHz$ which is the frequency supported by the smartphone audio hardware.



(a) Top-w accuracy with different sampling rates.

(b) Precision.

(c) Recall.

Figure 4.11: Performance of set-keystroke based processing using different sampling rates.

Effect of Sampling Rate

The impact of the sampling rate on the recognition accuracy is shown in Figure 4.11(a). The Microsoft surface keyboard is used in the experiment with the sampling rates of $48kHz$, $96kHz$ and $192kHz$. From Figure 4.11(a), we observe that higher sampling rate indeed improves the recognition accuracy as it provides higher TDoA resolution to discriminate the close by keys. In particular, the accuracy is improved from about 84.8% to 94.2% for top-1 candidate when increases the sampling rate from $48kHz$ to $192kHz$. However, the improvement on the top-3 candidates is marginal since these top-3 candidates usually covers these keys are spaced closely with the similar TDoA. The improved sampling frequency thus has limited improvement for the top-3 candidates.

Figure 4.11(b) and Figure 4.11(c) show the precision and recall for each key, respectively. We find higher sampling frequency in general improves the precision and recall, especially for the keys that hard to be distinguished at lower sampling frequency. For example, the keys w and a are physically close and the corresponding recalls and precisions are very low (at around 50%) when the sampling frequency is $48kHz$. They are improved to over 90% for both w and a at the sampling frequency of $192kHz$. This is also because higher sampling rate provides better TDoA resolution to distinguish close by keys.

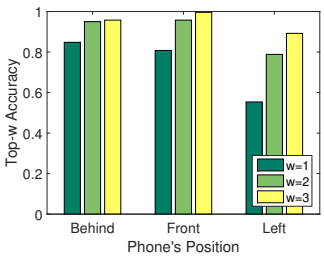


Figure 4.12: Top-w accuracy of set-keystroke processing with different placements of the phone to the keyboard.

Effect of Phone's Placement

Next, we study the performance under different phone placements. As shown in Figure 4.8, the phone is placed at three different positions (i.e.,*behind*, $front$ and $left$) close to the Apple wireless keyboard. Figure 4.12 depicts the top-w accuracy for three phone placements. We observe that the placements of $front$ and *behind* result in higher accuracy than that of $left$. This is inline with our analysis on phone placement shown in Figure 4.2(a). This also shows that the primary placement of phone-keyboard (i.e., *behind*) when the users use external keyboard is more vulnerable to keystroke snooping. In particular, top-1 2 and 3 accuracies are about 84.8%, 95%, and 95.7% for the primary placement respectively, whereas they are about 80.1%, 95.7%, and 99% for $front$ placement respectively.

4.5.3 Performance of Single-keystroke Based Processing

We evaluate the naive approach, the single-keystroke based processing, by using the same dataset as we used for the set-keystroke based processing. Figure 4.13 shows the overall accuracy under different sampling rates. As expected, the naive approach has worse performance for top-1 accuracy when comparing to the set-keystroke based processing. This is because the single-keystroke based processing identifies keys based on a single TDoA value without exploiting acoustic features and statistic information of keystrokes of the same key. In particular, the top-1 accuracy of the single-keystroke based processing is about 60% at $48kHz$. The accuracy however increases dramatically to 89.6% for the top-2 accuracy and to 97.7% for the top-3 accuracy. This is due to

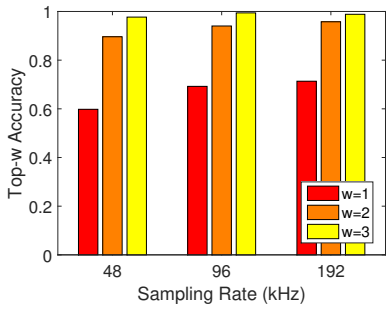


Figure 4.13: Top-w accuracy of single-keystroke processing with different sampling rates.

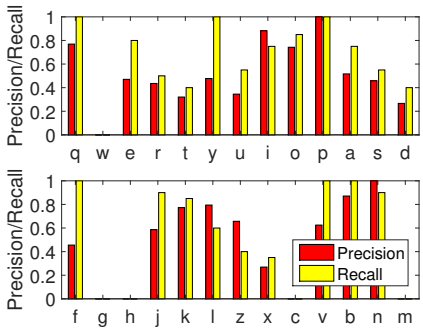


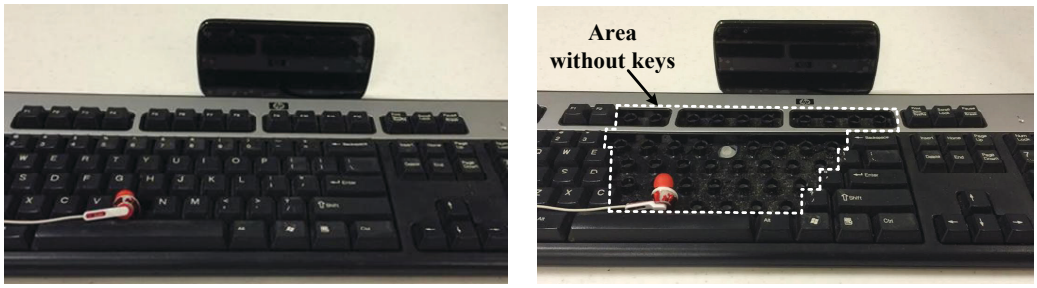
Figure 4.14: Precision and recall of single-keystroke processing with $48kHz$ sampling rate.

that the top-2 and top-3 candidates usually cover the close-by keys that are hard to distinguish with one single TDoA.

In addition, the accuracy can be further improved by increasing the sampling rate to $96kHz$ or $192kHz$. With $192kHz$, the single-keystroke based processing can achieve 95% and 98% accuracy for top-2 and 3 candidates respectively. Figure 4.14 further shows the precision and recall of each key at $48kHz$ sampling rate. Since the single-keystroke based processing is hard to distinguish two keys with theoretical TDoAs within one sample, several keys are mistakenly recognized as others, such as keys w,g,h,c and m shown in Figure 4.14.

4.5.4 Multi-path Investigation

Multi-path Effects through Keys . Like many other wireless signals, multi-path effects may change the characteristics of acoustic signal. For keystrokes, because



(a) Keyboard with keycaps. (b) Keyboard without keycaps.

Figure 4.15: Experimental Setups for multi-path investigation.

the sound sources are mostly inside each key and always below neighboring keys, it is important to understand the impact of multipath on the TDoA estimation in our system. Particularly, we conduct following experiments: as shown in Figure 4.15(a), we first use the Samsung Galaxy Note 3 to play a pre-recorded chirp sound signal via a earbud, which is to make sure the sound comes below neighboring key caps and thus has multipath effects, for 20 times at each target key's position on a regular keyboard. Next, we repeat the same experiment, but remove the neighboring key caps as shown in Figure 4.15(b). Note that the phone is placed at 90 degree angle with the keyboard and the microphones are at a higher level than the keys on the keyboard to better study the multipath effects through keys. We remove all the keys between the earbud and the phone in order to simulate a lower multi-path environment. Figure 4.16 shows the difference of measured TDoA between such keyboards with different levels of multi-path effects. The average difference is only about 1 sample, and we thus conclude that the impact of multi-path (key caps on keyboard) does not have much influence on the TDoA estimation.

Non Line of Sight Effects. In the experiment, we use two mobile phones(i.e., Samsung Galaxy Note 3 and HTC Evo 4G) on two tripod at heights 1 meter above the ground as shown in Figure4.17(a). Similarly, we use the Samsung phone to record the chirp sound played by the HTC phone for 20 times. We align two phones to make sure the measured TDoA is close to 0 in the line-of-sight scenario. Next, we repeat the experiment, but with a thick card board separator placed in between the two phones to simulate the non-line-of-sight scenario as shown in Figure 4.17(b). Figure 4.18(a)

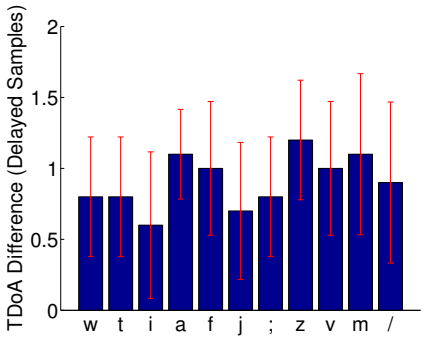


Figure 4.16: Differential TDoA between higher multi-path keyboard and lower multi-path keyboard (removed keys).

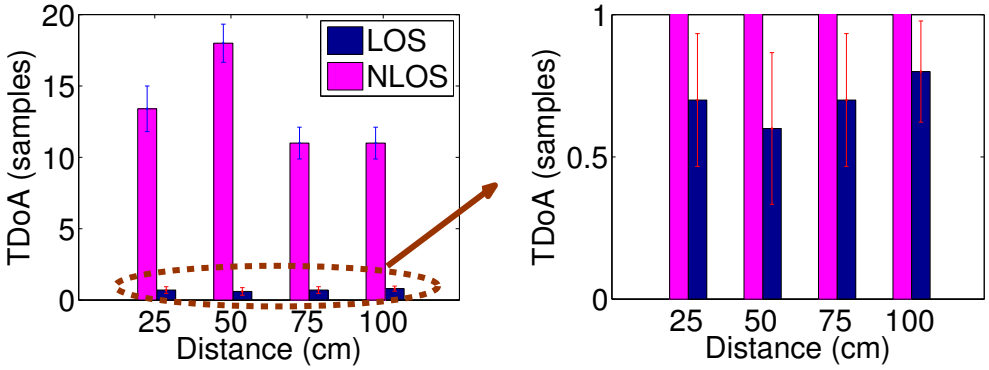
(a) Line-of-sight(LOS) environment. (b) Non-line-of-sight(NLOS) environment.

Figure 4.17: Experimental Setups for multi-path investigation.

shows the overall statistics of TDoAs in both the line-of-sight(LOS) scenario and non-line-of-sight(NLOS) scenario. Figure 4.18(b) is the enlarged part within the circle in Figure 4.18(a). Compared to the LOS scenario, the measured TDoAs increase dramatically in the NLOS scenario.

4.6 Discussion

Environmental Accuracy. There are several factors that have an important effect on accuracy including phone placement, multi-path, and noise. Our system has been evaluated with different phone placements close to the keyboard. Accuracy would significantly degrade for recordings taken at larger distances and meter-level distances would require much larger microphone separation, for example by using multiple cooperating devices. We believe, however, that close proximity is possible even in adversarial



(a) LOS and NLOS TDoA estimation. (b) Zoomed LOS and NLOS TDoA estimation.

Figure 4.18: TDoA estimation for LOS and NLOS environment.

settings, for example if the adversary co-opts the users own phone or if the attack takes place in relatively confined space (e.g., airplane). As any time-of-arrival related localization technique, our system relies on a detectable signal arriving on the line-of-sight path. If this path is significantly attenuated by an obstacle, our system will measure a reflected signal which leads to errors too large to allow for recovery of keystrokes (as illustrated in section 4.5.4). Phone placement close to the keyboard makes such an obstacle unlikely, however. We evaluate our system in typical office environments (i.e., two laboratory rooms with ambient noises (e.g., HVAC noise)), and our results show little impact under such ambient noises. Although we observe that loud noises (e.g., people talking) could impact the detection accuracy, we believe that additional filtering or context-based word correction could further improve the accuracy.

Security Concerns. To our knowledge, this is the first demonstration of acoustic keystroke recovery that raises more serious concerns regarding password snooping. It appears practical that malicious background apps with microphone access could recover passwords entered from a nearby keyboard (either an associated Bluetooth keyboard or a keyboard used for another device). If high definition stereo audio trickles down from professional video conference systems, to voice over ip and video calling apps, keys typed during a call could potentially be recovered by the remote party. It may also be possible for an adversary to inconspicuously place a phone near a victim's keyboard, particularly in tight settings such as an airplane. That said, the attack is

currently only possible with select phone models that expose stereo recording and have large microphone separation and even at future expected sampling rates of 192kHz there is only a moderate chance of accurately capturing a long random password on first attempt. Still, this significantly reduces the password entropy to a small set of candidates that can be brute-forced and the accuracies would be sufficiently higher for the many weaker passwords in use, when combining the keystroke recognition results with knowledge about common password patterns.

While there is already considerable awareness of privacy risks associated with microphones, this awareness usually extends only to spoken words and not necessarily to keystrokes. Users might therefore type sensitive information even if they know that recording devices are present. Overall, these results indicate that microphone access on mobile device should be tightly controlled and we hope to raise awareness to that the recoverable information from mobile device audio recordings extends far beyond spoken conversations.

Localization Implications. More generally, the results show that low-multipath scenarios exist where mobile audio enable mm-level ranging and localization. Such high accuracies could be exploited for numerous applications from motion tracking [34], over driver phone use detection [35], to user interface improvements [102]. Currently, app-level access to these audio capabilities is still very limited; the capabilities are primarily used for specific functions such as noise cancellation during calls or high definition audio playback. In light of these localization results, we argue that app-level software access to multiple microphones and high sampling rates for localization purposes should become a higher priority.

