THREATS AND OPPORTUNITIES OF MOBILE SENSING TECHNOLOGY IN PERSONAL PRIVACY AND PUBLIC SECURITY

by

CHEN WANG

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 Chen

And approved by

____

____

____

____

New Brunswick, New Jersey

OCTOBER, 2019

ABSTRACT OF THE DISSERTATION

Threats and Opportunities of Mobile Sensing Technology in Personal Privacy and Public Security

By CHEN WANG

Dissertation Director: Yingying Chen

The proliferation of the mobile devices (e.g., smartphones, smartwatches and fitness trackers) has brought great convenience to our daily lives. Mobile users can enjoy the online access anytime and anywhere through WiFi or cellular services, monitor daily activities (e.g., walking steps) via wearable devices, or flexibly access the devices via touch screens and microphones. The pervasive mobile sensors can further benefit the public sector, such as providing real-time data for public transportation, emergency and public safety protection. While the mobile technologies facilitate a wide range of useful applications to the users, an adversary may leverage them to derive the user's sensitive private information. This dissertation focuses on exploring the security threats of the mobile devices given the various embedded sensors. Moreover, we explore to utilize the mobile sensing technologies as opportunities for protecting not only the personal privacy but also the public security.

As the smartphone is the most popular mobile device worldwide, we first investigate to what extent the users' personal information such as social relationships and demographics could be revealed from their smartphones, in particular through the simple signal information of the pervasive Wi-Fi Access Points (AP) without examining any Wi-Fi traffic. We successfully derive the users' activities at daily visited places from the surrounding APs and utilize that as the basis to infer the users' social interactions and individual behaviors. Our approaches capture how closely people interact with each other based on their physical closeness to infer their social relationships and recognize the individual behaviors via their activity characteristics (e.g., activeness and time slots) at their daily visited places to estimate the users' demographics.

Moreover, the increasing popularity of wearable devices motivates us to examine the possible sensitive information leakage from the user's personal wearable devices. We demonstrate a serious security breach of wearable devices in the context of divulging secret information (i.e., key entries) while people are accessing key-based security systems (e.g., ATM machines). We develop a system to show that the motion sensors on a wearable device can be exploited to discriminate mm-level distances and directions of the user's fine-grained hand movements, which enables an adversary to reproduce the hand movement trajectories of the user to recover the secret key entries.

Besides security threats, we also find that mobile technologies bring unique opportunities to protect the personal privacy. We propose to use an off-the-shelf wearable device (e.g., a smartwatch or bracelet) as a secure token to secure the Voice Assistant (VA) systems (e.g., Google Home and Amazon Alexa), which have been shown to be under a high risk of sensitive information leakage in the various acoustic attacks (e.g., impersonation, replay and hidden command attacks). In particular, the proposed system exploits the motion sensors, readily available on most wearables, to describe the voice command in the vibration domain, which is then compared with the audio domain information (recorded by the VA device's microphone) to verify whether the voice command comes from the legitimate user.

Finally, we provide a low-cost and easy-to-scale solution to address the ever-increasing public safety concerns caused by the portable dangerous objects (e.g., lethal weapons, chemical explosives and home-made bombs) in the public places such as museums, stadiums, theme parks and schools. Our proposed detection system utilizes the fine-grained channel state information (CSI) from existing WiFi networks to detect the existence of suspicious objects hidden inside baggage and further identify the dangerous material type of the object without penetrating the user's privacy through physically opening the baggage. Compared to the existing X-ray based object scanning infrastructure, this detection system based on the commodity WiFi could become a game-changer, which significantly reduces the deployment cost and is easy to set up in numerous public venues.

Acknowledgements

I would like to express my deepest appreciation to my Ph.D. advisor and the committee chair, Dr. Yingying Chen. She offered me the golden opportunities to research in the area of mobile computing and cybersecurity and instructed me to solve real-world problems in many projects. She spent huge efforts and countless hours in working with me, giving me guidance, training me and shaping me to be a good researcher. More than that, she also taught me many things for being a good student and a good person as well. Without her guidance and support, I could not have been able to publish many good papers and complete my dissertation. I will never forget her great help to me during my job hunting for a faculty position. I believe that when you leave with a Ph.D. degree, you also take a part of your advisor with you, as your own. As I will start my academic career after graduation, I hope I could be half the advisor you have been to me.

I am also grateful to my thesis defense committee members Dr. Roy Yates, Dr. Hong Man, and Dr. Sheng Wei. Your comments and wisdom help me improve my dissertation a lot. I want to extend my thanks to my colleagues in Data Analysis and Information Security (DAISY) Lab, Yanzhi Ren, Xiaonan Guo, Jian Liu, Yan Wang, Hongbo Liu, Cong Shi, and Yang Bai. It has really been a great time to work with you these years. I will never forget the many late nights when we fought together for paper deadlines.

I would like to thank my parents, Jingxing Wang and Qin Song, for their endless love and continuous support. They have sacrificed so much for me only to help me fulfill my dream of studying abroad. I grew up in one of the poorest provinces in China and I lagged farther behind my peers since my college. My parents have to work very hard to support me, and they always encourage me whenever I feel small and insignificant. Without them, I could not have made this far. I would like to enjoy and cherish every minute I spend with them. I love them. I will never forget my grandfather, Wanghong Song, who was a clever old man. Thank you so much for your accompany.

And finally, last but by no means least, also to everyone in the Daisy Lab, WINLAB and Burchard B200 and my friends. I was so lucky to have you and I enjoyed the time I spent with you.

Dedication

To my mother Qin Song and father Jingxing Wang.

In memory of my grandfather Wanghong Song.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background and Motivation

Mobile Technology, one of the fastest-growing areas of innovation in the world, is a category of technology related to mobility. The earliest form of mobile technology is just a simple cordless mobile device used for making phone calls and sending messages. Now the mobile technology has evolved into multi-tasking mobile devices, such as smartphone, wearable devices and IoT devices that can provide us with anytime and anywhere various services, such as online shopping, internet banking, navigation, fitness tracking and personal assistance. The various embedded sensors such as microphones, motion sensors, touch screen, WiFi, Bluetooth and etc., help to facilitate the various applications by sensing the human activities, improves the device's interaction with users and get connected to the wireless network. The great convenience of mobile devices has significantly contributed to their wide deployment these years. Recent reports show that the number of smartphone users is expected to pass 5 billion in 2019 [27] and that of the connected wearable devices will be over 1.1 billion by 2020 [28].

While we enjoy the great convenience and efficiency by using mobile devices, they also expose many new interfaces to an adversary to attack us and raise many security issues and privacy concerns. Your private information such as credit information, PIN/password, location privacy, social relationships and demographics could attract an adversary's interest and are under a high risk to be disclosed from your mobile devices. For example, an adversary may spoof your identity to fool the authentication systems to directly access this privacy information stored in your mobile devices. Moreover, an adversary could derive your private information from the obtained mobile device sensor data, which are usually considered to have low threats. Therefore, while we leverage mobile technology to improve our daily lives, we also need to prevent an adversary from adversely using such technology. This dissertation aims to explore both the security threats and the opportunities of mobile technologies from two different perspectives, in personal privacy and in public security. We first investigate on the potential privacy leakages from the mobile devices and then explore using the mobile devices to secure the users privacy and protect the public security. In particular, the dissertation covers the following four topics, the personal privacy leakage from smartphones, the personal privacy leakage from wearable devices, the protection of personal privacy on voice assistant using wearable devices and the public security protection using commodity WiFi devices.

Disclosing Personal Privacy from Smartphones. While the mobile users enjoy the anytime anywhere Internet access by connecting their mobile devices through Wi-Fi services, the increasing deployment of access points (APs) have raised a number of privacy concerns. This work explores the potential of smartphone privacy leakage caused by surrounding APs. In particular, we study to what extent the users' personal information such as social relationships and demographics could be revealed leveraging simple signal information from APs without examining the Wi-Fi traffic. Our approach utilizes users' activities at daily visited places derived from the surrounding APs to infer users' social interactions and individual behaviors. Furthermore, we develop two new mechanisms: the *Closeness-based Social Relationships Inference* algorithm captures how closely people interact with each other by evaluating their physical closeness and derives fine-grained social relationships, whereas the *Behavior-based Demographics Inference* method differentiates various individual behaviors via the extracted activity features (e.g., activeness and time slots) at each daily place to reveal users' demographics. Extensive experiments conducted with 21 participants' real daily life including 257 different places in three cities over a 6-month period demonstrate that the simple signal information from surrounding APs have a high potential to reveal people's social relationships and infer demographics with an over 90% accuracy when using our approach.

Revealing Personal Privacy from Wearable Devices. The proliferation of wearable devices, e.g., smartwatches and activity trackers, with embedded sensors has already shown its great potential in monitoring and inferring human daily activities. This work reveals a serious security breach of wearable devices in the context of divulging secret information (i.e., key entries) while people accessing key-based security systems. Existing methods of obtaining such secret information rely on installations of dedicated hardware (e.g., video camera or fake keypad), or training with labeled data from body sensors, which restrict use cases in practical adversary scenarios. In this work, we show that a wearable device can be exploited to discriminate mm-level distances and directions of the user's fine-grained hand movements, which enable attackers to reproduce the trajectories of the user's hand and further to recover the secret key entries. In particular, our system confirms the possibility of using embedded sensors in wearable devices, i.e., accelerometers, gyroscopes, and magnetometers, to derive the moving distance of the user's hand between consecutive key entries regardless of the pose of the hand. Our Backward PIN-Sequence Inference algorithm exploits the inherent physical constraints between key entries to infer the complete user key entry sequence. Extensive experiments are conducted with over 7000 key entry traces collected from 20 adults for key-based security systems (i.e., ATM keypads and regular keyboards) through testing on different kinds of wearables. Results demonstrate that such a technique can achieve 80% accuracy with only one try and more than 90% accuracy with three tries. Moreover, the performance of our system is consistently good even under a low sampling rate and when inferring long PIN sequences. To the best of our knowledge, this is the first technique that reveals personal PINs leveraging wearable devices without the need for labeled training data and contextual information.