4.7 Conclusion

In this work, we show that microphones on a single off-the-shelf phone can be used to discriminate mm-level position differences, which not only creates potential security and privacy concerns related to recovering keystrokes being typed on a nearby keyboard, but could also benefits a broad range of applications relying on fine-grained localization (e.g.,

sensing touch interaction on surfaces around mobile devices and tracking speakers in multiparty conversations in a meeting room). The implemented system does not require any training or linguistic model, which makes it applicable in real-world adversarial context and has the capability to recover random typing (e.g., passwords). In particular, our system exploits digital acoustic signals received at the microphones from an off-the-shelf phone and leverages the integration of geometry-based TDoA and fine-grained acoustic signatures to exceed the resolution limit of TDoA and accurately identify keystrokes. Extensive experiments involving three types of keyboards demonstrate that, with $48kHz$ sampling rate, our proposed system can accurately identify a set of keystrokes with over 85% accuracy. The accuracy of our system can achieve as high as 94% with the higher sampling rate (i.e., $192kHz$). Additionally, our system can snoop even a single keystroke input at the accuracy of 97% among the top-3 candidate keys with $48kHz$ sampling rate.

Chapter 5

High-throughput and Inaudible Acoustic Communication with Non-linearity of Microphones

5.1 Background

5.1.1 Microphone System

The most commonly used microphones are electret condenser microphone (ECM) and micro-electro-mechanical system (MEMS) microphone. Due to the small package size and low power consumption, MEMS microphones currently dominate the market of audio device on mobile devices including smartphones and wearable devices, etc [103]. A MEMS microphone on a commercial off-the-shelf mobile device mainly consists of four components, i.e., transducer, pre-amplifier, inbuilt low-pass filter and analog-to-digital converter (ADC), as shown in Figure 5.1. When an acoustic signal carries the energy towards a microphone, the transducer of the microphone first transforms the mechanical sound waves to electric signals through the electromagnetic induction [104]. Then, the pre-amplifier enhances the electric signal to improve the signal-to-noise ratio (SNR) of the signal transformed from sound waves. Next, the inbuilt low-pass filter eliminates the high-frequency harmonic components from the electric signals so as to

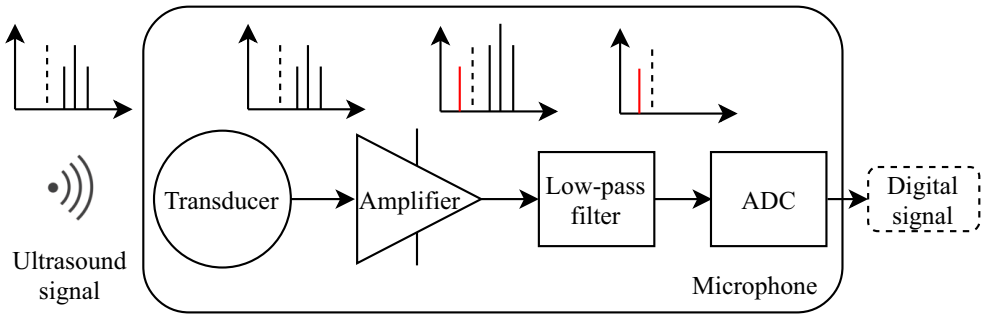


Figure 5.1: Microphone architecture and illustration of its non-linearity.

match the sampling rate of ADC in the microphone. The inbuilt low-pass filter usually sets the cut-off frequency as $24kHz$ due to the maximum sampling rate of $48kHz$ in the ADC of most widely-deployed mobile devices. After that, ADC samples the electric signals and stores the values as digital signals, which can represent the recorded acoustic signals.

5.1.2 Non-linearity of Microphone

Due to the limited sampling rate (i.e., $48kHz$) of the microphone in mobile devices, the devices can only record the acoustic signals within a specific frequency range (i.e., $< 24kHz$). However, when receiving an ultrasound signal, the pre-amplifier of the microphone exhibits *non-linearity* in the ultrasound frequency range [105, 106], which is feasible to make the ultrasound signal become recordable by the inbuilt microphones.

Specifically, the non-linearity of microphones can be modeled theoretically. Assume that the microphone receives an acoustic signal s_{in}. After the sound is picked up and amplified by the microphone's transducer and pre-amplifier, the recorded signal s_{out} can be represented as:

$$\begin{aligned} s_{out} = &\quad A_1 s_{in} + \sum_{i=2}^{\infty} \delta(f) A_i s_{in}^i \\ \approx &\quad A_1 s_{in} + \delta(f) A_2 s_{in}^2, \end{aligned} \tag{5.1}$$

where A_i is the energy gain for the ith order term and $\delta(f)$ is an indicator function. Although the non-linear output is an infinite power series, the value of A_i decreases with the increase of i and the third and higher order terms are extremely small. Thus we only consider the linear and quadratic terms. The indicator function $\delta(f)$ is defined as:

$$\delta(f) = \begin{cases} 0, & f < f_0 \\ 1, & otherwise, \end{cases} \tag{5.2}$$

where f_0 is the critical frequency of the non-linearity. We empirically find the critical frequency $f_0 \approx 18kHz$ in the most commercial mobile devices. This indicates that the pre-amplifier generates additional frequency components other than original frequency component, when the microphone receives an ultrasound signal. The quadratic

term of Equation 5.1 exhibits the non-linearity of microphones. The non-linearity of microphones provides the feasibility of utilizing ultrasound for communication. As shown in Figure 5.1, by selecting an appropriate modulation technique, the modulated data bits carried on an ultrasound carrier can be transmitted inconspicuously, and the microphone has potential to recover the original data bits with the non-linearity property. We are thus motivated to explore how to use such microphone's non-linearity to achieve high-throughput inaudible communication rather than using the limited near-ultrasound frequency band.

5.2 Achieving High Throughput While Keeping Inaudibility

5.2.1 Challenges

To achieve high-throughput and inaudibility of the acoustic communication simultaneously, the design of our system mainly involves the following challenges.

High-throughput and Inaudible Communication for General Mobile Devices. The high-throughput of acoustic communication requires to utilize a wide frequency bandwidth for data transmission. However, for most mobile devices, the inbuilt microphone only has limited ADC sampling rate (i.e., $48kHz$), thus the device can only record acoustic signals within $24kHz$ according to Nyquist theorem [49]. Additionally, the audio signal with the frequency larger than $18kHz$ is hardly audible to most humans [107]. In order to achieve inaudibility, a narrow frequency bandwidth (i.e., 18-$24kHz$) has to be used for communication, which significantly reduces the possible communication throughput. Therefore, it is essential to find a way to increase the frequency bandwidth for communication so as to improve the throughput, while keeping the inaudibility.

Robust Communication Under Various Environmental Factors. There are many environmental factors affecting the robustness of acoustic communication systems. For instance, acoustic frequency channels may be easily affected by other sound sources in the environment, such as HVAC (Heating, Ventilation, Air Conditioning)

noises, people talking, etc. Moreover, due to the omni-directional propagation of acoustic signals, the acoustic communication suffers from significant multipath effect, which may induce the time selective fading and frequency selective fading problem. Therefore, our system needs to transmit data via acoustic channels in a way that is robust to these noisy environments and signal interference factors.

5.2.2 Integrating Non-linearity with Signal Multiplexing and Modulation Techniques

To achieve high-throughput and inaudibility at the same time, we use *orthogonal frequency division multiplexing* (OFDM) technique and *amplitude modulation* (AM) with the *non-linearity model of microphone* to modulate the data bits onto multiple subcarriers in an ultrasound frequency band to transmit data.

Achieving High-throughput based on OFDM. In order to achieve high-throughput communication, we utilize OFDM technique to modulate the data signals on multiple subcarriers to convey multiple data bits concurrently. In a communication system, frequency division multiplexing (FDM) is a widely-used multiplexing technique, which divides the total frequency bandwidth into a series of non-overlapping frequency bands, i.e., subcarriers, and sets a guard band between every two subcarriers to avoid interferences. However, the guard band wastes the scarce spectrum of acoustic communication. To improve the efficiency of spectrum utilization, we use OFDM multiplexing that utilizes orthogonal subcarriers[1] in the system. Furthermore, the communication capability of an OFDM system increases as the increase of the frequency bandwidth. Our system thus aims at using the whole acoustic frequency band, including both audible and inaudible frequency band, to achieve high-throughput.

Enabling Inaudibility via AM Modulation and Microphone's Non-linearity. Due to the utilization of audible frequency band in OFDM, directly transmitting the

[1]Orthogonal subcarriers in OFDM have overlapping spectra between subcarriers, but this overlap of spectral energy does not interfere system to recover the original signal.

OFDM-multiplexed signals remains audible on ambient areas. To make the acoustic communication inaudible, we further integrate AM technique to transmit audible OFDM-multiplexed signal via an ultrasonic carrier. Specifically, we define the OFDM-multiplexed signal on the m_{th} OFDM subcarrier (subcarrier frequency is f_m) is: $m(t) = \cos(2\pi f_m t)$. Then $m(t)$ can be modulated with an ultrasound carrier signal $\cos(2\pi f_c t)$ ($f_c \gg f_m$) through AM. The modulated signal is:

$$s_{in} = \cos(2\pi f_c t) \cdot (1 + \cos(2\pi f_m t)). \tag{5.3}$$

Combined with the non-linearity of microphone, i.e., Equation 5.1, we can derive the signal s_{out} that is recorded by the device's inbuilt microphone as:

$$\begin{aligned} s_{out} =& A_1 \cos(2\pi f_c t) \\ &+ \frac{A_1}{2}(\cos(2\pi(f_c + f_m)t) + \cos(2\pi(f_c - f_m)t)) \\ &+ \frac{A_2}{4}(4\cos(2\pi f_m t) + 3cos(4\pi f_c t) + \cos(4\pi f_m t)) \\ &+ \frac{A_2}{8}(\cos(4\pi(f_c + f_m)t) + \cos(4\pi(f_c - f_m)t)) \\ &+ \frac{A_2}{2}(\cos(2\pi(2f_c + f_m)t) + \cos(2\pi(2f_c - f_m)t)) \\ &+ \frac{3A_2}{4}. \end{aligned} \tag{5.4}$$

From Equation 5.4, we can find that the frequency components contain $f_c, f_c - f_m, f_c + f_m, f_m, 2f_c, 2f_m,\ 2(f_c + f_m), 2(f_c - f_m), 2f_c + f_m, 2f_c - f_m$. Since f_c ($f_c \gg f_m$) is the ultrasound carrier frequency, the components with the frequency f_c can be eliminated with the low-pass filter in the microphone, as shown in Figure 5.1. After that, the signal becomes:

$$s_{out} = A_2 \cos(2\pi f_m t) + \frac{A_2}{4}\cos(4\pi f_m t) + \frac{3A_2}{4}, \tag{5.5}$$

which only contains the frequency components f_m and $2f_m$. In addition to the transmitted frequency component f_m, we note that the induced component $\frac{A_2}{4}\cos(4\pi f_m t)$ may impact the OFDM signals on other specific subcarrier (i.e., the one with subcarrier frequency $2f_m$). It would lead to significant errors when the system recovers the original signal $m(t)$. We next introduce how to eliminate this interference effect caused by the frequency component $2f_m$.

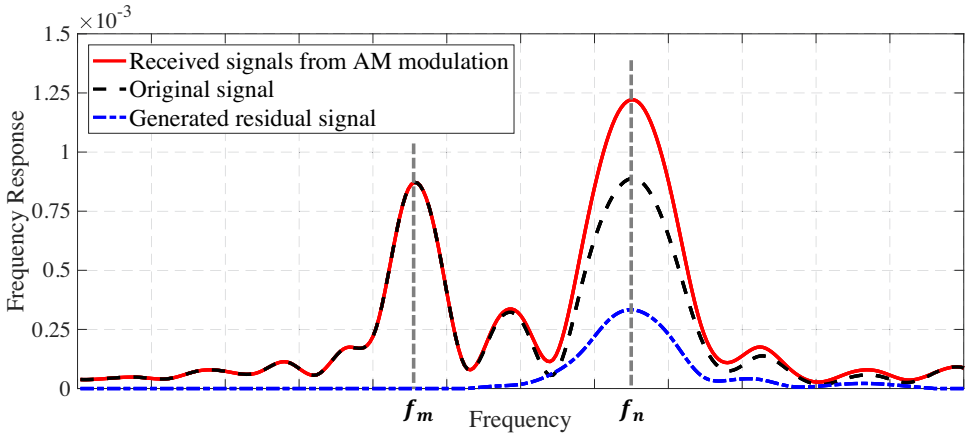


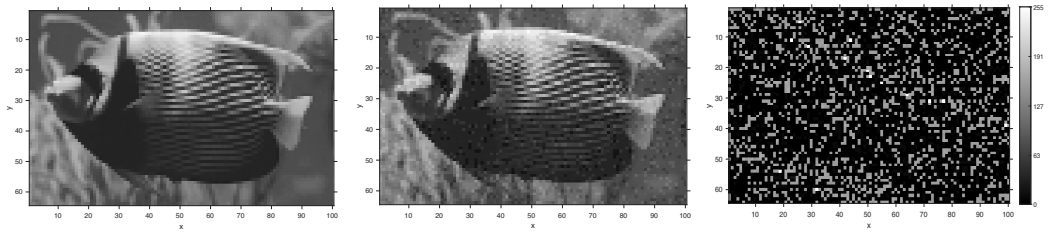
Figure 5.2: Illustration of the signals on the subcarrier f_n affected by the residual signal (e.g., $f_n = 2f_m$).

5.2.3 Eliminating Unrelated Residual Signals Induced by AM Modulation

As shown in Equation 5.5, the unrelated frequency component $2f_m$, which we call *unrelated residual signal*, may greatly impact the system's capability of recovering original signals and interfere with the OFDM signals transmitting over the subcarrier whose frequency is $2f_m$. We simulate that a signal, consisting of two frequency components (i.e., f_m and $f_n = 2f_m$), is transmitted with an ultrasound carrier through AM modulation. As shown in Figure 5.2, we observe that the received demodulated signal has a great interference on the frequency f_n due to the induced unrelated residual signal.

To eliminate the residual signal, we elaborately modify the OFDM-multiplexed signal before the AM modulation in the transmitter side. According to Equations 5.1 and 5.3, the linear term (i.e., $A_1 s_{in}$) of the non-linearity model contains ultrasound frequency f_c and thus would be filtered out by the inbuilt low-pass filter of the microphone. Hence, we only analyze the quadratic term (i.e., $A_2 s_{in}^2$) for the elimination scheme design. Specifically, we define the analog OFDM symbol waveform as $s(t)$ that carried the data bits to be transmitted. According to Equation 5.1, the quadratic term s_q can be represented as:

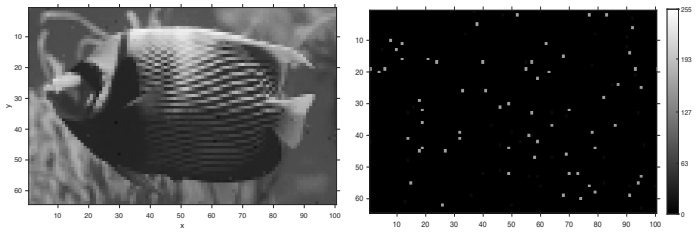
$$\begin{aligned} s_q =& A_2(\cos(2\pi f_c t) \cdot (1 + s(t)))^2 \\ =& \frac{A_2}{2}(1 + \cos(4\pi f_c t)) + \frac{A_2}{2}(1 + \cos(4\pi f_c t))(2s(t) + s^2(t)). \end{aligned} \tag{5.6}$$



(a) Transmitted image

(b) Received image w/o elimination scheme

(c) Error comparision w/o elimination scheme



(d) Received image wi-th elimination scheme

(e) Error comparision with elimination sche-me

Figure 5.3: Image comparison between transmitted and received images without and with unrelated residual signal elimination scheme (BER = 10.2% vs. BER = 0.3%).

We observe that the first term (i.e., $\frac{A_2}{2}(1+\cos(4\pi f_c t))$) of Equation 5.6 only contains the ultrasound frequency component $2f_c$, which can be neglected due to the low-pass filter in the microphone. As $s(t)$ contains frequency components crossing all the OFDM subcarriers, the second term (i.e., $\frac{A_2}{2}(1+\cos(4\pi f_c t))(2s(t)+s^2(t))$) of Equation 5.6 would produce unrelated residual signals, as per Equation 5.4. In order to eliminate these residual signals, we use following signal $s_e(t)$ to replace the OFDM symbol waveform $s(t)$ before AM modulation:

$$s_e(t) = \sqrt{(s(t)+1)} - 1. \tag{5.7}$$

Therefore, we have $2s_e(t) + (s_e(t))^2 = s(t)$, and Equation 5.6 would be changed to:

$$s_q = \frac{A_2}{2}(1+\cos(4\pi f_c t)) + \frac{A_2}{2}(1+\cos(4\pi f_c t))s(t). \tag{5.8}$$

Thus, by neglecting the components containing ultrasound frequency f_c, only the component $\frac{A_2}{2}s(t)$ can be preserved through the microphone's low-pass filter. With further modulating the modified OFDM signal (i.e., $s_e(t)$) with the ultrasound carrier through AM, the recorded signal in the receiver side would not contain the unrelated residual signals anymore.

To validate whether the proposed scheme could eliminate the unrelated residual signals, we conduct a simulation experiment, in which a software-defined transmitter (including OFDM, elimination approach and AM) and a software-defined receiver (including the microphone's pre-amplifier with non-linearity, built-in low-pass filter and OFDM demodulation process) are implemented. We use an additive white Gaussian noise (AWGN) model in the communication channel and SNR is set to 36. An image[2] (i.e., Figure 5.3(a), $6.4kB$) is transmitted from the transmitter to the receiver. Without the proposed elimination scheme (i.e., using $s(t)$ as OFDM symbol waveform), we observe that the received image (i.e., Figure 5.3(b)) contains significant errors (i.e., bit error rate (BER)=10.2%), and the error difference between the transmitted and received images is shown in Figure 5.3(c). Differently, by using the elimination scheme (i.e., using $s_e(t)$), the received image is almost error-free (i.e., BER=0.3%), as shown in Figure 5.3(d). The error difference between the transmitted and received images is shown in Figure 5.3(e). This result demonstrates that the proposed elimination scheme is efficient to reduce the effect caused by the unrelated residual signals for achieving robust acoustic communication.

5.3 System Overview

The architecture of BatComm is shown in Figure 5.4, which consists of two parts, i.e., transmitter and receiver.

Transmitter Design. The transmitter is responsible to modulate data bits to an ultrasound signal for the high-throughput and inaudible acoustic communication. The data bits are first encoded with BCH error correction code [108] and further re-ordered through an interleaving technique to reduce the unpredicted errors during the signal propagation. Then, the encoded data is converted to phase values through the digital modulation technique, i.e., differential phase shift keying (DPSK). To fully utilize the scarce frequency band for communication, the OFDM technique is further applied to modulate the phase values to multiple subcarriers for concurrent data transmission.

[2]Each pixel in the grayscale image contains 8 bits.

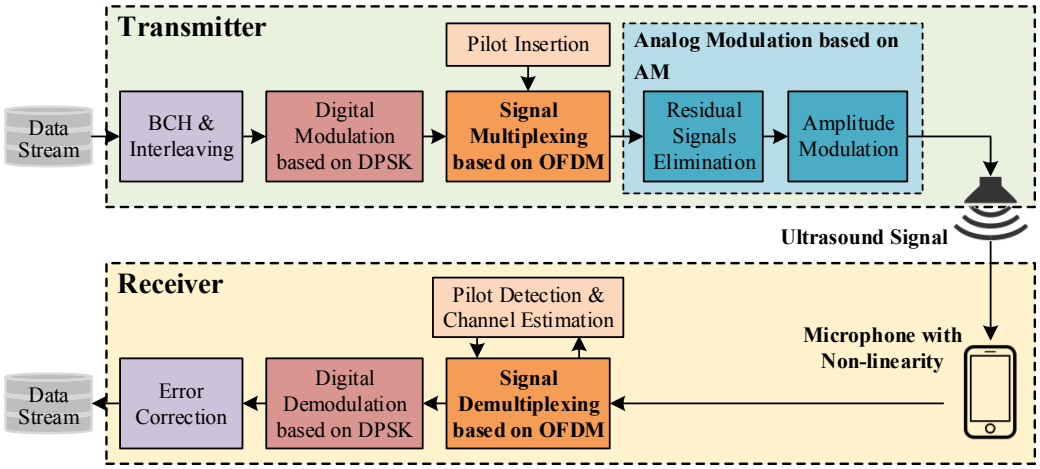


Figure 5.4: Architecture of BatComm.

During the OFDM, a pilot is inserted in OFDM signals for channel estimation so as to eliminate the impact of multipath effect on the received signals. After that, the OFDM symbol waveform $s(t)$ is modified to $s_e(t)$ for eliminating the unrelated residual signals, which are generated through AM under the non-linearity of microphones. Then the modified OFDM symbol waveform $s_e(t)$ is modulated onto the ultrasound carrier through AM for inaudible communication.

Receiver Design. The receiver in our system is a commercial mobile/IoT device (e.g., a smartphone) with an inbuilt microphone, which records and demodulates the received ultrasound signal to receive data. Taking advantage of the modeled non-linearity of microphone, the receiver can demodulate the received ultrasound signals to obtain the OFDM symbol waveform. After that, the receiver performs the demultiplexing on the recorded OFDM waveform to extract the phase values. Additionally, the pre-inserted pilot in the OFDM signals is used for channel estimation to eliminate the interference of multipath effect. Further, the extracted phase values are mapped into the digital data bits through DPSK demodulation. Finally, the receiver performs error correction on the digital data bits with the pre-inserted BCH code and the interleaving matrix to mitigate the unpredicted errors.

5.4 Transmitter Design

In practical communication scenarios, there exist many potential factors affecting the communication system, such as ambient noises, multipath effect, etc. To improve the robustness, we integrate a series of techniques (e.g., BCH code and interleaving, DPSK, pilot signal) in the transmitter design to make BatComm resilient to various interferences.

5.4.1 Error Correction via BCH Codes and Interleaving

In the proposed acoustic communication, it may have unpredicted errors induced in the propagation channel. To mitigate these errors, we first encode the digital data with BCH code [109], which is a widely-used error correction code in communication field. Specifically, the digital data is encoded with (N, K)-BCH code, where N is the length of the encoded data, K is the length of original digital data. The (N, K)-BCH encoded data uses N-K bits error correction code.

Additionally, BCH error correction code is satisfactory to correct randomly distributed errors in the signals. However, the errors usually burst in some specific domains of the signal, due to the intensive noises appearing in some specific frequency band or specific time period. To address this problem, we further apply a matrix-based interleaving approach to interleave the data stream in a particular known order, which could convert bursts of errors into random-like errors. Specifically, we first take a block of encoded data bits and fill it in a $M \times N$ matrix following the *row order*, i.e.,

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix}. \tag{5.9}$$

Then, this block of encoded data bits is transmitted following the *column order* of the matrix, i.e., $x_{11}, x_{21}, \cdots, x_{M1}, x_{12}, x_{22}, \cdots, x_{M2}, \cdots, x_{1N}, x_{2N}, \cdots x_{MN}$. Such an interleaving approach could greatly improve error correction based on BCH codes when burst errors occur.

5.4.2 Digital Modulation based on DPSK

In order to transmit data in the air, the digital data bits should be first modulated, which is necessary for digital-to-analog conversion. The most commonly used digital modulation techniques are amplitude shift keying (ASK), frequency shift keying (FSK) and phase shift keying (PSK). ASK and FSK utilize the amplitude and frequency of carrier signals to modulate the digital data bits, respectively. However, due to the vulnerability to noises and the requirement of wide bandwidth, they are inappropriate for acoustic communication. Moreover, PSK modulates the data bits on several absolute phase values, which is efficient to utilize the scarce acoustic spectrum. However, in practical communication scenario, the multipath propagation of acoustic signals and ambient noise may induce unpredictable phase shift on the signals, which leads to errors in the absolute phase values.

To solve this problem, we use differential phase shift keying (DPSK) for the digital modulation in our system. Specifically, we modulate n-bit digital data to one of the 2^n possible phase values, which are uniformly spread in the range of $[0, 2\pi]$. Instead of transmitting this modulated phase value p^t, the transmitter sends the phase value p_i, satisfying $p_i = p_{i-1} + p^t$, and uses the phase difference between two successive transmitted samples (i.e., p_{i-1}, p_i) to carry the modulated phase value. In the receiver side, the system can demodulate the data by mapping the differential phase values into the digital bits.

5.4.3 Signal Multiplexing based on OFDM

After digital modulation, our system uses multiple orthogonal subcarriers in OFDM to carry these modulated samples for achieving high-throughput communication. Particularly, OFDM transforms the modulated phase values on multiple subcarriers from frequency domain to a time-domain analog waveform through inverse fast Fourier transform (IFFT) (i.e., 1024 points in our system). Considering the unpredicted interference during the propagation of acoustic signals, we further extend the standard OFDM, making it suitable for our high-throughput inaudible acoustic communication.

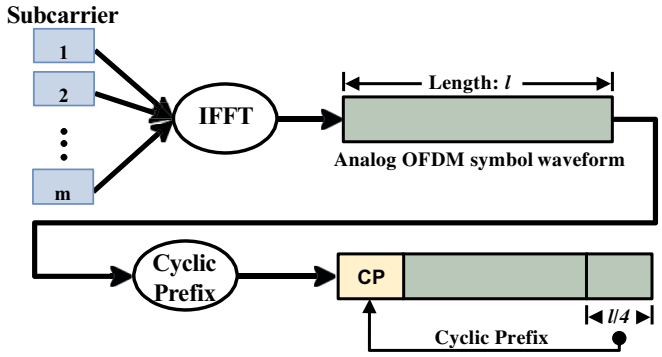


Figure 5.5: OFDM symbol structure in multiplexing.

Adding Preamble and Cyclic Prefix for Synchronization Issue. To reliably transmit data from the transmitter to receiver, the transmitter is required to synchronize with the receiver. To achieve the synchronization, we divide the data stream to be transmitted into a set of frames. In each frame, there are 30 OFDM symbols, each of which modulates data bits onto multiple OFDM subcarriers. To ensure that the receiver can recover the complete data from acoustic signals, it is necessary for the receiver to precisely find the beginning of each frame. We thus add a preamble in the beginning of each frame, and the preamble is designed following the protocol of IEEE 802.11a [110]. Moreover, the multipath effect introduces the inter-symbol interference [111] between OFDM symbols. To eliminate the interference, the transmitter adds a cyclic prefix in the beginning of each symbol, which is designed as the last quarter of the symbol, as shown in Figure 5.5. With the preamble and cyclic prefix, the receiver can find the beginning of each frame and obtain OFDM symbols precisely.