Securing Voice Assistants using Wearable Devices.

Due to the open nature of voice input, voice assistant (VA) systems (e.g., Google Home and Amazon Alexa) are under a high risk of sensitive information leakage (e.g., personal schedules and shopping accounts). Though the existing VA systems may employ voice features to identify users, they are still vulnerable to various acoustic attacks (e.g., impersonation, replay and hidden command attacks). In this work, we focus on the security issues of the emerging VA systems and aim to protect the users' highly sensitive information from these attacks. Towards this end, we propose a system, *WearID*, which uses an off-the-shelf wearable device (e.g., a smartwatch or bracelet) as a secure token to verify the user's voice commands to the VA system. In particular, WearID exploits the motion sensors, readily available on most wearables, to describe the voice command in the vibration domain and verify it across two domains (i.e., wearable's motion sensor vs. VA device's microphone).

Our *cross-domain* design (audio vs. vibration) exploits the distinct vibration sensing interface and its short sensing range to sound (e.g., 25 cm) to verify voice commands and shield against the acoustic attacks that cannot be thwarted by using the microphone. However, examining the similarity of two sensing modalities is not trivial. The huge sampling rate gap (e.g., 8000Hz vs. 200Hz) causes the two data types hard to compare and even tiny data noises are magnified during such comparison. Moreover, as not designed for capturing sounds, the motion sensors show distinct response characteristics to sounds in terms of amplitude and frequency. In this work, we investigate the complex relationship between the two sensing modalities and develop a spectrogram-based algorithm to convert the microphone data into low-frequency "motion sensor data" to facilitate cross-domain comparison.Our system then examines the similarity of the voice commands in two domains to verify whether the voice command originates from the legitimate user. We report on extensive experiments to evaluate the WearID system under various audible and inaudible attacks. The results show WearID can verify voice commands with 99.8% accuracy in the normal situation and detect 97% fake voice commands from the various impersonation and replay attacks and hidden voice and ultrasound attacks.

Protecting Public Security Using Commodity Wi-Fi Devices. The growing needs of public safety urgently require scalable and low-cost techniques on detecting dangerous objects (e.g., lethal weapons, homemade-bombs, explosive chemicals) hidden in baggage. Traditional baggage check involves either high manpower for manual examinations or expensive and specialized instruments, such as X-ray and CT. As such, many public places (i.e., museums and schools) that lack of strict security check are exposed to high risk. In this work, we propose to utilize the fine-grained channel state information (CSI) from off-the-shelf WiFi to detect suspicious objects that are suspected to be dangerous (i.e., defined as any metal and liquid object) without penetrating into the user's privacy through physically opening the baggage. Our suspicious object detection system significantly reduces the deployment cost and is easy to set up in public venues. Towards this end, our system is realized by two major components: it first detects the existence of suspicious objects and identifies the dangerous material type based on the reconstructed CSI complex value (including both amplitude and phase information); it then determines the risk level of the object by examining the object's dimension (i.e., liquid volume and metal object's shape) based on the reconstructed CSI complex of the signals reflected by the object. Extensive experiments are conducted with 15 metal and liquid objects and 6 types of bags in a 6-month period. The results show that our system can detect over 95% suspicious objects in different types of bags and successfully identify 90% dangerous material types. In addition, our system can achieve the average errors of 16ml and 0.5cm when estimating the volume of liquid and shape (i.e., width and height) of metal objects, respectively.

1.2 Dissertation Organization

The organization of the dissertation is as follows. In Chapter 2, we investigate to what extent the user's private information such as social relationships and demographics could be revealed from the smartphone. We demonstrate that this sensitive personal privacy could be revealed from the simple signal information of the pervasive WiFi access points, which are periodically scanned by the user's smartphone but is considered to have low threats. Next, Chapter 3 examines the privacy leakage from the user's wearable devices, and we develop a training-free system to reveal the user's personal PIN number from the key-based security systems (e.g., ATM machine) by leveraging the wearable device's motion sensors. After examining the potential security threats in the mobile devices, Chapter 4 and 5 of this dissertation focus on using the mobile technology for enhanced security. In Chapter 4, we propose a framework, which utilizes the user's wearable device as a security token to secure the personal privacy in the popular voice assistant systems (e.g., Google Home and Amazon Alexa). The proposed framework leverages the wearable device's motion sensors and the voice assistant device's microphone to verify the voice commands in both vibration domain and audio domain. Rather than personal privacy, Chapter 5 studies the opportunity of using mobile technologies to protect public security. We develop a low-cost dangerous object detection system, which utilizes the commodity WiFi devices (e.g., laptops) to protect the public security by detecting the suspicious in-baggage objects in public places. Finally, Chapter 6 concludes the dissertation.

Chapter 2

Disclosing Personal Privacy from Smartphones

2.1 Background

Wi-Fi networks are becoming increasingly pervasive, to the point where public Wi-Fi access is readily in place in numerous cities [22]. And the number of public Wi-Fi Access Points (APs) is expected to hit 340 million globally by 2018, resulting in one public Wi-Fi AP for every twenty people worldwide [15]. More commonly, retail stores, offices, universities and homes are usually Wi-Fi enabled for providing high bandwidth and cost-effective connectivity to the Internet for the mobile users. While the mobile users enjoy the anytime anywhere Internet access by connecting their mobile devices (e.g., smartphones) to the Wi-Fi networks, the surrounding APs have raised a number of privacy concerns. For example, mobile users could be located and tracked based on the ubiquitous APs, such as using Google location service [18].

In this work, we study the potential of privacy leakage caused by surrounding APs and explore to what extent the personal information, in particular users' social relationships and demographics, could be derived. Prior work in demographics inference based on Wi-Fi network mainly rely on the context information obtained from passively sniffed users' Wi-Fi traffic[48, 73]. For example, Cheng *et al.* examine users' Internet browsing activities by collecting their in-the-air traffic in public hotspots [48], whereas Huaxin *et al.* infer user demographic information by passively sniffing the Wi-Fi traffic meta-data [73]. These methods need to examine the Wi-Fi traffic and are thus not scalable to large number of users due to the high deployment overhead involved. Existing work in social relationships inference primarily depend on the encounter events detected by either bluetooth [100], Wi-Fi SSID list [47], or GPS locations [39]. These approaches can only perform coarse-grained social relationships inference by examining whether users have interactions or not instead of studying users' behaviors and how closely they interact with each other. They can neither provide fine-grained social relationships (such as advisor-student, colleagues, friends, husband-wife, neighbors) nor identify specific role of the user in the relationship.

It is known that GPS, motion sensors and contact lists on mobile devices can exhibit privacy, but how much a user's privacy could be leaked from the ubiquitous access points is unclear. In this work, we demonstrate that by examining the simple signal features of the surrounding APs it is possible to infer users' fine-grained social relationships and demographics without sniffing any Wi-Fi traffic. Specifically, the availability of surrounding Wi-Fi APs is periodically scanned by mobile devices because of their default systems purpose to optimize network service via continuously seeking better Wi-Fi signals and remembered APs [25, 11] and accessing such information only requires a common permission, which is considered with low risk [97]. Signal features such as the time-series of BSSIDs (i.e. MAC addresses) and Received Signal Strength (RSS) are then extracted from these scanned APs and analyzed to derive users' activities at daily visited places. Our system exploits the rich information of users' daily interactions and behaviors embedded in these derived activities and discloses fine-grained social relationships (including advisor-student, supervisor-employee, colleagues, friends, husband-wife and neighbors) as well as demographic information (such as occupation, gender, religion, marital status).

Our approach of using simple signal features of APs can be easily applied to a large number of users. For example, advertisers or third party companies could mine users' personal information for targeted advertising or recommending services. However, such an approach could cause significant privacy leakage if it is utilized by advertisers with aggressive business attempts, who could simply publish free apps to users while these free apps actively collect users' surrounding AP information and send back to the server to derive users' social relationships and demographics.

In particular, we describe people's daily places in three dimensions (i.e. temporal, spatial and contextual) to infer people's *activities* at each place. For users performing activities at the same place, we calculate *physical closeness* of the users (e.g., whether staying at the same room, adjacent rooms or inside the same building) and extract users' activeness (e.g., walking around or sitting) together with other features (e.g., time slots and duration) to characterize their activities at daily places. We then develop *Closeness-based Social Relationships Inference* algorithm to capture where, when and how closely people interact to derive fine-grained social relationships. We design *Behavior-based Demographics Inference* method to capture individual behavior based on users' various daily activities to reveal demographic information including occupation, gender, religion and marriage. We conduct extensive experiments with 21 participants carrying their smartphones to collect surrounding Wi-Fi AP information in their real daily life across three cities over 6 months and study to what extent we can derive these participants' social relationships and demographic information.

The primary contributions of this work are as follows:

- We demonstrate that simple signal information (e.g., time-series of MAC addresses and RSS) from users' surrounding Wi-Fi APs can reveal private information including both social relationships and demographics.
- We develop statistical methods to detect and characterize users' daily visited places based on the AP signal information and further infer the context of daily places by deriving users' activity features (e.g., activeness, time slots and duration)
- We design closeness-based social relationships inference algorithm to analyze when, where and how closely users interact with each other and reveal users' detailed social relationships (e.g., advisor-student, supervisor-employee, colleagues, friends, husband-wife, customer relationship and neighbors).
- We further abstract people's various behaviors (e.g., home, working and leisure behaviors) to infer their demographic information such as occupation, gender, religion, and marital status.
- We show with experimental study of 21 participants that by using our system one can achieve over 91% accuracy of inferring social relationships and over 90% accuracy of deriving demographic information via examining the simple signal features from surrounding APs.