Inserting Pilot for Channel Estimation. Due to the multipath propagation of omni-directional acoustic signals, there exist time and frequency selective fading in the received signals [112]. To mitigate the time selective fading induced by multipath propagation, we insert pre-defined phase values, *comb-type pilot*, on one subcarrier (i.e., the subcarrier #502, corresponding to $23.53kHz$)[3]. The pilot symbol, *block-type pilot*, is used to mitigate the frequency selective fading at the receiver end, which is discussed in Section 5.5.1.

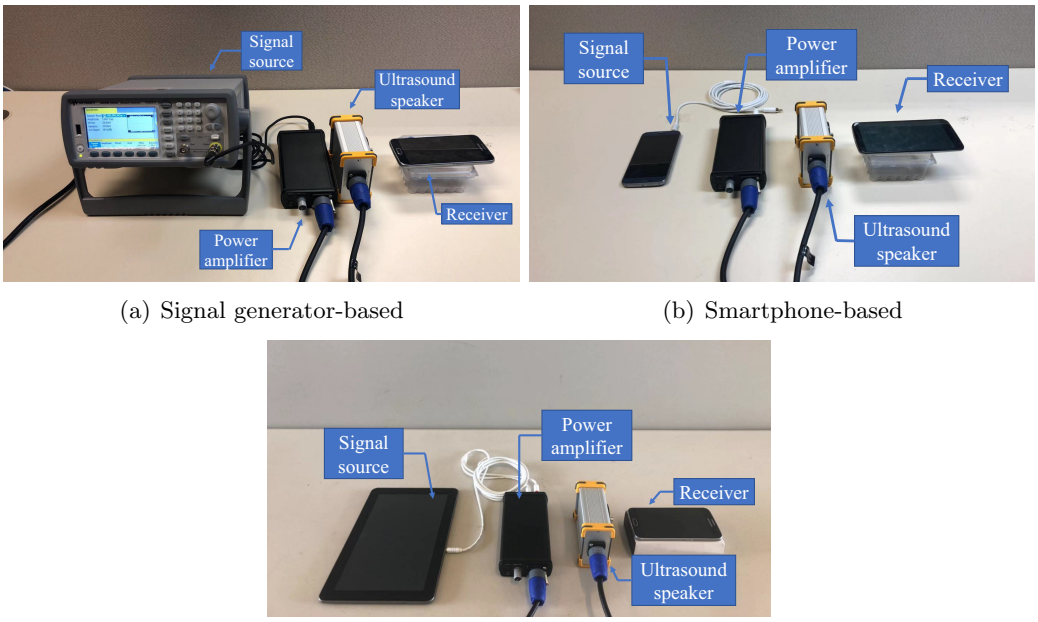
[3]1024-point IFFT/FFT in OFDM could have 512 orthogonal subcarriers corresponding to the bandwidth of 0-$24kHz$.

Frequency Bandwidth Selection to Resist Interference. Due to the interference of ambient noises in the environment, parts of the available acoustic frequency bands introduce significant errors in the communication. The normal sound sources (e.g., human speaking) usually generate acoustic signals lying in the frequency less than $8kHz$ [57]. On the other hand, commercial smartphones can only record the acoustic signals with the frequency up to $24kHz$, due to the limited sampling rate. To achieve high-throughput communication while keeping our system resilient to the daily noises, the operation bandwidth for the OFDM subcarriers is chosen as 8.06-23.53kHz, corresponding to the OFDM subcarrier #172 to #502 in our system.

5.4.4 Analog Modulation based on AM Towards Inaudibility

The analog OFDM symbol waveform $s(t)$ needs to be modulated onto an ultrasound carrier, so as to ensure the communication out of human perception. Particularly, as mentioned in Section 5.2, the combination of OFDM and AM techniques would produce unrelated residual signals, which would largely interfere our system and produce significant transmission errors. Thus, we use the modified OFDM symbol waveform $s_e(t)$, as per Equation 5.7, before AM modulation to eliminate the interference from the residual signals. Then we use AM modulation to modulate $s_e(t)$ onto the ultrasound carrier with frequency f_c, according to Equation 5.3.

Furthermore, the selection of the frequency f_c is critical. This is because a low carrier frequency induces the overlapping between AM-modulated ultrasound signal and OFDM-multiplexed signal, while a high carrier frequency leads to low-power transmitted signals, i.e., the transmitted signals are with low signal-to-noise ratio (SNR). As mentioned in Section 5.1.2, after the ultrasound is recorded by the microphone, the frequency components (e.g., $f_c - f_m, f_c + f_m, f_c$) including the carrier frequency f_c should be filtered out by the inbuilt low-pass filter. It indicates that the lowest frequency component (i.e., $f_c - f_m$) should be larger than the cut-off frequency of the filter f_l. Thus, we have $f_c \geq f_l + f_m$. According to Section 5.4.3, the operation bandwidth of OFDM subcarriers is around 8.06-23.53kHz. Therefore, the carrier frequency should meet the requirement of $f_c \geq 48kHz$. To maximize the SNR of the transmitted signal,



(a) Signal generator-based

(b) Smartphone-based

(c) Tablet-based

Figure 5.6: Illustration of three experimental settings. Signal source is a signal generator or mobile device (e.g., smartphone, tablet), and the receiver is commercial off-the-shelf smartphones.

unless mentioned otherwise, we set $f_c = 48kHz$ in our system.

5.5 Receiver Design

In this section, we introduce the technical details of how the receiver demodulates the ultrasound signals to recover the transmitted data bits.

5.5.1 Signal Demultiplexing based on OFDM

Due to the non-linearity of the microphone and the design of our residual signal elimination scheme, the transmitted OFDM waveform could be automatically picked up by the microphone, as mentioned in Section 5.2.2 and Section 5.2.3. In BatComm, the receiver first detects the preamble in the received signal to synchronize the OFDM frames, then demodulates the signal through OFDM technique and finally performs channel estimation to mitigate the multipath effect based on the pre-inserted pilot signal.

Preamble Detection and Synchronization. The receiver first needs to detect

the preamble of each OFDM frame so as to synchronize the signal frames. Since the prior knowledge of preamble is known to both the transmitter and receiver, we apply the correlation to detect preamble and find the beginning of the transmission:

$$R(n) = \sum_{i=0}^{M} x(i)y(n+i)^*, \tag{5.10}$$

where x and y are the preamble and received signal, respectively, M are the length of the preamble and n is the beginning index in a segment of the received signal for correlation. $()^*$ represents the conjugate operation. Based on the correlation values, the beginning of the preamble n_s can be found through $n_s = \arg\max_n R(n)$. After we detect each OFDM frame's beginning, the receiver further removes the cyclic prefix to extract the data from OFDM signals. Since the cyclic prefix serves as a guard between two successive OFDM symbols, the inter-symbol interference can be eliminated through removing the cyclic prefix. After that, the receiver demodulates the OFDM symbols through FFT operation to derive the phase values for further processing.

Pilot Detection and Channel Estimation. Due to the omni-directional propagation property of sound, the unpredicted propagation would introduce unexpected time and frequency selective fading errors in received signals. To deal with it, the receiver performs the channel estimation based on the pre-inserted pilot, which is discussed in Section 5.4.3. Specifically, the received pilot signal can be represented as $Y_p = HX_p + n$, where X_p and Y_p are transmitted and received pilot signals (i.e., inserted phase values), respectively. H is the channel response, and n is the ambient noises. Hence, the channel response can be represented as:

$$H = \frac{Y_p}{X_p} - \frac{n}{X_p} \approx \frac{Y_p}{X_p}, \tag{5.11}$$

where the ambient noise n almost has no effect on the channel response since the signals are modulated at the frequency higher than $8kHz$. With the estimated channel response, the data transmitted on other subcarriers can be calibrated as:

$$Y_c = H^{-1}Y_r, \tag{5.12}$$

where Y_r and Y_c are the received and calibrated signals, respectively. Through such a

channel estimation, the multipath effect can be mitigated in the received signals, which improves the robustness of communication.

5.5.2 Digital Demodulation & Error Correction

After demodulated OFDM signals, the receiver would further perform the digital demodulation based on DPSK to obtain the digital data bits. Specifically, the receiver first derives the differential phase values and extracts the digital data bits through the constellation mapping scheme. After that, the receiver recovers the digital data bits through interleaving. Since the dimension of matrix (i.e., Equation 5.9) for interleaving is known for the receiver, the receiver performs the reversed process of the interleaving to obtain the corrected-order data. Then, the receiver utilizes the BCH code to correct the unpredicted errors in the digital data bits.

5.6 Performance Evaluation

5.6.1 Experimental Setup & Methodology

Device and setting. To evaluate the performance of BatComm, we implement three settings of the transmitter to meet various application requirements. For these three settings, as shown in Figure 5.6, the signal source is a Keysight 33509B signal generator, a Galaxy S6 and a Samsung Tab P7510, respectively. The signal source devices are all tuned to the maximum volume for signal transmitting. In the frontend of these settings, we use an Avisoft ultrasonic dynamic speaker Vifa [113] and a portable ultrasound power amplifier [114] to transmit ultrasound signals. In addition, we use commercial off-the-shelf mobile devices (i.e., a Galaxy S6, a Galaxy Note 4, and a Samsung Tab P7510) as receivers to evaluate BatComm. We use the primary microphone, which is located at the bottom of the devices, to record the acoustic signal. Unless otherwise mentioned, the operation bandwidth for OFDM subcarriers is 8.06-23.53kHz, the digital modulation scheme is 16DPSK, the carrier frequency f_c for AM is 48kHz and error correction is based on $(63, 45)$-BCH code. The distance between the ultrasound speaker and the receiver (i.e., smartphone) is 3cm, which is natural and appropriate

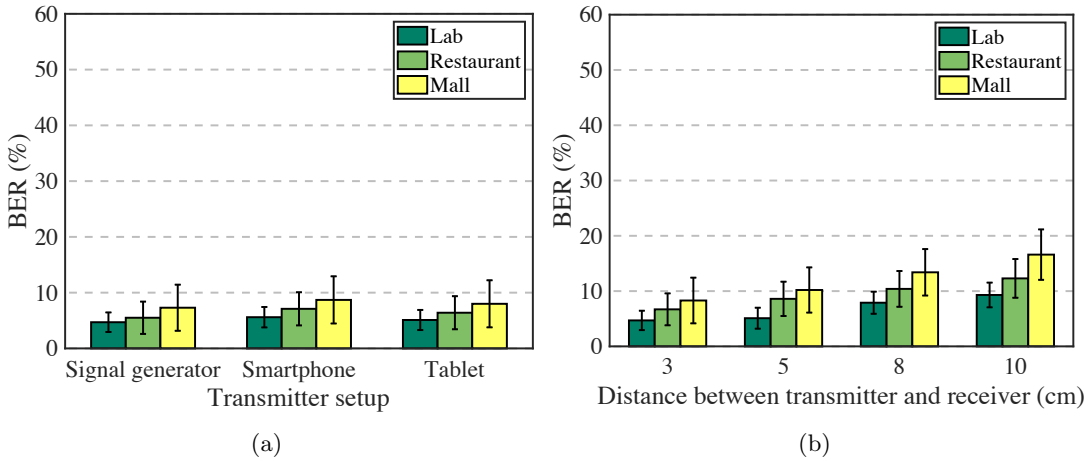


Figure 5.7: BER of BatComm under different (a) transmitter settings and environments (b) distances between transmitter and receiver (Throughput = 34.13$kbps$).

for a short-range communication application. We also evaluate longer distance, which is discussed in Section 5.6.2.

Data Collection. In order to evaluate BatComm and visually check the occurrence of errors, we transmit various grayscale images (i.e., 6.4kB for each) from the transmitter to the receiver. In each round of transmission, the transmitter transmits the modulated data through ultrasound signals, and we repeatedly conducted 20 rounds of transmissions for each environmental setup with various communication parameter settings. To test the impact of environmental noises on BatComm, we perform the experiments in three representative environments: lab, restaurant, and mall. The measured background noise levels of these three environments are 39.7dB, 57.3dB and 80.2dB, respectively.

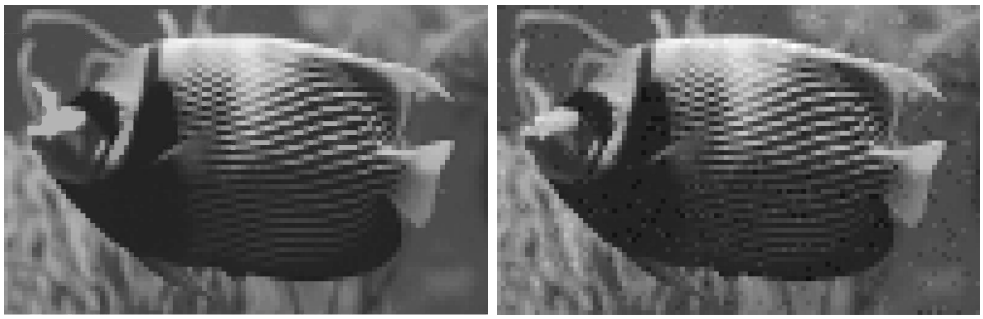
Evaluation Metrics. We mainly use two metrics to evaluate the performance of BatComm. (1) *Throughput*: Assume a data stream of D bits is transmitted from the transmitter to the receiver with a time of T seconds. The throughput of acoustic communication is defined as $\frac{D}{T}$ bits per second (bps); and (2) *Bit Error Rate (BER)*: Assume the system transmits n_t bits digital data. Due to noise, interference, distortion or bit synchronization errors, n_e bits data is altered during the communication. The BER is defined as $\frac{n_e}{n_t}$, which is presented as a percentage.

5.6.2 Overall System Performance

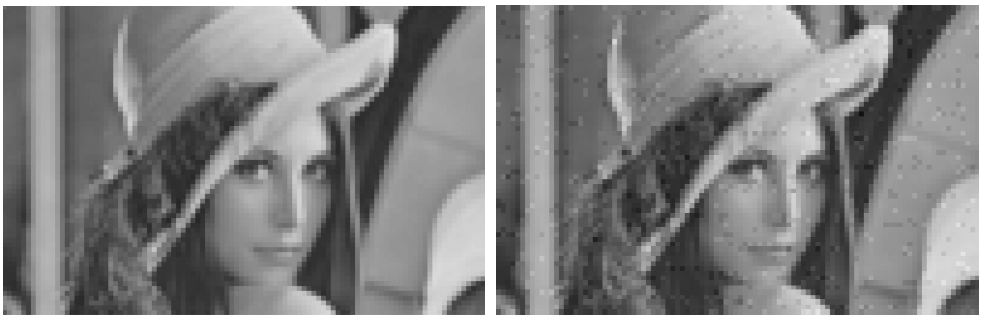
With the operation parameters in Section 5.6.1, BatComm can achieve a throughput of $34.13kbps$, which is over $10\times$ higher than the state-of-the-art solutions. Note that different environments and transmitter settings do not affect the throughput of the communication system.

Environments and Transmitter Settings. We evaluate the BER of BatComm under different environments and transmitter settings, as shown in Figure 5.7(a). It can be observed that the average BER of BatComm under the three settings in the quiet lab are all less than 6% with the standard derivation less than 2%. Figure 5.8 shows two sets of transmitted and received images during the experiments. We find that although the BERs for the two image transmission are 4.7% and 5.1%, the received images are clear for human recognition. This result indicates that such a BER is satisfactory for the acoustic communication. Moreover, all three transmitter settings achieve a comparable performance (i.e., BER difference is less than 1%), indicating that BatComm can use various device as the signal source at the transmitter end. Additionally, the system performance degrades with the increase of background noise levels. However, BatComm can still achieve an average BER of around 8% in the noisy mall environment (with a noise level around $80dB$), which indicates our system is robust to different environments.

Transmitter-Receiver Distances and Environments. We also evaluate the impact of distance between the transmitter and receiver on BatComm. Figure 5.7(b) shows BER of BatComm under different distances and environments. We observe that the BER of our system slightly increases as the distance increases. This is because as the propagation distance increases, SNR of the acoustic signals decreases, which induces more errors in the recorded signals. In the quiet lab environment, the average BER is around 5% with a standard deviation less than 2% under a distance between the transmitter and receiver less than $8cm$, which is a natural and appropriate distance for the short-range communication application. Even for the mall, the BER is 8.3% under the distance less than $5cm$. This result indicates that BatComm can achieve high



(a) Fish (transmitted) (b) Fish (received)



(c) Lena (transmitted) (d) Lena (received)

Figure 5.8: Image comparison between transmitted (i.e., (a), (c)) and received images (i.e., (b), (d)), where BERs are 4.7% and 5.1% (Throughput = 34.13$kbps$).

throughput with an acceptable BER for almost all the short-range applications.

5.6.3 Impact of OFDM Bandwidth

The bandwidth for OFDM multiplexing directly affects the throughput of the acoustic communication system. Also, since the background noises would impact the audible frequency band used in OFDM, the bandwidth, especially the lower bound of bandwidth, affects the BER of acoustic communication system. Hence, we evaluate the performance of BatComm under different bandwidths (i.e., different lower bound of the bandwidth). Figure 5.9(a) shows the throughput and BER of BatComm under different bandwidths. We can see that the throughput increases as the OFDM bandwidth increases, which is consistent with the theoretical analysis. Moreover, it can be observed that the BER also slightly increases as the OFDM bandwidth increases. This is because as the bandwidth increases, the lower bound of bandwidth decreases. Usually, the low-frequency band is easily affected by the background noises (e.g., human speaking). Hence, the wider bandwidth introduces more errors in recorded signals and

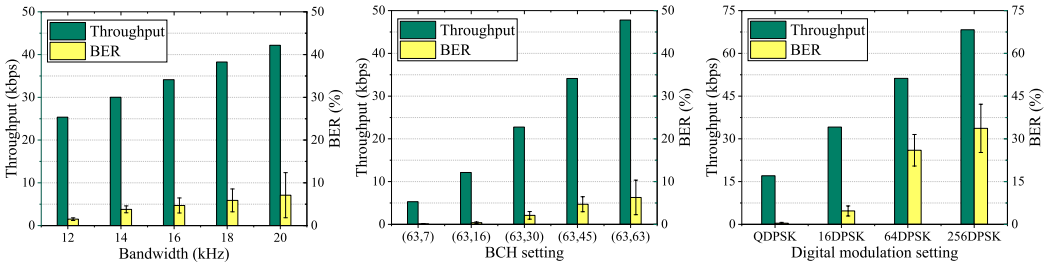


Figure 5.9: Performance of BatComm with different (a) bandwidth, (b) BCH settings, (c) digital modulation settings.

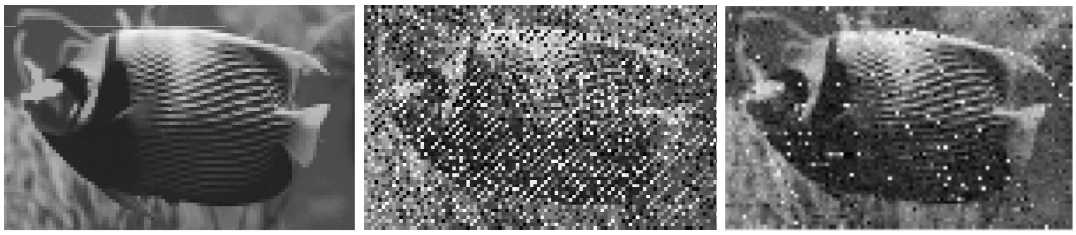
would decrease BER of the system. However, even for the $20kHz$ bandwidth of OFDM operation (4-$24kHz$), the average BER of BatComm achieves 7.1% with a standard deviation of 5.3%, while the throughput of BatComm can achieve $42.18kbps$, which indicates a satisfactory performance.

5.6.4 Impact of BCH Code

In our system, BCH code is used to mitigate the unpredicted errors in recorded signals. However, different BCH settings affect the performance of the acoustic communication system. Particularly, the digital data is encoded with (N, K)-BCH code, where N is the length of encoded data bits, K is the length of original digital data bits. Figure 5.9(b) shows the throughput and BER of BatComm under different BCH settings. We observe that without BCH error correction (i.e., $(63, 63)$ BCH setting), BatComm can achieve a high throughput of $47.49kbps$ and the average BER is only 6.3% with a standard deviation of 4.1%. By adding more BCH coding bits for error correction, both throughput and BER would decrease. To meet specific application requirements, we can use some particular BCH settings to achieve a near-zero BER (e.g., $5.31kbps$ with 0.1% BER; $12.13kbps$ with 0.4% BER).

5.6.5 Impact of Digital Modulation

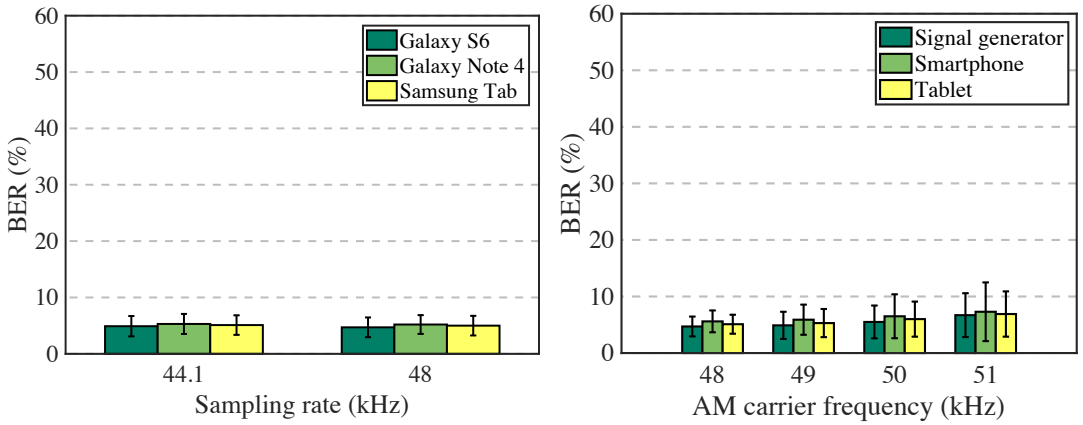
In BatComm, DPSK modulates data bits into phase values for the digital modulation. Figure 5.9(c) shows throughput and BER of BatComm under different digital



(a) (b) (c)

Figure 5.10: Image comparison between transmitted (i.e., (a)) and received images under 64DPSK (i.e., (b)) and 256DPSK (i.e., (c)), respectively. The BERs are 26.0% and 33.7%.

modulation settings. We can see that both the throughput and BER increase when the digital modulation setting changes from QDPSK to 256DPSK. Specifically, for QDPSK and 16DPSK, the average BERs of our system are 1.7% and 4.7% with a standard deviation 0.5% and 1.8% respectively, which are satisfactory for acoustic communication. However, for 64DPSK and 256DPSK, although the throughput is quite high (i.e., larger than 50$kbps$), the achieved BER is not quite satisfactory (i.e., higher than 20%). This is because when the digital modulation setting changes from QDPSK to 256DPSK, the number of data bits modulated by each phase increases, which narrows the difference between adjacent phases. This makes the data bits modulated in phase values highly possible to be misjudged. However, depending on the type of transmitted files, human perception is not always consistent with the BER. Figure 5.10 shows two received images under 64DPSK and 256DPSK respectively. We can find that although the BER under 256DPSK is larger than that under 64DPSK, the received image under 256DPSK is clearer than that under 64DPSK, as shown in Figure 5.10(c) and 5.10(b) respectively. This is because each pixel of the grayscale image contains 8-bit data. In 256DPSK, every 8-bit data is modulated with one phase. Hence, the errors only induce in a pixel itself without the affecting other pixels, which makes the received image clearer for human perception. Based on this observation, BatComm can use specific type of digital modulation scheme for specific file transmission (e.g., 256DPSK for 8-bit grayscale image transmission) to further improve the throughput without significant loss of user experience.



(a) Throughput = 30.01$kbps$ (44.1kHz) 34.13$kbps$ (48kHz)

(b) Throughput = 34.13$kbps$

Figure 5.11: BER of BatComm under different (a) sampling rates and receiver devices (b) AM carrier frequencies.

5.6.6 Impact of Receiver Devices and Sampling Rate

To ensure that BatComm is capable of transmitting data to most mobile devices, we evaluate the performance of our system using three mobile devices (i.e., Galaxy S6, Galaxy Note 4, and a Samsung Tab P7510) as receivers with different sampling rate (i.e., 44.1kHz and 48kHz). According to Nyquist theorem, different sampling rates lead to different bandwidth for communication, which affects the throughput. Specifically, the throughput of BatComm achieves 34.13$kbps$ and 30.01$kbps$ under 48kHz and 44.1kHz sampling rate, respectively. The BERs of our system under different receiver sampling rates are shown in Figure 5.11(a). We observe that the comparable low BERs of these device models can be achieved under different sampling rates. Specifically, the overall BERs for the three receiver models are 5.1% and 5.0% for 48kHz and 44.1kHz sampling rate, respectively. These results demonstrate that BatComm is capable for most commercial off-the-shelf mobile devices.

5.6.7 Impact of AM Carrier Frequency

As mentioned in Section 5.4.3, due to the limited frequency response of ultrasound speaker, the ultrasound carrier frequency in AM affects the performance of acoustic communication system. Since the AM carrier frequency f_c should satisfy $f_c \geq 48kHz$

to avoid the interference between ultrasound signal and automatically demodulated signals, we evaluate the performance of BatComm under different carrier frequencies that larger than $48kHz$. Figure 5.11(b) shows the BER of BatComm under different AM carrier frequencies. We can see that the BER increases as the AM carrier frequency increases. This is because the response of ultrasound speaker decreases as the signal frequency increases, which leads to a poorer SNR of the received signal. This result is consistent with our theoretical analysis. It can be also observed from Figure 5.11(b) that the difference on BERs under different transmitter setups is less than 1%, which exhibits similar results with Figure 5.7(a).