2.2 Related Work

In this work, we aim to understand the privacy leakage of smartphone users, in particular discovering users' social relationships and demographics, by analyzing only the availability of surrounding APs without sniffing any Wi-Fi traffic. Obtaining such information requires limited permission other than turning on GPS or accessing to contact lists. Our work is related to the research efforts in using various information collected from Wi-Fi network and/or smartphone for meaningful places extraction [67, 68, 46, 52], social relationships inference [127, 51, 47, 100, 60], and demographics derivation [48, 73, 102].

As the contextual location can be used for learning the person's interest and providing content-aware applications, there have been active studies on extracting contextual meaning of the locations people visited. For example, Kang *et al.* design a cluster-based method to extract meaningful places from traces of location coordinates collected from GPS and Wi-Fi based indoor location system [67]. Kim *et al.* propose SensLoc that utilizes a combination of acceleration, Wi-Fi, and GPS sensors to find semantic places, detect user movements, and track travel paths [68]. These existing methods however only focus on individual users' visited locations without analyzing the interactions between them. Besides, the obtained meaningful places may be not sufficient to infer the higher level personal information, such as fine-grained social relationship and demographics, due to the lack of information about the users' daily behaviors and social interactions.

Information in Wi-Fi networks and smartphones have been used in literature to infer users' social relationships. For example, Wiese *et. al* [127] use the smartphone contact list to mine personal relationships. Moreover, the similarity of smartphones' SSID lists is used to reveal users' social relationships [47]. These methods can only derive coarse-grained social relationships without analyzing the behaviors and interactions among people. Vicinity detection via Bluetooth or Wi-Fi signals opens opportunities for social interaction analysis and the strength of friendship ties can be inferred from such wireless signals [100, 60]. However, these vicinity detection methods only consider the relative interaction between people without interaction context (e.g., place context and behaviors). They are unable to differentiate the specific type of various social relationships, such as family members and friends. Our previous work focuses on extracting the social relationship from smartphone App leaked information such as GPS location, IMEI and network location[124]. It could only derive the social relationships in a coarse-grained manner. In this work, we take a closer look and study the privacy leakage just from the surrounding APs and derive people's activities and various closeness levels of social interactions for inferring detailed relationships demographic information.

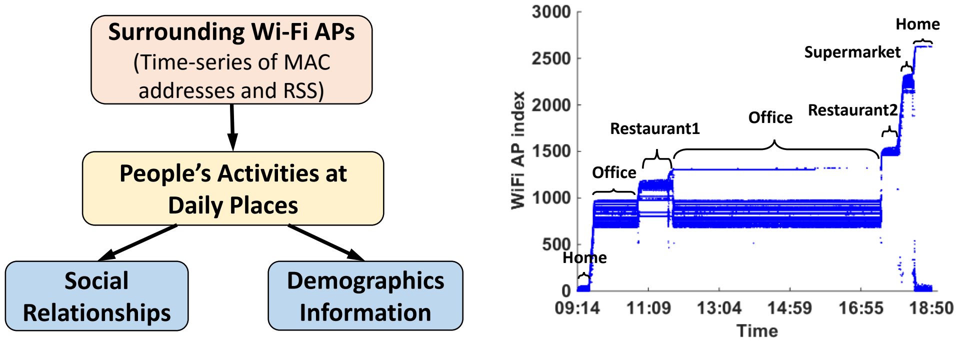
More recently, Wi-Fi traffic monitoring and smartphone Apps have been used to infer users' demographic information. For example, Cheng *et al.* examine the user's Internet browsing activities (e.g., domain name querying, web browsing) by collecting their Wi-Fi traffic in public hotspots [48]. They are able to reveal the travelers' identities, locations or social privacy. Huaxin *et al.* design an approach to infer user demographic information by sniffing the Wi-Fi traffic meta-data [73]. Seneviratne *et al.* design a system to predict various user traits by analyzing the snapshot of installed Apps [102]. Different from the above work, we study the capability of examining the simple signal information of surrounding APs to derive demographic information without sniffing any Wi-Fi traffic or examining the installed Apps.

2.3 System Design

2.3.1 Preliminaries

Environment-Behavior research reveals that an individual's activities such as work-related, household and leisure activities are related to the places they visit [99]. And such activities at daily visited places can be analyzed and mined to infer users' personal information such as social relationships and demographics [80]. Thus by leveraging the users' activities at daily places as a bridge, we could start from the non-contextual surrounding AP information to infer users' social relationships and demographics. This connection is depicted in Figure 5.1(a). The surrounding Wi-Fi APs reflect users' surrounding wireless environments, which can be utilized to determine users' daily visited places and activities. The *daily places* in our work refer to the abstract locations that users visit in their daily lives, such as home, workplace, restaurants, stores and churches. By analyzing users' activities at daily places, we could derive the social interactions between users and abstract individual's behavior. Such information is then further utilized to mine users' social relationships and demographics. Note that contrary to the existing work in social relationships and demographics inference, we only utilize the availability of surrounding APs' simple signal information without requiring to sniff any Wi-Fi traffic contents.

To study how the surrounding APs can be utilized to detect a user's daily places and activities, we conduct preliminary experiments by recording the APs on the user's smartphone at the regular rate of one scan per 15 seconds, because a Wi-Fi device usually scans every 5 - 15 seconds for providing the user non-interrupted Wi-Fi connection to cope with the user's place change [105, 10]. Figure 5.1(b) shows the recorded time-series of a user's surrounding APs (differentiated by BSSIDs) for one day, as well as the groundtruth of visited places. As the AP index is assigned to each unique AP in sequence, the later observed AP has larger index. The observation is that the detected AP lists have large overlaps when the user stays at the same place, while the AP lists are distinct when the user moves to a different daily place. This suggests that we may utilize the changes of the observed AP list to detect the user's daily visited places as well as the entrance/departure time and the staying duration. Moreover, the user's activities at daily places (e.g., the user's mobility at work and during leisure time) can be derived to reflect individual demographics. Furthermore, we observe that the same place or the places in the neighborhoods may share some APs (e.g., office and restaurant 1). Their physical closeness may be obtained by checking how many surrounding APs they share, which is useful for analyzing social interactions.



(a) Connection from surrounding APs to social relationships & demographics.

(b) Illustration of observed APs by a user's smartphone in one day.

Figure 2.1: Preliminary studies.

2.3.2 Challenges

Robust Daily Places and Activity Detection Using APs. Lacking the pre-knowledge of AP deployment, the accurate and robust detection of daily places and activities from ubiquitous APs is challenging. And the ubiquitous unstable and mobile APs even add to the difficulties. Additionally, the daily places need to be abstracted with sufficient spatial resolution (e.g., differentiating rooms and floors) for further deriving users' mobility and their physical closeness during interaction.

Determining the Context of Daily Places. Deriving the context of a user's daily visited places from the non-contextual AP signal information is challenging. Moreover, a place may exhibit different contexts to different users. For example, stores are leisure places to most people but the workplace to the store staff. This requires us to search for the deep implication behind the individual's activities at the place instead of relying on traditional place context based on the place function.

Fine-grained Social Relationships Inference. Fine-grained relationships inference needs the information on not only who have interactions but also on how closely they interact. Our systems needs to have the capability to define multiple closenesses between users. Furthermore, specifying the role of each user in a relationship (e.g., husband or wife) may needs the assistance from demographic information (e.g., gender).

Demography Inference without Context. Inferring a user's demographics with non-contextual simple signal information of surrounding APs is challenging. Different from the previous work relying on the content obtained from monitoring the Wi-Fi traffic, our system explores the possibility to abstract users' behaviors based on their various activities at daily

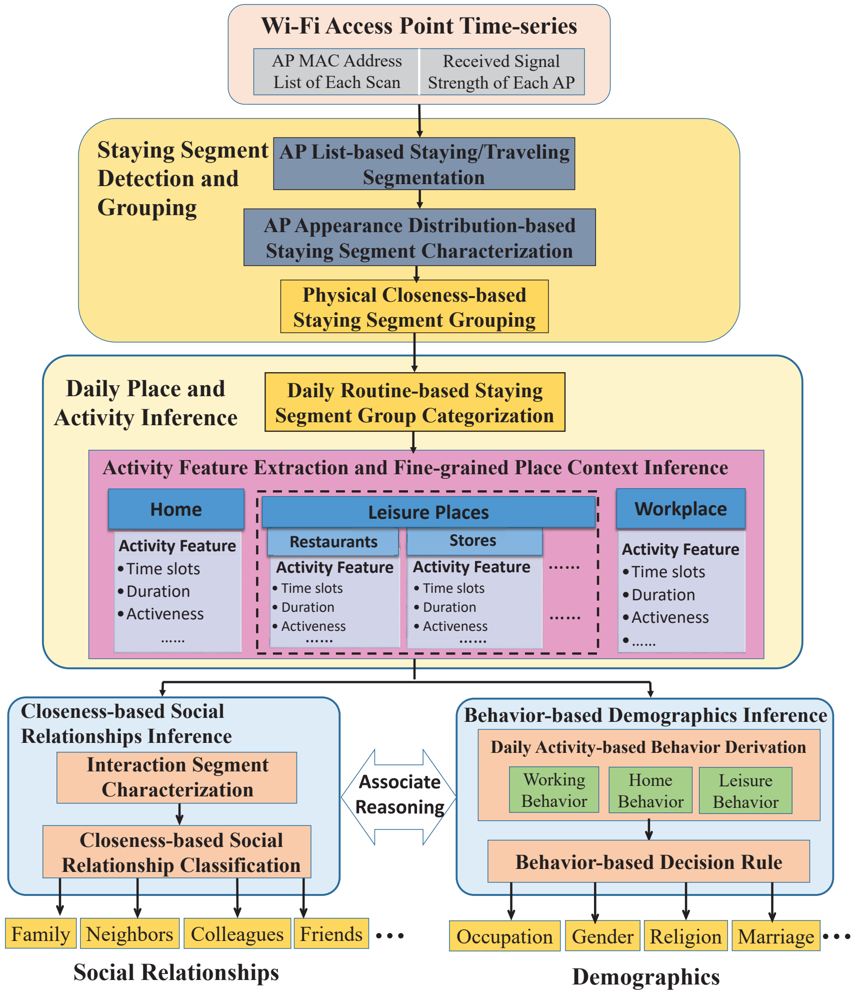

Wi-Fi Access Point Time-series
AP MAC Address List of Each Scan
Received Signal Strength of Each AP
Staying Segment Detection and Grouping
AP List-based Staying/Traveling Segmentation
AP Appearance Distribution-based Staying Segment Characterization
Physical Closeness-based Staying Segment Grouping
Daily Place and Activity Inference
Daily Routine-based Staying Segment Group Categorization
Activity Feature Extraction and Fine-grained Place Context Inference
Home
Activity Feature
•Time slots
•Duration
•Activeness
......
Leisure Places
Restaurants
Activity Feature
• Time slots
• Duration
• Activeness
......
Stores
Activity Feature
• Time slots
• Duration
• Activeness
......
......
......
Workplace
Activity Feature
•Time slots
•Duration
•Activeness
•......
Closeness-based Social Relationships Inference
Interaction Segment Characterization
Closeness-based Social Relationship Classification
Associate Reasoning
Behavior-based Demographics Inference
Daily Activity-based Behavior Derivation
Working Behavior
Home Behavior
Leisure Behavior
Behavior-based Decision Rule
Family
Neighbors
Colleagues
Friends
•••
Social Relationships
Occupation
Gender
Religion
Marriage
•••
Demographics

Figure 2.2: Wi-Fi AP distribution-based social relationships and demographics inference framework.

places for demographic inference.