5.7 Discussion

Using Dual Microphones to Improve Performance. In this work, we use the primary microphone of the mobile device (receiver) to pick up ultrasound signals for receiving data bits. However, most commercial mobile devices are equipped with multiple microphones, which are used for stereo recording and noise reduction. Due to different deployment positions of the microphones on the device, the signals recorded by the microphones came from different propagation paths and contain different properties of channel fading effects, background noise levels, etc. It thus has great potential to use multiple microphones to perform noise reduction (e.g., a previous work [115]) and recording-quality calibration, so as to improve the acoustic communication performance. We leave this in our future work.

Long-range Acoustic Communication. BatComm concentrates on using the inaudible acoustic signals for short-range wireless communications, where the transmitter and the receiver should be within several centimeters. Relying on the properties of short-range and inaudibility, we believe it would reduce the privacy risks associated with the acoustic communication. However, the capability of long-distance high-throughput inaudible acoustic communication, which still remains unexplored, would greatly extend its possible applications. Existing work [116] demonstrates a potential long-range inaudible voice attack, which can successfully convey voice command to Amazon Echo and Google Home-like devices within a $25ft$ range. Toward this end, our future work

is interested in providing the high-throughput, inaudible and long-range acoustic communication to extend the application scenarios.

5.8 Conclusion

In this work, the proposed system, BatComm, integrates OFDM and AM techniques with the non-linearity of microphone to achieve the high-throughput and inaudibility for the acoustic communication simultaneously. The combination of OFDM and AM under the non-linearity induces an unrelated residual signal, which leads to significant errors in the communication. To eliminate the residual signal, BatComm modifies the OFDM symbol waveform before AM to counteract the signal and improve the performance. Moreover, to mitigate the interference in practical scenarios, a series of interference reduction techniques (e.g., DPSK digital modulation, pilot-based channel estimation, BCH error correction code, interleaving) are integrated into BatComm for improving the robustness. Extensive experiments demonstrate that BatComm can achieve a throughput as high as $47.49kbps$, which is over $17\times$ higher than the existing acoustic communication solutions.

Chapter 6

Related Work

In this chapter, we present the related research work and compare our approaches with the others. We first review existing studies to perform sleep monitoring and tracking vital signs (i.e., breathing and heart rates) in Section 6.1. We then discuss existing studies on user authentication in Section 6.2. We further review the existing efforts on keystroke recognition in Section 6.3. Finally, in Section 6.4, we study the previous work on short-range acoustic communication.

6.1 Fine-grained Sleep Monitoring Leveraging Off-the-shelf WiFi

Breathing rate, heart rate and statistics of sleep events are important indicators for evaluating one's sleep quality, stress level and various health conditions. In general, the methods used to track such information during sleep can be categorized into four groups: dedicated sensor based, smartphone and wearable sensor based, touch-free sensor based and RF signal based.

Traditional approaches use dedicated sensors to measure vital signs during sleep. For example, Polysomnography (PSG) [4] measures body functions including breathing rate, eye movements (EOG), heart rhythm (ECG) and muscle activity by attaching multiple sensors to a patient. Such systems incur high cost and are usually limited to clinical usage. Recent advances of smartphones and wearable sensors have enabled in-home sleep monitoring by utilizing the built-in accelerometer and microphone [117, 118, 11, 12]. These methods mainly provide coarse-grained monitoring including the detection of body movements, snoring, or regular sleep events, and are not able to monitor breathing rate, which is a critical indication of sleep irregularity such as sleep apnea. They also require users to place smartphones close-by and wear

sensors during sleep. Recent smartphone based approaches [119, 120] can track breathing using either earphone or acoustic FMCW (frequency modulated continuous wave) on smartphones. Moreover, a more direct solution Zephyr [121] uses accelerometer and gyroscope measurements from a standard smartphone held on a person's chest for respiratory rate estimation. However, these solutions cannot provide the information of heart rate and they are also require users to place smartphones close-by even on the users' chest while asleep. Touch-free sensor based solutions either use the sensors attached to the mattress [75] or install a camera to capture the chest movement for breathing rate estimation [122]. These systems however require professional installations and cannot estimate heart rate.

Most related to our work is the RF signal based monitoring mechanisms, such as the use of Doppler radar [5], ultra-wideband [6], Frequency Modulated Continuous Wave (FMCW) radar [8, 7] or Received Signal Strength (RSS) [9, 10, 123] for monitoring the vital signs of breathing rate. In particular, these mechanisms [5, 6, 7, 8] rely on specialized hardware including Universal Software Radio Peripheral (USRP), FMCW radar and Doppler radar. These systems incur high cost and high complexity, making them impractical for large scale deployment. N. Patwari et al. [9, 10] use received signal strength (RSS) measurements (e.g., using 16 frequency channels in IEEE 802.15.4) extracted from wireless sensor nodes to detect the breathing rate. Their approaches require additional wireless network infrastructure and high-density placement of sensor nodes. UbiBreath [123] can track a user's breathing rate and detect apnea using RSS measurements from WiFi-enabled devices. However the coarse-grained channel information of RSS is not able to capture the heart rate. Additionally, Phuc *et al.* use a specialized radar (i.e., iMotion radar [124]) to capture the subtle phase changes of the continuous 2.4 GHz wave signal, which are associated with a user's body movements caused by breathing, to estimate the user's breathing volume [125]. BodyScan [126] can recognize a diverse set of human activities while also estimating the user's breathing rate, by analyzing the Channel State Information (CSI) of the radio signals transmitted/received by two designed wearable devices worn on the user's hip and wrist.

Different from the previous work, our system re-uses existing WiFi network for tracking vital signs of breathing and heart rates concurrently without dedicated/wearable sensors or additional wireless infrastructure. By exploiting fine-grained channel state information provided by off-the-shelf WiFi devices, our system captures both the breathing rate as well as heart rate. Our system thus performs device-free, continuous fine-grained vital signs monitoring without any additional cost. It has the potential to be widely deployed in home and many other non-clinical environments.

6.2 Towards Finger-input Authentication on Ubiquitous Surfaces via Physical Vibration

User authentication becomes a critical step under the growing privacy concerns. Traditional user authentications utilize text-based passwords [18]. To ensure that a user's password cannot be easily guessed, the user has to memorize long strings of random characters, making it inconvenient [20]. Graphical passwords are proposed to ease the memory burden by letting users choose their pre-selected images from random choices of pictures [19, 20, 22] or Cued Clicked Points (CCP) in a sequence of images [127]. Additionally, grid lock pattern based approaches [21, 128] have been widely adopted to keep the user's mobile devices protected. Recent graphical authentication methods can resist shoulder surfing attacks by utilizing the Convex Hull Click Scheme [129] or the eye-gaze version of CCP [130]. However, these strategies eventually perform the authentication based on the knowledge of the passwords (e.g., text-based, image-based and lock pattern-based) and cannot tell whether the password is entered by the legitimate user or not.

To ensure that the secret inputs used for authentication are physically from the legitimate user, biometrics-based schemes (e.g., fingerprints [24], iris patterns [23], retina patterns [25], and face [26]) have been drawn considerable attention recently. However, physiological biometrics are sensitive personal information, which may involve privacy concerns, thus are not widely accepted. To reduce the privacy concerns, a compromised approach is to authenticate users based on their behavioral characteristics, including

unique keystroke dynamic [131], mouse movements [132], and gait patterns [133]. Although these approaches are less sensitive in terms of privacy, they are designed for continuous user verification during the period that the user operates the keyboard, moves a mouse or takes a walk, rather than one-time authentication.

To provide authentication to the emerging smart access systems needed by corporate facilities, apartment buildings, hotel rooms, and smart homes, techniques involving intercom [134], camera [135], access card [136] and fingerprint [24] have been explored. For example, KinWrite [135] uses Kinect, a vision-based platform, to capture the user's 3D handwriting patterns for authentication. These approaches usually involve expensive hardware, complex installation process, and diverse maintenance efforts. Recent studies successfully combine 2D handwriting and behavior features such as corresponding writing pressure, writing speed, and correlation between multiple fingers on touch screens to provide enhanced security [27, 28, 29]. The limitation is that the authentication relies on touch screens, which may suffer from smudge attacks [30] and are not always available in smart access systems. Toward this end, we propose VibWrite that extends the authentication process beyond touch screens to any solid surface leveraging vibration signals. VibWrite will have the authentication capability in a broad array of applications including entry access (e.g., smart building, car doors) and supporting customized services in appliances and devices at smart homes. The authentication process combines password and human physical traits, and supports three types of secret independently including PIN, lock pattern, and gesture input for emerging smart access systems.

6.3 Snooping Keystrokes with mm-level Audio Ranging on a Single Phone

There have been active research efforts in keystroke recognition based on the acoustic emanation or vibration of the keystroke [36, 38, 39, 40, 37, 137, 138, 139, 41]. Acoustic emanation based approaches[36, 38, 39, 40] mainly rely on the observation that each key produces unique acoustic signal when typed, whereas the vibration based methods [137,

138, 139, 41] capture the correlation between the vibration of the keystroke and the location of the keystroke occurred. Vibration based methods all require training efforts to label the keystrokes and usually have less recognition accuracy than that of acoustic emanation based approaches.

In particular, Asonov *et al.* [36] observe that the sound of keystrokes differs slightly from key to key and build a supervised learning based approach to recognize keystrokes. This problem is then revisited by Zhuang *et al.* [38] through adding the language modeling to boost the English text recognition. Berger *et al.* [39] propose a dictionary-based approach leveraging the observation the keystroke sounds correlate to their physical positions on the keyboard. UbiK [37] proposes to locate the location of keystrokes made on solid surfaces leveraging multi-path fading with moderate training efforts. More recently, Zhu *et al.* [40] proposes to utilize microphones on three phones to identify the keystroke of nearby keyboard. The requirements of three phones and the achieved accuracy (i.e., 72.2%) make their approach less feasible for real attack scenarios. Comparing to the above research efforts, our approach is able to achieve high keystroke recognition accuracy by using a single phone without any training.

Another body of related work is smartphone based localization or ranging using acoustic signals [32, 34, 33, 35, 31, 140, 141, 142, 143, 144, 145, 146]. Beepbeep [32] and SwordFight [34] propose phone-to-phone ranging systems that can achieve centimeter level accuracy. Qiu *et al.* [33] develops a 3D continuous localization system for phone-to-phone scenarios with about 10 cm accuracy. The above work however requires application-level communication between two involved phones. Yang *et al.* [35] introduces an acoustic relative-ranging system that classifies phone's position inside the car. This approach relies on customized beep sound for acoustic signal detection. Tarzia *et al.* [31] introduces a technique based on ambient sound fingerprint achieve room-level accuracy. Constandache *et al.* [140] deploys extra acoustic infrastructure inside the building for correcting users' movement traces captured by the accelerometer and compass. In our work, we exploit dual microphones on smartphone to locate the keystroke with high accuracy without customized beep sound or phone-to-phone communication.

6.4 High-throughput and Inaudible Acoustic Communication with Non-linearity of Microphones

Audible Acoustic Communication. Early work [53] evaluates the impact of digital modulation technique, such as amplitude shift keying (ASK) and frequency shift keying (FSK), on human perception and achieves a human-pleasant communication with throughput up to $400bps$. Moreover, Dhwani [55] uses self-jamming coupled with self-interference cancellation at the receiver to provide a secure acoustic communication channel between the devices, which achieves a throughput of $2.4kbps$. Additionally, PriWhisper [54] achieves a secure audible acoustic communication, with around $1kbps$ throughput, using FSK and a jamming technique. However, all of the aforementioned studies directly utilize audible frequency band for communication, which raise disturbance to humans and are also vulnerable to the interference of ambient noises.

Inconspicuous Acoustic Communication. To improve user experience, another body of works embeds the data underlying the daily sound for acoustic communications. For instance, Matsuota et al. [56] embed data bits into a piece of music imperceptibly through OFDM multiplexing, which achieves around $40bps$ throughput. Yun et al. [58] develop a modulated complex lapped transform (MCLT) based approach and achieve around $600bps$ throughput. In addition, Dolphin [57] implements a dual-channel acoustic communication, i.e., transmitting both daily sound and underlying data simultaneously. Specifically, Dolphin applies OFDM to modulate data on high-frequency carriers (i.e., 8-$20kHz$) to embed the data on acoustic signals, and then utilizes the masking effects of human auditory system to transmit the OFDM-modulated data and daily sounds simultaneously without arousing human perception. The throughput achieved in Dolphin is around $500bps$. These approaches hide the data into daily sound, making them not always applicable and still annoying to humans in some cases.

Inaudible Acoustic Communication. To achieve inaudible acoustic communication, existing studies use near-ultrasound band (i.e., 18-$20kHz$). Chirp [50] demonstrates that the near-ultrasound chirp signal can be used for acoustic communication through chirp binary orthogonal keying technique. Moreover, Ka et al. [51] further

design a chirp quaternary orthogonal keying-based approach to realize acoustic communication between TVs and mobile devices leveraging near-ultrasound chirp signals. However, these studies can only achieve a low communication throughput (i.e., 15-16bps). Additionally, U-Wear [52] implements a near-ultrasound acoustic communication for wearable devices. This work can achieve a throughput of 2.76$kbps$ by employing Gaussian minimum-shift keying technique. However, due to the limited frequency band for acoustic communication, these aforementioned studies can only achieve a relatively low throughput, which is unsatisfactory for many emerging applications.

Ultrasound Recording with COTS Microphones. Typically, ultrasound signal can only be recorded by specialized hardware [147] due to the limited sampling rate of the mobile devices' built-in microphones. However, recent studies [148, 149] reveal the non-linearity of microphones, which shows the ultrasound recording capability of these microphones of mobile devices. Specifically, Backdoor [148] demonstrates that ultrasound can be recorded by microphones with the non-linearity, which is applicable for wireless communication. In addition, Dolphin Attack [149] shows the feasibility of launching inaudible-voice-command attacks on speech recognition systems (e.g., Apple Siri, Google Now). In particular, regular voice commands are modulated on ultrasound carriers to achieve inaudibility. Leveraging the non-linearity of microphones, voice commands can be demodulated from the ultrasound signals, and further recognized by the speech recognition systems.

Different from existing approaches, this work effectively uses both OFDM multiplexing and the non-linearity of the mobile device's inbuilt microphone, and proposes a high-throughput inaudible acoustic communication system for general mobile devices. The proposed system significantly improves the communication throughput (i.e., over 17$\times$ higher than the existing solutions) while maintaining a low bit error rate.

Chapter 7

Dissertation Conclusion

In conclusion, this dissertation investigates possible applications and potential security breach with pervasive sensing in future Internet of Things. Specifically, we make the following contributions:

We first showed that the existing WiFi network of IoT can be re-used to capture vital signs of breathing rate and heart rate through using only one AP and a single WiFi-enabled IoT device. Such an approach can also be extended to non-sleep scenarios when the user is stationary. Our proposed system extracts fine-grained channel state information (CSI) from off-the-shelf WiFi device to detect the minute movements and provide accurate breathing and heart rates estimation concurrently. Moreover, we developed algorithms that have the capability to track breathing rates of a single person as well as two-person in bed cases, which cover typical in-home scenarios. The proposed system also have the capability to distinguish different sleep events and track people's sleep postures, which can help people understand their sleep status/quality. Extensive experiments in both lab and two apartments over a three-month period show that our system can achieve comparable or even better performance as compared to existing dedicated sensor based approaches.

We then developed the first vibration-signal-based finger-input authentication system, which can be deployed on any solid surface for smart access and IoT systems (e.g., apartment entrances, car doors, electronic appliances and corporate desks). VibWrite captures intrinsic human physical characteristics presenting at specific location/surface for authentication through extracting unique features (e.g., frequency response and cepstral coefficient) in the frequency domain. The proposed system has the flexibility to support three types of secrets (i.e., PIN, lock pattern, and gesture) to meet different

application requirements by developing new techniques of virtual grid point derivation, featured-based dynamic time warping (DTW) and distribution analysis based on earth mover's distance (EMD). VibWrite is implemented using a single pair of low-cost vibration motor and receiver, which involves minimum hardware installation and maintenance. We performed extensive experiments including authenticating legitimate users and modeling various types of attacks. The results demonstrate that VibWrite can effectively verify legitimate users with over 95% accuracy within two trials and less than 3% false positive rate.

We further demonstrated that a single off-the-shelf phone can recover keystrokes by exploiting mm-level acoustic ranging and fine-grained acoustic features. We developed a training-free approach on a smartphone that does not require a linguistic model, allowing it to recover random keystrokes (e.g., random passwords). We exploited recent mobile audio hardware advances to stretch the limits towards mm-level audio localization accuracy. Moreover, we developed a keystroke snooping framework, which leverages hardware advances (i.e., stereo recording with high sampling rate) of off-the-shelf mobile devices to narrow down possible positions of a keystroke. The framework further exploits the geometry-based information (i.e., TDoA) and unique acoustic signatures of keystrokes to ping-point their positions on a keyboard. We conducted extensive experiments with three kinds of keyboards to show that an off-the-shelf phone with $48kHz$ microphone sampling rate can accurately identify a set of keystrokes with over 85% accuracy. With higher sampling rate (e.g., $192kHz$), the accuracy could be increased to over 94% accuracy. Even for a single keystroke input, our system can achieve 97% accuracy of identifying keystrokes in the top-3 candidate keys with $48kHz$ sampling rate. We believe that these are the first results to raise serious concerns about acoustic password snooping.

Finally, we developed the first high-throughput inaudible acoustic communication system, BatComm, applicable to general mobile and IoT devices. The achieved throughput (i.e., as high as $47.49kbps$) is over $17\times$ higher than existing acoustic communication solutions. In order to maximize the throughput while keeping inaudibility, we theoretically modeled the non-linearity of the device's inbuilt microphone and innovatively used

OFDM multiplexing technique together with the non-linearity model to transmit data over multiple narrow-band channels in an ultrasound frequency band. Relying on the non-linearity of microphones, mobile devices could recover the modulated data on the entire audio frequency band (i.e., $< 24kHz$). We proposed a residual-signal elimination scheme, which elaborately modifies the analog OFDM symbol waveform, to mitigate the effect caused by the unrelated residual signals produced by AM. To achieve robust high-throughput inaudible acoustic communication, the proposed system explores microphone's non-linearity and integrates a series of interference reduction techniques including DPSK modulation, interleaving, BCH codes, pilot-based channel estimation, etc. Extensive experiments in various realistic settings demonstrate that BatComm can achieve a high-throughput and low bit error rate (BER) (e.g., $47.49kbps$ with 6.3% BER; $25.37kbps$ with 1.5% BER and $17.58kbps$ with 0.4% BER) while keeping inaudibility, which outperforms all the state-of-the-art solutions.

References

[1] "Sleep apnea: What is sleep apnea?" *NHLBI: Health Information for the Public. U.S. Department of Health and Human Services*, 2010.

[2] P. X. Braun, C. F. Gmachl, and R. A. Dweik, "Bridging the collaborative gap: Realizing the clinical potential of breath analysis for disease diagnosis and monitoring–tutorial," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3258–3270, 2012.

[3] G. S. Chung, B. H. Choi, K. K. Kim, Y. G. Lim, J. W. Choi, D.-U. Jeong, and K. S. Park, "Rem sleep classification with respiration rates," in *6th International Special Topic Conference on Information Technology Applications in Biomedicine (ITAB)*. IEEE, 2007, pp. 194–197.

[4] C. A. Kushida, M. R. Littner, T. Morgenthaler, C. A. Alessi, D. Bailey, J. Coleman Jr, L. Friedman, M. Hirshkowitz, S. Kapen, M. Kramer *et al.*, "Practice parameters for the indications for polysomnography and related procedures: an update for 2005," *Sleep*, vol. 28, no. 4, pp. 499–521, 2005.

[5] Y. Chen, D. Misra, H. Wang, H.-R. Chuang, and E. Postow, "An x-band microwave life-detection system," *IEEE Transactions on Biomedical Engineering*, vol. 33, no. 7, pp. 697–701, 1986.

[6] J. Salmi and A. F. Molisch, "Propagation parameter estimation, modeling and measurements for ultrawideband mimo radar," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 11, pp. 4257–4267, 2011.

[7] F. Adib, Z. Kabelac, H. Mao, D. Katabi, and R. C. Miller, "Demo: Real-time breath monitoring using wireless signals," in *MobiCom*, 2014.

[8] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person motion tracking via rf body reflections," *MIT technical report*, 2014.

[9] N. Patwari, L. Brewer, Q. Tate, O. Kaltiokallio, and M. Bocca, "Breathfinding: A wireless network that monitors andlocates breathing in a home," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 30–42, 2014.

[10] O. J. Kaltiokallio, H. Yigitler, R. Jäntti, and N. Patwari, "Non-invasive respiration rate monitoring using a single cots tx-rx pair," in *IPSN*, 2014, pp. 59–70.

[11] Fitbit, http://www.fitbit.com/.

[12] Jawbone Up, https://jawbone.com/up.

[13] "Access control market," http://www.marketsandmarkets.com/Market-Reports/access-control-market-164562182.html?gclid=CjwKEAjw9MrIBRCr2LPek5-h8U0SJAD3jfhtuAGYZKVZqHc8ZSphI146GhgExcOxXIts14fKyENFLRoCXG7w_wcB, 2017.

[14] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling, "Distinguishing users with capacitive touch communication," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 197–208.

[15] P. Nguyen, U. Muncuk, A. Ashok, K. R. Chowdhury, M. Gruteser, and T. Vu, "Battery-free identification token for touch sensing devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 109–122.

[16] "How to implement fingerprint authentication in automobiles," http://www.electronicproducts.com/Sensors_and_Transducers/Sensors/How_to_implement_fingerprint_authentication_in_automobiles.aspx, 2017.

[17] "Capacitive sensor," http://www.sensorwiki.org/doku.php/sensors/capacitive, 2017.

[18] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[19] R. Dhamija and A. Perrig, "Deja vu-a user study: Using images for authentication," in *USENIX Security Symposium*, 2000.

[20] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," in *Proceedings of the 21st Annual Computer Security Applications Conference*. IEEE, 2005.

[21] A. T. Timmons and O. D. Altan, "Grid unlock," Feb. 2 2010, uS Patent App. 12/698,321.

[22] A. De Angeli, M. Coutts, L. Coventry, G. I. Johnson, D. Cameron, and M. H. Fischer, "Vip: a visual approach to user authentication," in *Proceedings of the working conference on advanced visual interfaces (ACM AVI)*, 2002, pp. 316–323.

[23] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern recognition*, vol. 43, no. 3, pp. 1016–1026, 2010.

[24] A. Arakala, J. Jeffers, and K. J. Horadam, "Fuzzy extractors for minutiae-based fingerprint authentication," in *International Conference on Biometrics*. Springer, 2007, pp. 760–769.

[25] C. Mariño, M. G. Penedo, M. Penas, M. J. Carreira, and F. Gonzalez, "Personal authentication using digital retinal images," *Pattern Analysis and Applications*, vol. 9, no. 1, pp. 21–33, 2006.

[26] B. Duc, S. Fischer, and J. Bigün, "Face authentication with gabor information on deformable graphs," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 504–516, 1999.

[27] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and i know it's you!: implicit authentication based on touch screen patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 987–996.

[28] Y. Ren, C. Wang, Y. Chen, M. C. Chuah, and J. Yang, "Critical segment based real-time e-signature for securing mobile transactions," in *Proceedings of IEEE Conference on Communications and Network Security (CNS)*, 2015, pp. 7–15.

[29] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos, "User-generated free-form gestures for authentication: Security and memorability," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 176–189.

[30] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," *Woot*, vol. 10, pp. 1–7, 2010.

[31] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proceedings of the 9th international conference on Mobile systems, applications, and services (ACM MobiSys)*, 2011.

[32] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proceedings of the 5th international conference on Embedded networked sensor systems (ACM SenSys)*, 2007.

[33] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, "On the feasibility of real-time phone-to-phone 3d localization," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (ACM SenSys)*, 2011.

[34] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, "Swordfight: enabling a new class of phone-to-phone action games on commodity phones," in *Proceedings of the 10th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, 2012, pp. 1–14.

[35] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Detecting driver phone use leveraging car speakers," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2011.

[36] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *2012 IEEE Symposium on Security and Privacy*, 2004.

[37] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, 2014, pp. 14–27.

[38] L. Zhuang, F. Zhou, and J. Tygar, "Keyboard acoustic emanations revisited," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, 2005, pp. 373–382.

[39] Y. Berger, A. Wool, and A. Yeredor, "Dictionary attacks using keyboard acoustic emanations," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006, pp. 245–254.

[40] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 453–464.

[41] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp) iphone: decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 2011, pp. 551–562.

[42] MEDICI, "Sound-based payments as an inclusive technology for the developing world," [Online]. Available: https://gomedici.com/sound-based-payments-as-an-inclusive-tech- nology-for-the-developing-world/, 2016.

[43] Wired, "When wi-fi won't work, let sound carry your data," [Online]. Available: https://www. wired.com/story/when-wifi-wont-work-let-sound-carry-your-data/, 2018.

[44] ComputerWorld, "The hottest wireless technology is now sound!" [Online]. Available: https://www.computerworld.com/article/2861717/the-hottest-wireless-technology-is-now-sound.html, 2014.

[45] Verifone, "Verifone.com," [Online]. Available: https://www.verifone.com/en/us, 2019.

[46] Paytm, "Paytm.com," [Online]. Available: https://paytm.com/, 2019.

[47] ToneTag, "Cashless & contactless payments solution — homepage — tonetag," [Online]. Available: https://www.tonetag.com/, 2019.

[48] Techcrunch, "Alipay launches sound wave mobile payments system in beijing subway," [Online]. Available: https://techcrunch.com/2013/04/14/alipay-launches-sound-wave-mobile-payments- system-in-beijing-subway/, 2013.

[49] H. Landau, "Sampling, data transmission, and the nyquist rate," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[50] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp signal-based aerial acoustic communication for smart devices," in *Proc. IEEE INFOCOM*, Hong Kong, China, 2015, pp. 2407–2415.

[51] S. Ka, T. H. Kim, J. Y. Ha, S. H. Lim, S. C. Shin, J. W. Choi, C. Kwak, and S. Choi, "Near-ultrasound communication for tv's 2nd screen services," in *Proc. ACM Mobicom*, New York, NY, USA, 2016, pp. 42–54.