2.3.3 System Overview

The basic idea of our system is to analyze users' activities at daily routine-based places that are derived from users' surrounding APs for fine-grained social relationships and demographics inference. The proposed system takes as inputs the information of users' surrounding APs perceived by their smartphones at each scan, including the list of AP MAC addresses and RSS, to infer fine-grained social relationships and demographics. Figure 3.4 presents our system flow.

First, the *Staying Segment Detection and Grouping* component detects and characterizes users' daily visited places in three steps. *AP List-based Staying/Traveling Segmentation* analyzes the overlap of the AP lists over consecutive scans and divides the time-series into staying and traveling periods. *Staying Segment Characterization* estimates the significance of each surrounding AP by calculating its appearance rate within the staying segment. It then categorizes the APs by their significance to describe the spatial information of each staying segment. The spatially close-by staying segments are then grouped together as one unique place by using *Closeness-based Staying Segment Grouping.*

The next component is to derive the activities at daily places which is an important building block of social relationships and demographics inference. It is carried out by using *Daily Place and Activity Inference*, which involves *Daily Routine-based Staying Segment Group Categorization* and *Daily Activity Feature Extraction and Fine-grained Place Context Inference. Daily Routine-based Staying Segment Categorization* classifies the grouped staying segments (i.e. unique places) into three contextual categories (i.e. home, leisure and workplace) based on people's daily routines. At last, *Daily Activity Feature Extraction and Fine-grained Place Context Inference* derives people's activity features including the staying time slots, duration and activeness and assigns detailed contextual information to these places by leveraging the derived activity features and geo-information, such as restaurants or stores in leisure places, campus or office buildings in workplaces.

Finally, our system infers users' social relationships and demographics based on the derived activities at daily places. In particular, it first calculates the physical closenesses of the interactions between users. It then uses *Interaction Segment Characterization* and *Closeness-based Social Relationships Classification* to infer when, where and how closely people interact with each other for inferring their possible relationships such as family, neighbors, colleagues, and friends. To derive a user's demographics, *Behavior-based Demographics Inference* applies *Daily Activity-based Behavior Derivation* to abstract people's various behaviors including working behaviors, home behaviors and leisure behaviors, based on the activities at daily places. It then utilizes *Behavior-based Decision Rule* to infer users' demographic information (e.g., occupation, gender, marriage and religion) based on the behavior abstraction. At last, the *Associate Reasoning* can be applied to social relationships and demographics to improve the accuracy of inference results, such as identifying the specific role of the user in a relationship (e.g., husband-wife and advisor-student).

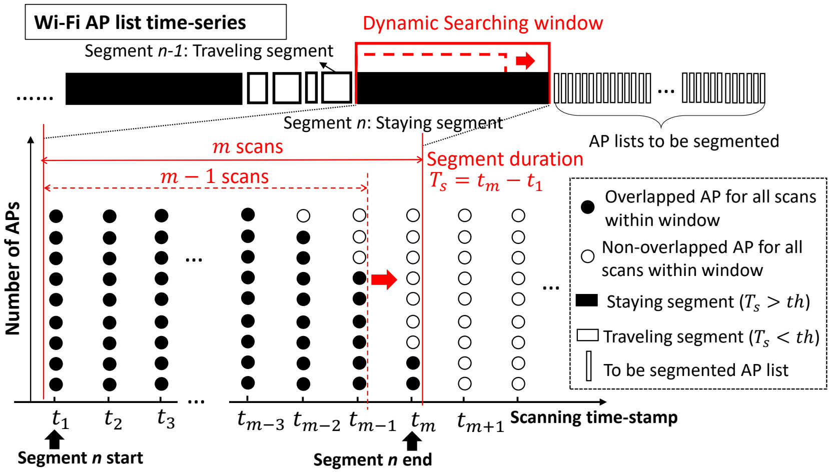

Wi-Fi AP list time-series
Dynamic Searching window
Segment n-1: Traveling segment
......
Segment n: Staying segment
AP lists to be segmented
m scans
m − 1 scans
Segment duration
$T_s = t_m - t_1$
Number of APs
Overlapped AP for all scans within window
Non-overlapped AP for all scans within window
Staying segment ($T_s > th$)
Traveling segment ($T_s < th$)
To be segmented AP list
t_1 t_2 t_3 t_{m-3} t_{m-2} t_{m-1} t_m t_{m+1}
Scanning time-stamp
Segment n start
Segment n end

Figure 2.3: Staying/traveling segmentation leveraging dynamic searching windows to analyze the overlapped AP lists over consecutive scans.

2.4 Staying Segment Group Detection and Characterization

2.4.1 AP List-based Staying/Traveling Segmentation

As observed in the preliminary study of Figure 5.1(b), the discovered AP BSSID lists of consecutive scans have large overlaps when the user stays at the same place, while the similarity of the AP lists is rapidly diminished when the user moves to a different place. We thus take the advantage of the AP list similarity (i.e. BSSID list similarity) in consecutive scans to detect the staying and traveling segments. We define *staying segment* as the Wi-Fi AP-list time-series segment that captures the temporal and spatial information when the user stays at a location. And we analyze the overlap of the AP lists within a dynamic searching window of consecutive scans to perform staying segmentation.

In particular, Figure 2.3 illustrates the proposed AP List-based Staying/Traveling Segmentation in identifying the staying segment n. The dynamic searching window starts at t_1 and iteratively expands to the next scan. In each iteration, we analyze the overlapped APs of all the scans within the searching window. The number of solid dots at each scanning time $t_i (i = 1, 2, \dots)$ indicates the number of overlapped APs that are found within the window from t_1 to t_i. When the searching window iteratively expands to the next scan, the number of overlapped APs may decrease. When no overlapped AP is found in the expanded searching window (e.g., the window from t_1 to t_m), such searching window is identified as one possible staying segment. We note that because it may take several scans to travel out of an AP's range, this approach can detect short staying segments even when the user is traveling. We next check whether the segment duration $T_s = t_m - t_1$ is greater than a threshold τ (e.g., $\tau = 6$ minutes) to further confirm valid staying segments and filter out the false staying segments. Meanwhile, the user's entrance/departure time and corresponding staying duration could also be obtained.

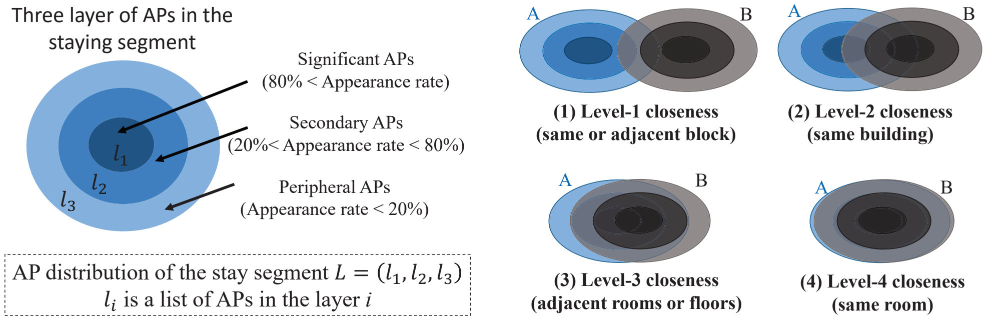
2.4.2 AP Appearance Rate Distribution-based Staying Segment Characterization

We next characterize the visited places by deriving Wi-Fi AP appearance distribution in the detected staying segments. The discovered AP BSSID list can be used to describe the wireless environment of the user in the staying segment. However, not all the APs have the same significance for characterizing the spatial information. Some APs may appear only in a few scans due to weak Wi-Fi signals, while others are more stable and appear almost in every scan. We calculate the *appearance rate* of each discovered AP to represent its significance, and then classify the APs into different categories based on their significance. In particular, the *appearance rate* of an AP is defined as $R = \frac{N_a}{N}$, where N_a is the appearance number of this AP and N is the total number of scans in the detected staying segment. The appearance rates together with BSSIDs of the discovered APs are used to characterize the spatial information of the staying segment, which has the potential to both differentiate places with good resolution but also measure people's physical closeness.

We empirically divide the APs of a staying segment into three layers $l_i, i = 1, 2, 3$ (i.e. lists of significant APs, secondary APs and peripheral APs) according to their appearance rate. As shown in Figure 2.4(a), the significant APs are those with appearance rate larger than 80%, the peripheral APs are the ones with the appearance rate less than 20%, and the rest of APs are secondary APs. Then the spatial information of the staying segment can be characterized by *AP set vector* $L = (l_1, l_2, l_3)$, which can tolerate the noise generated by the unstable APs, mobile APs or even missing AP scans.

2.4.3 Estimating Physical Closeness between Staying Segments

Measuring the physical closeness between different users' staying segments can capture how closely people interact with each other. It can also be used to group the same user's staying segments that are close to each other as one place. In particular, we leverage the AP set vector to measure the physical closeness between staying segments. Given two staying segments A and



(a) Appearance rates and significance of the APs in a staying segment.

(b) Four kinds of closeness between staying segments A and B.

Figure 2.4: AP appearance rate distribution-based staying segment characterization.

B and their AP set vectors L_A and L_B, we calculate the *closeness matrix* M as follows:

$$M = L_A^{-1} L_B = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \tag{2.1}$$

where r_{ij} is the overlapping rate between subsets l_{Ai} and l_{Bi} of AP set vectors L_A and L_B, respectively. The overlapping rate r_{ij} can be obtained by

$$r_{ij} = \frac{OverlapApNum(l_{Ai}, l_{Bj})}{min(Num(l_{Ai}), Num(l_{Bj}))}, i, j = 1, 2, 3. \tag{2.2}$$

Based on the statistical analysis with 431 staying segments collected from 167 places in 3 cities, we empirically quantify the physical closeness expressed by the closeness matrix M into five levels:

$$\begin{cases} C_0 = \left\{M : \sum_{i,j=1}^{3} r_{ij} = 0\right\}; & (Completely \quad separated) \\ C_1 = \left\{M : r_{33} > 0 \; and \; \sum_{i,j=1}^{3} r_{ij} - r_{33} = 0\right\}; & (Same \quad street \quad block) \\ C_2 = \left\{M : \sum_{i,j=1}^{3} r_{ij} - r_{33} - r_{11} > 0 \; and \; r_{11} = 0\right\}; & (Same \quad building) \\ C_3 = \{M : 0 < r_{11} < 0.6\}; & (Adjacent \quad rooms) \\ C_4 = \{M : r_{11} \geq 0.6\}, & (Same \quad room) \end{cases} \tag{2.3}$$

where C_1, C_2, C_3, C_4 are four mutually exclusive closeness sets with increasing closeness level as shown in Figure 2.4(b), representing the same street block, the same building, the adjacent rooms and the same room respectively. $C_0 = \overline{C_1 \cup C_2 \cup C_3 \cup C_4}$ means two staying segments are completely separated. We use level-i closeness to express closeness in set C_i.

2.4.4 Physical Closeness-based Staying Segments Grouping

We note that the same user's multiple staying segments may correspond to the same place as the user may pay multiple revisits. We thus combine these staying segments together by checking whether there is level-4 closeness between them and keep all the time slots. The grouped staying segments represent non-redundant places visited by the user and contains the user's activities. We can then characterize the user's activities at each unique place.

2.5 Daily Place and Activity Inference

In this section, we explore to what extent we can understand the contextual information of the places visited by people and their activities at the places, which facilitate the social relationships and demographics inference.

2.5.1 Daily Routine-based Place Inference

Compared to the physical information (e.g., longitude and latitude), the contextual information (e.g., name and type) of a place contains more meaningful information related to people's social relationships and demographics. To obtain such information, we exploit the simple signal information of surrounding APs (i.e., BSSIDs and RSSs) that is readily available in most mobile devices, to determine the daily place meanings of staying segments based on people's daily routines.

Daily Routine-based Places

Recent reports [13, 12] indicate that people's daily routines mainly consist of three categories of activities: 1) working and work-related activities (*working activities*); 2) sleeping and household activities (*home activities*); and 3) *leisure activities*. Based on the understanding of people's daily routines, we define three categories of *daily routine-based places*, namely *Workplace* (e.g., office buildings and universities), *Home*, and *Leisure Place* (e.g., stores, restaurants, and churches), to describe contextual information of the places. Different from categorizing daily places based on their generic nature [70], our daily routine-based categorization of daily places reflects the meaning of a place to a person instead of its function, which may vary from person to person to better describe the context of a place for every individual. For example, the same

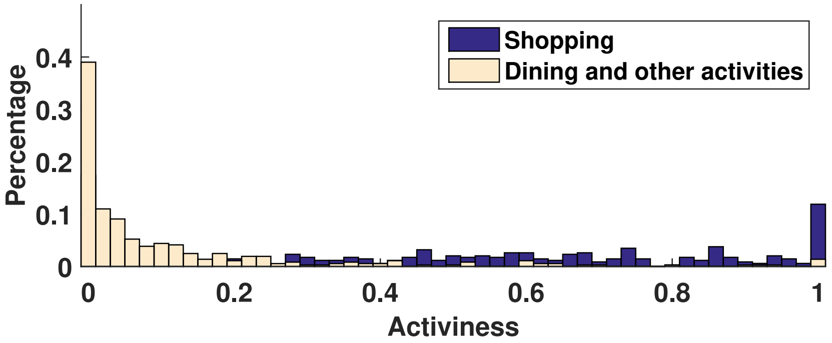


Figure 2.5: Distribution of activeness score computed from each AP during staying segments when people are shopping or dinning.

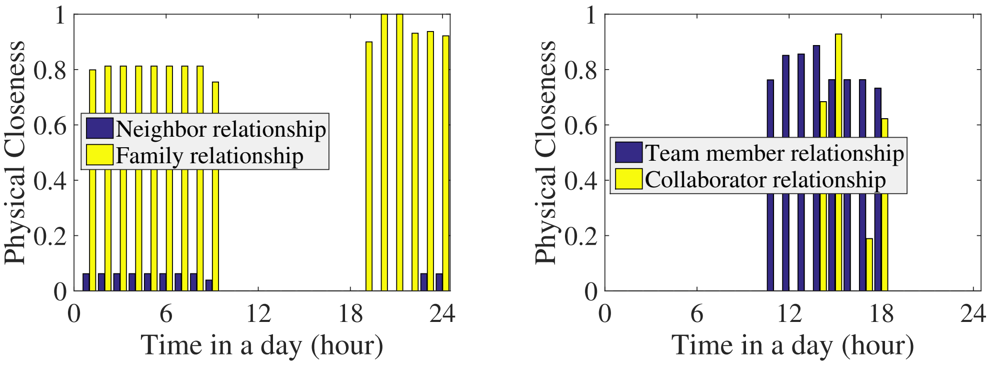
restaurant could be a workplace for waiters and waitresses, but it is a leisure place for customers. This advantage enables inferring the fine-grained social relationships and demographics.

Staying Segment Categorization based on Daily Routines

Next, we determine the contextual information of a place (i.e. staying segment) by categorizing it into one of the three defined daily routine-based places. The basic idea is to examine *common time spans* of the staying segments in a day with the daily routines of working and home activities, respectively. Whichever staying segment results in the longest overlapped time with the daily routine of working or home activities will be labeled as containing the Workplace or Home. The rest of staying segments are determined as containing the Leisure Places. Since people may move between different rooms for work-related activities, after determining the Workplace, we further combine the staying segments that have at least level-1 closeness with the staying segments of Workplace together to represent the whole working area. The common time spans are chosen corresponding to the majority people's daily routines from the reports [13, 12]: working activities - $8:00$AM$\sim 4:00$PM; home activities - $7:00$PM$\sim 6:00$AM; leisure activities - rest free hours of a day.

Fine-grained Place Context Inference

Our system is designed to derive more fine-grained place contexts (e.g. restaurants or stores in the Leisure Places and universities or office buildings in the Workplace) by leveraging Geo-information, activity features of the places and the SSID context of user associated AP. We find that the APs' BSSIDs (MAC addresses) in a staying segment generate fine-grained place contexts through certain web-based services (e.g., Google Map Geolocation API [20], Google



(a) Spatial closeness difference.

(b) Temporal closeness difference.

Figure 2.6: Illustration of social relationships classification derived from temporal and spatial closeness based on one day's data.

Place API [19] and unwired labs Location API [24]). However, the place contexts obtained from the Geo-information is sometimes not unique especially in a crowded business area. Therefore, to refine the place contexts from the Geo-information, we further examine the activity features in the staying segment based on the decision rules, made from people's general time use pattern [23] and the basic knowledge of activeness at various place contexts. Moreover, if the user is associated with an AP, the semantic meaning of the AP SSID can be utilized as assistance, if available, to identify detailed contexts (e.g. company names) of the place.

2.5.2 Activity Feature Extraction

We determine three activity features (i.e., including *activeness*, *visiting time slots* and *staying duration*) that can capture the users' mobilities and the differences between activities at the daily routine-based places. *Activeness* (i.e. active or static) describes the person's status at a place, e.g., shopping in a store is active while dinning in a restaurant is static. *Visiting time slots*, including the person's one or multiple entrance/departure time at a daily routine-based place, captures the person's specific pattern of visiting the place, e.g., faculties may leave office several times in one day for teaching, conference, lunch *et al.* *Staying duration* captures the time nature of the activities such as buying coffee for 10 minutes or doing hair cut for one hour. We note that all the other activity features, except the activeness, can be easily obtained by examining the temporal information of the staying segments. Therefore, we discuss how to derive the activeness for each staying segment in detail.