[52] G. E. Santagati and T. Melodia, "A software-defined ultrasonic networking framework for wearable devices," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 960–973, 2017.

[53] C. V. Lopes and P. M. Aguiar, "Aerial acoustic communications," in *Proc. IEEE WASPAA*. IEE, 2001, pp. 219–222.

[54] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, and D. Ma, "Priwhisper: Enabling keyless secure acoustic communication for smartphones," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 33–45, 2014.

[55] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: secure peer-to-peer acoustic nfc," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, Hong Kong, China, 2013, pp. 63–74.

[56] H. Matsuoka, Y. Nakashima, and T. Yoshimura, "Acoustic communication system using mobile terminal microphones," *NTT DoCoMo Tech. J*, vol. 8, no. 2, pp. 2–12, 2006.

[57] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: real-time hidden acoustic signal capture with smartphones," in *Proc. ACM Mobicom*, New York, NY, USA, 2016, pp. 29–41.

[58] H. S. Yun, K. Cho, and N. S. Kim, "Acoustic data transmission based on modulated complex lapped transform," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 67–70, 2010.

[59] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *MobiCom*, 2014, pp. 617–628.

[60] "Sleeping Positions To Stay Healthy: The Best And Worst Ways To Sleep During The Night," 2014, http://www.medicaldaily.com/sleeping-positions-stay-healthy-best-and-worst-ways-sleep-during-night-296714.

[61] J. F. Murray, *The normal lung: the basis for diagnosis and treatment of pulmonary disease*. Saunders, 1986.

[62] P. Sebel, M. Stoddart, R. Waldhorn, C. Waldman, and P. Whitfield, *Respiration, the breath of life*. Torstar Books New York, 1985.

[63] "Target heart rates - aha," *Target Heart Rates. American Heart Association*, 2014.

[64] L. Davies and U. Gather, "The identification of multiple outliers," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 782–792, 1993.

[65] R. K. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, 2002.

[66] "NEULOG Respiration Monitor Logger Sensor," http://www.neulog.com/.

[67] W. Xi, J. Zhao, X. Li, K. Zhao, S. Tang, X. Liu, and Z. J., "Electronic frog eye: Counting crowd using wifi," in *INFOCOM*, 2014.

[68] J. T. Bigger, J. L. Fleiss, R. C. Steinman, L. M. Rolnitzky, R. E. Kleiger, and J. N. Rottman, "Frequency domain measures of heart period variability and mortality after myocardial infarction." *Circulation*, vol. 85, no. 1, pp. 164–171, 1992.

[69] A. M. Katz, *Physiology of the Heart*. Lippincott Williams & Wilkins, 2010.

[70] "Sleep position gives personality clue," 2003, http://news.bbc.co.uk/2/hi/health/3112170.stm.

[71] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[72] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[73] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[74] Zephyr Technology, http://zephyranywhere.com/.

[75] SleepIQ, http://bamlabs.com/.

[76] Y.-C. Tung and K. G. Shin, "Expansion of human-phone interface by sensing structure-borne sound propagation," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 277–289.

[77] R. Dong, A. Schopper, T. McDowell, D. Welcome, J. Wu, W. Smutz, C. Warren, and S. Rakheja, "Vibration energy absorption (vea) in human fingers-hand-arm system," *Medical engineering & physics*, vol. 26, no. 6, pp. 483–492, 2004.

[78] J. R. Smith, K. P. Fishkin, B. Jiang, A. Mamishev, M. Philipose, A. D. Rea, S. Roy, and K. Sundara-Rajan, "Rfid-based techniques for human-activity detection," *Communications of the ACM*, vol. 48, no. 9, pp. 39–44, 2005.

[79] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 529–538.

[80] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and mfcc features for speaker recognition," *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 52–55, 2006.

[81] G. A. ten Holt, M. J. Reinders, and E. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Thirteenth annual conference of the Advanced School for Computing and Imaging*, vol. 300, 2007.

[82] Y. Rubner and S. U. C. S. Dept, *Perceptual metrics for image database navigation*, ser. Report STAN-CS-TR. Stanford University, 1999, no. 1621. [Online]. Available: http://books.google.com/books?id=5b1EAQAAIAAJ

[83] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L.-S. Peh, "Low cost crowd counting using audio tones," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012, pp. 155–168.

[84] B. Sklar, *Digital communications*. Prentice Hall NJ, 2001, vol. 2.

[85] "Pearson product moment correlation coefficient," http://en.wikipedia.org/wiki/Pearson-product-moment-correlation-coefficient, 2017.

[86] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at urlhttp://www.csie.ntu.edu.tw/ cjlin/libsvm.

[87] "Geovision," https://www.surveillance-video.com/security-84-fr20200-0010.html, 2017.

[88] "Facepass," https://crownsecurityproducts.com/facepass-facial-recognition-time-clock.html, 2017.

[89] "Cr300," https://amgtime.com/hardware-biometric-fingerprint-reader-tm100, 2017.

[90] T. Wei, S. Wang, A. Zhou, and X. Zhang, "Acoustic eavesdropping through wireless vibrometry," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 130–141.

[91] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: passive recovery of sound from video," *ACM Transactions on Graphics*, 2014.

[92] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "Smartphone based user verification leveraging gait recognition for mobile healthcare systems," in *Proceedings of IEEE SECON*, 2013.

[93] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 363–373.

[94] W.-H. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *Information Systems Security and Privacy (ICISSP), 2015 International Conference on*. IEEE, 2015, pp. 1–11.

[95] J. Zhu, P. Wu, X. Wang, and J. Zhang, "Sensec: Mobile security through passive sensing," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013, pp. 1128–1133.

[96] "Deep learning," http://www.ceva-dsp.com/app/deep-learning/, 2017.

[97] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[98] Y. Obuchi, "Mixture weight optimization for dual-microphone mfcc combination," in *IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2005, pp. 325–330.

[99] D. J. MacKay, *"Information theory, inference, and learning algorithms"*. Cambridge University Press, 2003.

[100] "Snapdragon 800 processors," https://www.qualcomm .com/products/snapdragon/processors/800, 2015.

[101] A. KJ, "Android l update to bring sound quality that will please audiophiles," http://www.ibtimes.co.uk /android-l-update-bring-sound-quality-that-will-please-audiophiles-1454695,2014.

[102] J. Schwarz, D. Klionsky, C. Harrison, P. Dietz, and A. Wilson, "Phone as a pixel: enabling ad-hoc, large-scale displays using mobile devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 2235–2238.

[103] C. Inc, "Comparing mems and electret condenser (ecm) microphones," [Online]. Available: https://www.cui.com/blog/comparing-mems-and-electret-condenser-microphones, 2019.

[104] G. West and J. Macnae, "Physics of the electromagnetic induction exploration method," in *Electromagnetic Methods in Applied Geophysics: Volume 2, Application, Parts A and B*. Society of Exploration Geophysicists, 1991, pp. 5–46.

[105] M. T. Abuelma'atti, "Analysis of the effect of radio frequency interference on the dc performance of bipolar operational amplifiers," *IEEE Transactions on Electromagnetic compatibility*, vol. 45, no. 2, pp. 453–458, 2003.

[106] G. K. Chen and J. J. Whalen, "Comparative rfi performance of bipolar operational amplifiers," in *IEEE International Symposium on Electromagnetic Compatibility*. IEEE, 1981, pp. 1–5.

[107] M. Schechter, S. Fausti, B. Rappaport, and R. Frey, "Age categorization of high-frequency auditory threshold data," *The Journal of the Acoustical Society of America*, vol. 79, no. 3, pp. 767–771, 1986.

[108] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68 – 79, 1960.

[109] P. Monte, R. Tanner, and S. C. C. R. L. University of California, *A Table of Efficient Truncated BCH Codes*, ser. Technical report (University of California, Santa Cruz. Computer Research Laboratory). Computer Research Laboratory, University of California, Santa Cruz, 1988.

[110] IEEE, "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[111] S. Chen and C. Zhu, "Ici and isi analysis and mitigation for ofdm systems with insufficient cyclic prefix in time-varying channels," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 78–83, 2004.

[112] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital communication receivers: synchronization, channel estimation, and signal processing*. John Wiley & Sons, Inc., 1997.

[113] A. Bioacoustics, "Ultrasonic dynamic speaker vifa," [Online]. Available: http://www.avisoft.com/usg/vifa.htm, 2019.

[114] ——, "Portable ultrasonic power amplifier," [Online]. Available: http://www.avisoft.com/usg/pwramp2.htm, 2019.

[115] M. Jeub, C. Herglotz, C. Nelke, C. Beaugeant, and P. Vary, "Noise reduction for dual-microphone mobile phones exploiting power level differences," in *Proc. IEEE ICASSP*. Kyoto, Japan: IEEE, 2012, pp. 1693–1696.

[116] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: the long-range attack and defense," in *Proc. USENIX NSDI*, Boston, MA, USA, 2018, pp. 547–560.

[117] Sleep as Android, https://sites.google.com/site/sleepasandroid/.

[118] T. Hao, G. Xing, and G. Zhou, "isleep: unobtrusive sleep quality monitoring using smartphones," in *Sensys*, 2013.

[119] Y. Ren, C. Wang, J. Yang, and Y. Chen, "Fine-grained sleep monitoring: Hearing your breathing with smartphones," in *INFOCOM*, 2015.

[120] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (ACM Mobisys)*, 2015, pp. 45–57.

[121] H. Aly and M. Youssef, "Zephyr: Ubiquitous accurate multi-sensor fusion-based respiratory rate estimation using smartphones," in *The 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM)*, 2016, pp. 1–9.

[122] J. Penne, C. Schaller, J. Hornegger, and T. Kuwert, "Robust real-time 3d respiratory motion detection using time-of-flight cameras," *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 5, pp. 427–431, 2008.

[123] H. Abdelnasser, K. A. Harras, and M. Youssef, "Ubibreathe: A ubiquitous non-invasive wifi-based breathing estimator," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2015, pp. 277–286.

[124] Y. Li, C. Gu, T. Nikoubin, and C. Li, "Wireless radar devices for smart human-computer interaction," in *Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on*. IEEE, 2014, pp. 65–68.

[125] P. Nguyen, X. Zhang, A. Halbower, and T. Vu, "Continuous and fine-grained breathing volume monitoring from afar using wireless signals," in *The 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM)*. IEEE, 2016, pp. 1–9.

[126] B. Fang, N. D. Lane, M. Zhang, A. Boran, and F. Kawsar, "Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (ACM Mobisys)*, 2016, pp. 97–110.

[127] S. Chiasson, P. C. van Oorschot, and R. Biddle, "Graphical password authentication using cued click points," in *Computer Security–ESORICS 2007*. Springer, 2007, pp. 359–374.

[128] W. Meng, W. Li, L. Jiang, and L. Meng, "On multiple password interference of touch screen patterns and text passwords," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 4818–4822.

[129] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2006, pp. 177–184.

[130] A. Forget, S. Chiasson, and R. Biddle, "Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 1107–1110.

[131] K. Revett, "A bioinformatics based approach to user authentication via keystroke dynamics," *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 7–15, 2009.

[132] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 139–150.

[133] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1961–1974, 2015.

[134] T. Ohshima, T. Morita, T. Tanaka, and N. Yamamoto, "Indoor apparatus of intercom system and method for controlling indoor apparatus," June 22 2006, US Patent App. 11/472,432.

[135] J. Tian, C. Qu, W. Xu, and S. Wang, "Kinwrite: Handwriting-based authentication using kinect." in *NDSS*, 2013.

[136] M. I. Rose and L. W. Hoevel, "Access card for multiple accounts," June 23 1998, US Patent 5,770,843.

[137] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: your finger taps have fingerprints," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, 2012, pp. 323–336.

[138] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications*, 2012, p. 9.

[139] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2012, pp. 113–124.

[140] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: human localization using mobile phones," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking (ACM MobiCom)*, 2010.

[141] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2012, pp. 305–316.

[142] C. Jiang, M. Fahad, Y. Guo, J. Yang, and Y. Chen, "Robot-assisted human indoor localization using the kinect sensor and smartphones," in *Proceedings of the IEEE/RSJ International Conference onIntelligent Robots and Systems (IROS 2014),*. IEEE, 2014, pp. 4083–4089.

[143] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, 2013, pp. 235–248.

[144] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu, "Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones," in *Proceedings IEEE INFOCOM*, 2014, pp. 370–378.

[145] B. Xu, G. Sun, R. Yu, and Z. Yang, "High-accuracy tdoa-based localization without time synchronization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1567–1576, 2013.

[146] M. Uddin and T. Nadeem, "Rf-beep: A light ranging scheme for smart devices," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2013, pp. 114–122.

[147] T. Hosman, M. Yeary, J. K. Antonio, and B. Hobbs, "Multi-tone fsk for ultrasonic communication," in *IEEE Instrumentation & Measurement Technology Conference*, 2010, pp. 1424–1429.

[148] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proc. ACM Mobisys*, Niagara Falls, NY, USA, 2017, pp. 2–14.

[149] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proc. ACM CCS*, Dallas, TX, USA, 2017, pp. 103–117.