Activeness Estimation. We devise a unique activeness estimation approach to determine the activeness of the user at a place by only utilizing the RSS of APs observed in the staying

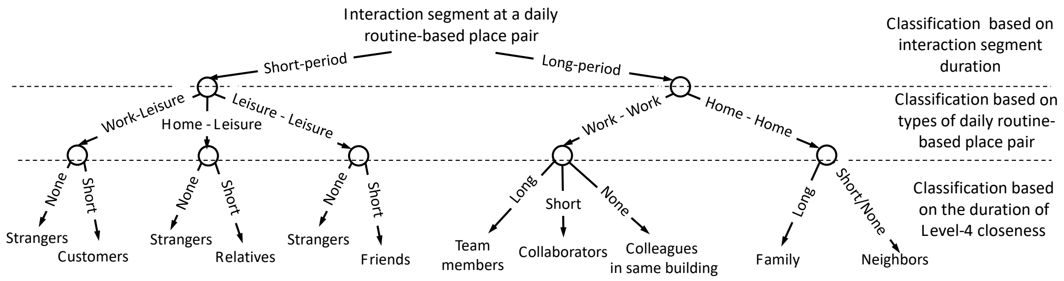

Interaction segment at a daily routine-based place pair
Short-period
Long-period
Work-Leisure
Home - Leisure
Leisure - Leisure
Work - Work
Home - Home
None
Short
Long
Short/None
Strangers
Customers
Strangers
Relatives
Strangers
Friends
Team members
Collaborators
Colleagues in same building
Family
Neighbors
Classification based on interaction segment duration
Classification based on types of daily routine-based place pair
Classification based on the duration of Level-4 closeness

Figure 2.7: Decision tree of closeness-based social relationships classification.

segment (This is the only place we apply RSS in this work). The intuition behind this approach is that the user's position changes within a place result in changing distances to every surrounding AP and thus unstable RSS from each AP. From the time series of RSS in a staying segment, we derive a time series of *RSS stability* of the i^{th} AP, denoted as $\Lambda_i = \{\lambda_1, \ldots, \lambda_j, \ldots, \lambda_t\}$, where λ_j is the standard deviation of RSS calculated based on a sliding time window W. Then we further derive the *activeness score* of a staying segment by using the equation:

$$\psi_i = \frac{\sum_{j=1}^{t-w+1} v_j}{t-w+1},\ v_j = \begin{cases} 1, \lambda_j > \lambda_{th} \\ 0, otherwise, \end{cases} \tag{2.4}$$

where the λ_{th} is a threshold of standard deviation of RSS. To ensure the robustness, we only consider significant APs (80% $\leq$ appearance rate) in each staying segment for deriving the activeness score, because the significant APs can capture the person's activeness in the entire staying segment. Thus, the activeness score is the ratio of active period over entire duration at the place. As an illustration, Figure 2.5 shows the distribution of the activeness score of all significant APs in the staying segments, when a user is dinning at a restaurant (i.e. sitting statically) or shopping in a store (i.e., walking actively), respectively. We observe more APs of dinning have lower activeness scores (less than 0.2) compared with shopping, indicating that the activeness score can well differentiate people's static and active status. We empirically set a threshold to the activeness score of each significant AP and further determine the activeness (i.e., active or static) of a staying segment based on the majority vote over all significant APs.

2.6 Social Relationships and Demographics Inference

In this section, we present how our system utilizes the activity features provided by staying segments to derive the user's fine-grained social relationships and demographics.

2.6.1 Closeness-based Social Relationships Derivation

The social relationship is about how two people interact with each other in their daily lives, including both face-to-face interaction and event the hidden interaction without encountering. Therefore, to infer social relationships, we need to understand not only a person's activities at a place, but also how the person interacts with other people at different places. Towards this end, we define the *interaction segment* based on the staying segments between two people to capture the temporal and spatial characteristics of their interactions. The basic idea is that, we first extract and characterize the interaction segments between a target user and other people based on their staying segments and corresponding activity features. Then we utilize the temporal and spatial patterns of the closenesses of the interaction segments as well as the individual daily place contexts to derive fine-grained relationships.

Interaction Segment Characterization

We generate interaction segments based on the staying segments of two people in the same day. Specifically, we first find the temporally overlapped segments between the daily staying segments from the two people. Then we estimate the physical closeness between every two overlapped segments by using the Equation 2.1. Only long overlapped segments (i.e., time duration is longer than 10min) with at least level-1 closeness are considered as valid interaction segments. Each overlapped segment is described by three characteristics: 1) *interaction time slot*, 2) *daily routine-based place pair* based on the two users' same or different personal daily place contexts at the interaction place (e.g., Home-Home or Work-Leisure), and 3) *physical closeness*, which correspond to *when, where* and *how closely* the two people interact, respectively. Finally, the characterized interaction segments represent users' interaction at the place.

Closeness-based Social Relationships Classification

After determining the interaction segments, we classify the user's social relationships leveraging the temporal and spatial patterns of the physical closeness in the interaction segments. Our approach is based on the intuition that different types of social relationships show different temporal patterns for various levels of physical closeness in the overlapped daily routine-based place, which reveal different degrees of interactions between two people. Figure 2.6 illustrates this intuition by comparing the interaction segment characteristics for two pairs of social relationships (i.e., neighbor and family, and team member and collaborator), which can be differentiated from spacial closeness degree or temporal pattern difference.

We design a triple-layer decision tree for relationships classification based on examining the characteristics of the interaction segments between two people (i.e., the temporal and spatial patterns of their physical closeness). Figure 2.7 illustrates the flow of the decision tree. In the first layer, the decision tree takes the detected interaction segment of two people in one day as input, and classifies it into two classes (i.e., Short-period and long-period interaction segment) by examining the duration of the *interaction time slot* in the interaction segment. The intuition behind this layer is that people usually spend most time at several places (e.g., homes, offices, or schools) and shorter time at other places (e.g., diners, grocery stores, and post office) and so as their interactions at these places. In the second layer, we make finer decisions from the result of the first layer. In particular, we examine the *daily routine-based place pair* of the interaction segment to further classify the interaction based on the people's individual daily place contexts. Because the short-period interaction should happen at least at one person's leisure place in logic, the short-period interaction segment leads to three possible branches: workplace-leisure, home-leisure and leisure-leisure. And the long-period interaction segment leads to the pairs of workplace-workplace and home-home. In the last layer, we further detail the classification of the interaction by analyzing the *physical closeness* of the interaction segment to infer fine-grained relationships. Specifically, we examine whether the level-4 closeness of the interaction segment is non-zero or not, which suggest the two people have or not have the face-to-face interaction in the place. The duration of the face-to-face interaction allows the decision tree to further distinguish social interaction into 8 categories of fine-grained relationships: Customers, Relatives, Friends, Team members, Collaborators, Same-building Colleagues, Family and Neighbors, as well as excluding strangers.

The decision tree infers the possible relationships between two people based on their one-day social interactions. But making relationships inference based on one-day observation may sometimes be opportunistic. For instance, students in the same school may be regarded as strangers or classmates depending on whether a face-to-face interaction is detected in one day. In order to reduce the opportunistic inferences, we propose to infer the relationships in a relative long time period (e.g., multiple days, one week or several weeks) and utilize a majority-vote approach to make the final decision.

2.6.2 Behavior-based Demographics Inference

Next, we discuss how to utilize the activity features to further capture people's behavior characteristics at various daily places and infer people's demographics (e.g., occupation, gender, religion and marriage).

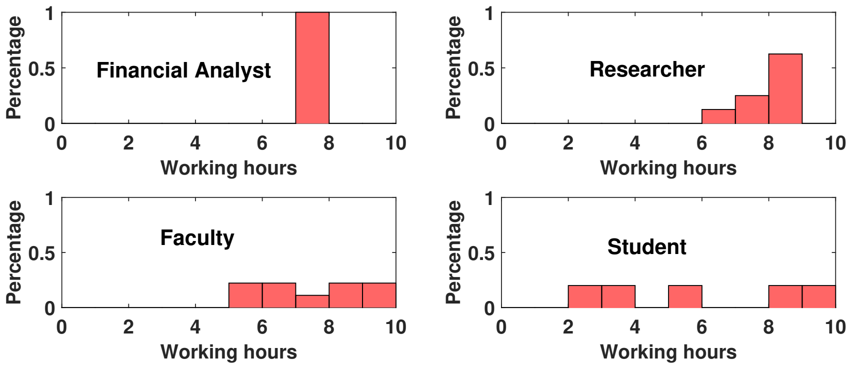


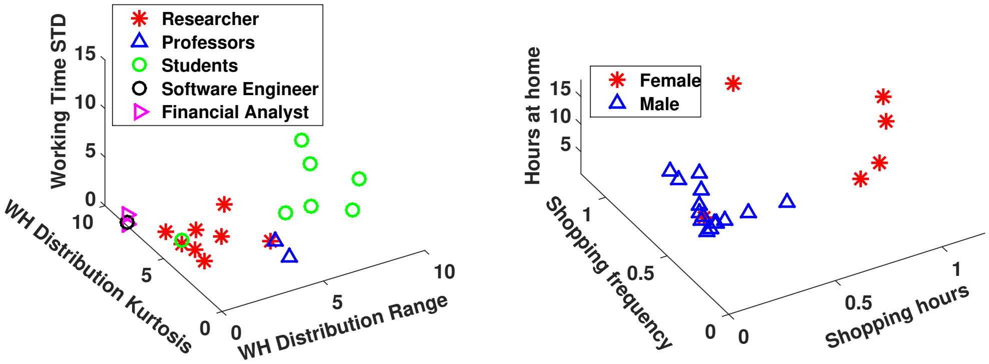
Figure 2.8: Histogram of people's working duration in a week.

Behavior Derivation at Daily routine-based Places

In this work, we define the *behavior* as the mannerisms made by an individual in the daily routine-based place during a period of time (e.g, several days). A behavior usually consists of a series of activities, and thus can be described by the temporal and spatial statistics of the activity features extracted from the staying segments across different days. In particular, we define three kinds of behaviors: 1) *home behavior*, 2) *working behavior*, and 3) *leisure behavior* based on three daily routine-based place categories. We utilize the activity features of the same daily routine-based place across multiple days to derive the features that can characterize the three behaviors. We note that the leisure behavior can be further specified according to the fine-grained daily routine-based places in Section 2.5.1.

Occupation Inference

Occupation is the job or profession of the user, which is related to the working behavior. The inference approach is based on the fact that people of different occupations have different working time slots and duration at Workplace (may include single or multiple nearby places), which reveals different working behaviors in temporal and spacial. Figure 2.8 illustrates the intuition by showing the working duration histogram of 4 users with different occupations in a week. We find that office staff has the most concentrate working duration, followed by Researchers, Faculties and Students, because company office uses more regular timetable compared with school. Meanwhile, Faculties need to leave office for teaching and faculty meeting, which leads to wider working duration distribution compared with Researchers. On the other hand, Students have the most scattered working durations because they have different number of classes for each



(a) Working behavior-based occupation inference results.

(b) Shopping and home behavior-based gender inference.

Figure 2.9: Illustration of behavior-based occupation and gender inference results.

day and flexible hours at library for study.

We derive three specific working behavior features to differentiate working behaviors for multiple days at working place. *Working hour(WH) Distribution range* describes the range of the working duration histogram, which shows the flexibility of working hours. *Working time STD* is the average standard deviation of the start and ending time of working across multiple days and *WH Distribution Kurtosis* is a descriptor of the distribution shape, which represents how concentrate the working duration is distributed. Figure 2.9(a) illustrates that the three working behaviors can well separate different types of occupations, which suggests that we can utilize a threshold-based approach to determine people's occupations by using these features. We note that different occupations may have similar working behaviors, such as financial analyst and software engineer, we can further narrow the choices for the occupation inference by leveraging the supplementary place contexts from Geo-information and user associated AP SSIDs as in Section 2.5.1.

Gender Inference

The information of user gender is more implicit compared with occupation, because there is no information from surrounding APs, which directly links to this biological characteristic. However, we find that males and females usually behave differently in some specific scenarios. For example, females tend to spend more time on housework and in-store shopping, while males tend to work for longer hours [17]. Such behavior difference shows the trend of the majority people and exists in many countries according to the survey. Thus our basic idea is to examine a person's behavior characteristics at home or in shops. From activity features, we derive three behavior features for gender inference: *shopping duration*, *shopping frequency* and *home duration*, which mainly capture the behavior patterns at home and leisure behavior at shops. Figure 2.9(b) illustrates that the three devised behavior features can well capture the differences between males and females in their behaviors at home and in shops. Additionally, we also check the user's associated AP SSIDs at leisure places, if any, to look for the particular leisure places that can differentiate gender, such as nail spa and beauty salon.

Religion Inference

We further demonstrate that it is possible to infer people's religion status (i.e. Christian or Non-Christian) from surrounding APs. The intuition is that Christian usually goes to church every Sunday and shows a regular pattern of leisure behavior around the church. Therefore, we extract three religion behavior features: *church attendance days*, *church attendance duration* and *church attendance frequency*, and apply a threshold-based method to decide Christian. We note that, by including more religion activities, we can also cover other religions or religious sects.

Relationships and Demographics Refinement

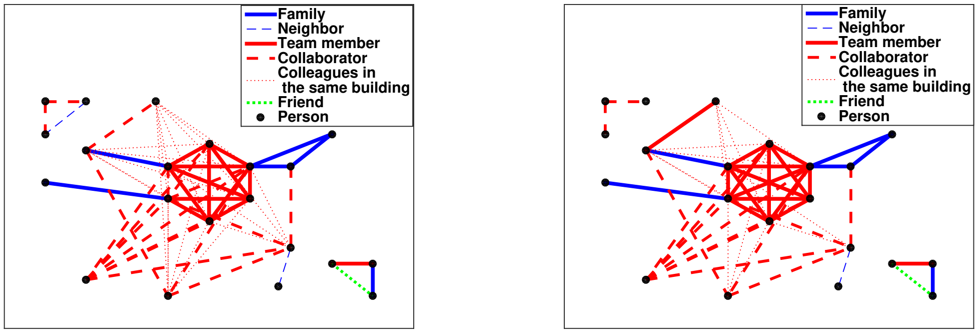
We find that the inferred relationships and demographics results can be mutually complementary. We then adopt several rules for the relationship and demographics refinement. For example, the family relationship between a male and a female is refined as the couple relationship or married; the collaborator between a faculty and a student (or a company supervisor and a software engineer) is refined as the advisor-student (or supervisor-employee) relationship.

2.7 Performance Evaluation

2.7.1 Experiment Methodology

Data Collection

Due to the limitation of the man power, we choose the representative occupations, working hours and age groups for experiments to evaluate the feasibility of our approach. We recruit 21 volunteers (i.e., 6 females and 15 males) across three cities to collect surrounding APs information in their daily lives for over 6 months. The volunteers age from 20 to 40 and are mainly from six occupations, including financial analyst, Ph.D. candidate, Master student,



(a) Social relationships inference.

(b) Social relationships groundtruth.

Figure 2.10: Social relationships comparison between inference results and the groundtruth.

undergraduate, assistant professor, and software engineer. We ask the volunteers to install a tool developed for data collection on their own phones and run it in the background throughout every day during the experiments. The users are asked to fill a questionnaire to input the groundtruth. The IRB is approved.

Hardware and Software

We include a variety of Android mobile devices in the real experiments including Samsung, Huawei, LG and Xiaomi. We develop a tool on Android platform to collect information of surrounding APs at a given frequency, i.e., 4 scans/min, which is the AP scanning frequency of many android systems [105]. For each scan, our tool collects the simple information of surrounding APs, including BSSIDs, SSID, scanning time stamp and RSS.

Evaluation Metrics

We use the following two metrics to evaluate the performance of our inference: *Detection Rate.* The ratio of correctly identified results over the total numbers in groundtruth. *Inference Accuracy.* The ratio of correct inference results over the total number of inference results.

2.7.2 Evaluation of Social Relationships Inference

We first examine the performance of social relationships inference from surrounding Wi-Fi APs. Figure 2.10 shows the comparison between the inferred social relationships (i.e., Figure 2.10(a)) among the 21 volunteers and the groundtruth from the questionnaire (i.e., Figure 2.10(b)) in graphs of relationships. Each point in the graph represents a volunteer and different types of

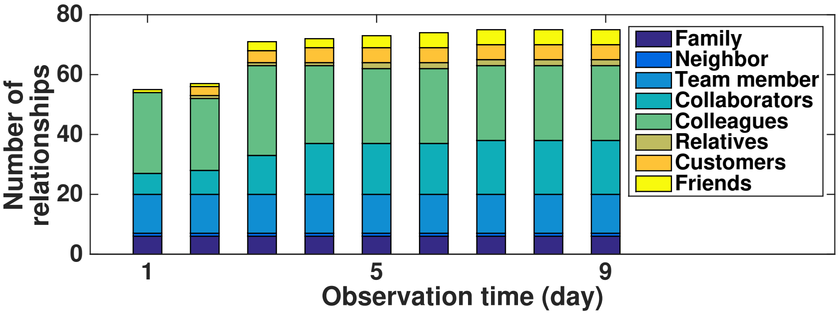


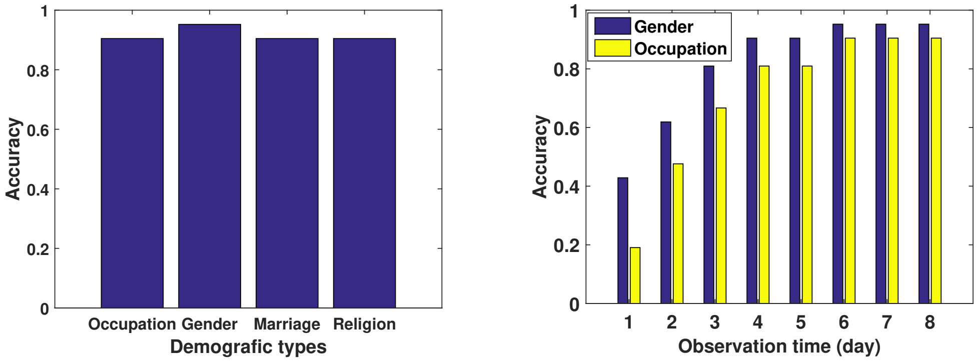
Figure 2.11: Social relationships inference results based on different length of observation time.

lines between points represent the different relationships between two volunteers. Compared to the groundtruth, the overall detection rate of social relationships inference is 91%, suggesting that our system can efficiently detect various relationships from surrounding AP information. In addition, our system also detects *hidden relationships*, which represent the potential relationship that is recognizable by our system but unknown to the two volunteers due to the lack of face-to-face interactions. We find that certain relationships (e.g., colleagues and neighbors) may contain such hidden relationship.

Table 2.1 shows the detailed statistics of our social relationships inference results. We observe that we achieve 100% detection rate for Relatives, Family and Neighbor, whereas achieve 83.3%, 94.1%, 89.5% and 87.5% detection rate for Friends, Team members, Collaborators and Colleagues, respectively, indicating that our method can accurately detect different relationships based on interaction features characterized from surrounding APs. For the misclassified relationships, one team-member relation is classified as collaborators due to irregular working time; two collaborators are classified as colleagues in the same building due to low interaction frequency. The overall inference accuracy is 95.8% when we compare the detected relationships with the groundtruth. We further detect 10 hidden relationships (i.e., 9 colleagues and 1 neighbor), while these relationships are not realized by the volunteers but can be derived from their

Table 2.1: Social relationships inference.

Relationships	**Groundtruth**	**Inference**	**Correct**	**Hidden**
Relatives	2	2	2	0
Friends	6	5	5	0
Team members	17	16	16	0
Collaborators	19	18	17	0
Colleagues	24	23	21	9
Family	6	6	6	0
Neighbor	1	1	1	1



(a) Demographics Inference Results.

(b) Demographics Inference with different observation time.

Figure 2.12: Accuracy of behavior-based demographics inference.

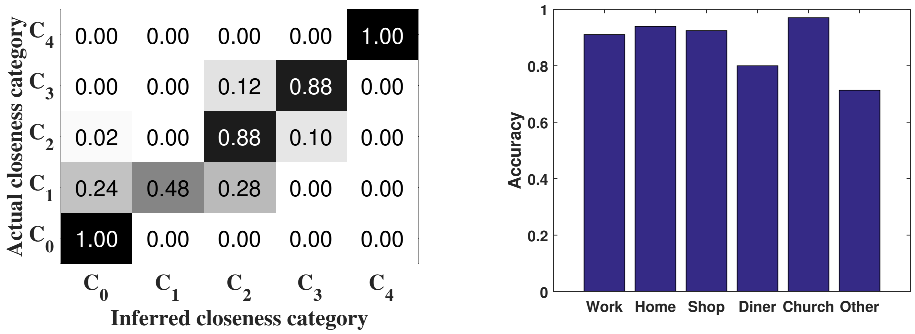
questionnaires, indicating our system can accurately detect most relationships in daily life.

Figure 2.11 shows the relationships inference results under different length of observation time. We observe that most regular relationships (i.e., family, neighbor, team member) can be detected in the first day. As for other relationships, since their interactions do not occur every day, we need to observe for more days to make a decision. The relationship inference results become stable after 5 $\sim$ 7 days, indicating that our system can detect most relationships in people's daily life based on their social interactions in one week.

2.7.3 Evaluation of Demographics Inference

Accuracy of Demographics Inference

Figure 2.12(a) shows the overall accuracy of inferring demographics. For all the demographics in our study, our system achieves over 90.5% accuracy for Occupation, Religion and Marriage, whereas the accuracy of gender inference is 95.2% for the 21 volunteers, suggesting that it is possible to accurately infer people's demographics from surrounding AP information. We further study the performance of gender and occupation inference with different length of observation time as shown in Figure 2.12(b). The inference results converge after 5 days, suggesting that people's behavior features derived in a short period (i.e., one week) can accurately infer the demographics.



(a) Classification confusion matrix of 4 kinds of physical closeness.

(b) Classification of detailed daily routine-based places.

Figure 2.13: Classification accuracy of physical closeness and daily routine-based places.

Fine-grained Social Relationships Derived from Demographics

By leveraging the derived demographics information, we further obtained refined relationships. Based on the gender information, we successfully detect all the two couples from the 21 volunteers. Besides, from the occupation inference, we specify the relationship of collaborators, e.g. who is superior and who is subordinate. In specifically, we correctly differentiate 4 superior-subordinate from 5 collaborator pairs. These results show it is possible to accurately infer fine-grained social relationships and demographics from surrounding AP information.

2.7.4 Performance of Daily Place Extraction

We randomly select 100 staying segments to examine whether our different levels of physical closeness can reflect the true relations between their physical locations. Figure 2.13(a) presents the confusion matrix of the inferred four kinds of closenesses and the results show that our system can achieve over 88% accuracy for measuring most levels of closeness except for C_1, whose inference relies on the remote APs or unstable signals. We note that the lowest level C_1 does not affect the social relationships and demographics inference as both of them mainly rely on C_4 and C_3.

Finally, we evaluate the accuracy of the contextual meaning inference with 594 detected places. Figure 2.13(b) shows we can achieve over 90% accuracy for Workplace and Home and over 80% accuracy for detailed Leisure places (e.g., Shop, Diner, Church and Other). The results demonstrate the possibility to measure the physical closeness between places and infer complex contextual meaning of daily places only from user's surrounding APs.

2.8 Discussion

Due to the limited manpower and shortage of public available data sources (i.e., containing the scanned AP signal information in large-scale areas), we evaluate our system by recruiting 21 volunteers with representative occupations and social relationship types. Furthermore, the study is based on the users' daily life activities across three cities without being restricted in a confined area. Since the participants' activities at daily places are employed as the inference basis in this work, we believe our system has the capability to successfully infer fine-grained social relationships and demographics in larger areas when given the opportunity. We demonstrate that the privacy leakage from the simple signal information of surrounding APs is significant and should arouse public attention. For the future work, we will continue our efforts to enlarge the Wi-Fi AP dataset and investigate more potential privacy leakages from such simple radio signals surrounding our daily lives.

2.9 Summary

In this work, we show that by analyzing the information from surrounding Wi-Fi Access Points (APs), the users' fine-grained social relationships and demographics could be disclosed. We present a scalable inference system that has the potential to derive people's activities at daily visited places leveraging surrounding APs and utilize such information to infer fine-grained social relationships and demographics. This implemented system only uses the simple signal features of surrounding APs such as MAC addresses and Received Signal Strength without requiring to obtain the context information by sniffing the Wi-Fi traffic. In particular, we describe people's daily places in three dimensions (i.e. time, space and context) to infer people's activities and extract their activity features as well as their physical closeness at same places. Our *Closeness-based Social Relationships Inference* algorithm further analyzes people's physical closeness to capture when, where and how closely people interact to reveal fine-grained social relationships, while the *Behavior-based Demographics Inference* method extracts people's various individual behavior from their activity features to infer demographics. By using the data collected by 21 participants in their daily lives over 6 months, our system confirms the possibility of using surrounding APs to infer people's social relationships and demographics with over 90% accuracy.

Chapter 3

Revealing Personal Privacy from Wearable Devices

3.1 Background

The convenience of wearable devices, such as smartwatches and fitness bands (e.g., Fitbit and Jawbone), has greatly stimulated the growth of the market of mobile devices in recent years; market researchers estimated that 72.1 million wearable devices will be shipped in 2015, which will be about 173% from the 26.4 million wearable devices shipped in 2014 [8]. Such increasing popularity of wearable devices has enabled a broad range of useful applications, including fitness tracking, falling detection, gesture control and user authentication. Since such wearable devices have the ability to capture users' hand movements and derive human dynamics directly, a major concern arises on whether a user's sensitive information could be leaked and obtained by adversaries including the user's PIN sequence when accessing an ATM machine or using debit cards for payment.

In this work, we demonstrate that a user's personal PIN sequence could be leaked through his wearable devices (e.g, smartwatch or fitness tracker), when accessing a key-based security system. Such systems are very common in daily lives. Examples include accessing ATM cash machines, electronic door locks, and keypad-controlled enterprise servers. A key-based security system requires people to enter personal key combinations on the keypad for identity verification. With people tending to wear wearable devices around-the-clock, the movements of their wrists during the key entry process to a security system (i.e., clicking keys and moving between clicks) are captured by the sensors on wearable devices. As such, wearables could cause a new way of sensitive information leakage when a user accesses the key-based security systems. In particular, adversaries can obtain sensor readings of wearables via sniffing Bluetooth communications [108, 96] or installing malwares [6] on the devices, and further infer the user's PIN sequence (e.g., ATM PIN sequences or key sequences on access control panels) for his own use.

There has been active study on sensitive information leakage when using key-based security systems. Traditional attacks rely on either shoulder surfing or hidden cameras [81, 35]. Such attacks require direct visual contact to key entry actions and additional installation efforts.

Furthermore, Shukla *et al.* propose a side-channel attack utilizing a camera-based method to recover smartphone lock PINs from the user's spatial-temporal hand dynamics without directly seeing the keypad on screen [107]. The proposed method has a low inference accuracy and requires cameras to capture the user's hand and the back side of the touch screen. Two recent work [118, 78] propose to utilize sensors in smartwatches to infer user's typed words or passwords. The MoLe [118] system relies on a linguistic model to infer user's typed words, which is difficult to work with non-contextual inputs. Liu *et al.*[78] devise a system that requires training of the sensor data to classify user inputs.

In contrast to these prior studies, we develop a training-free, context-free technique to reveal a user's private PIN sequence (to a key-based security system) when a wrist-worn wearable device is employed. The wrist-worn wearable devices could be either smartwatches or fitness trackers. While the digital smartwatch is designed to be worn on either hand, the user can wear it on the right hand without the concern on traditional watch designed to adjust time easily when wearing it on the left hand. Additionally, many people tend to wear fitness tracker on the right hand while keeping wearing traditional watch on the left hand. The basic idea is to exploit embedded sensors in wearable devices to capture dynamics of key entry activities and derive fine-grained hand movement trajectories traversing secret key entries. While wearable devices have equipped with various sensors, it is challenging to accurately recover such fine-grained hand-movement trajectories that exhibit only mm-level difference in distance between keys via low-fidelity sensors. In addition, due to hand vibrations and rotations, the coordinate system of a wearable device is not always aligned with a fixed reference, which makes it hard to track the hand movements by using sensor readings directly. Additionally, in order to obtain a person's key entries without user cooperation or drawing any attention, the adversary has to achieve the PIN sequence with no training or contextual information.

To address these challenges, our approach examines the inherent physics phenomenon extracted from the user's key entry activities via wearable sensors and develops distance calculation and direction derivation schemes to produce mm-level accuracy when estimating the moving distance and angle between two consecutive key entries. To obtain the complete PIN sequence, our backward PIN-sequence inference algorithm exploits the physical constraints of distance between keys and temporal sequence of key entry activities to construct a tree of candidate key entries for determining the PIN sequence in a reversed manner, because in many practical cases, the "Enter" key is the last key after the user enters his/her PIN sequence. The mm-level precision of estimating the fine-grained moving distance and direction between two keys and the backward PIN-sequence inference algorithm enable our system to obtain the user's PIN sequence without training and contextual information. Through extensive real experiments, we find that our PIN sequence inference algorithm can achieve high accuracy regardless of different types of wearables and layouts of keypads. Furthermore, the performance of our system is consistently good even under low sampling rate (e.g., 25Hz) or when inferring long PIN sequences. Such a technique can easily be extended to support password recovery when people type on keyboards while wearing wearables.

We summarize our main contributions as follows:

- We demonstrate that a single wrist-worn wearable device can reveal a user's PIN sequence to key-based security systems. We develop a training-free approach by exploiting the inherent physics meaning extracted from sensor readings on wearables. Such an approach does not require contextual information, allowing it to recover random key entries.

- We develop the distance estimation and direction derivation schemes that capture the fine-grained hand movements at mm-level precision.

- We show that it is possible to infer a complete user's PIN number via a backward PIN-sequence inference algorithm. By exploiting spatial and temporal constraints of PIN entries and the fine-grained hand movement analysis, our approach can accurately pinpoint the location of each PIN entry with the right sequence.

- We conduct extensive experiments with 20 participants wearing two types of smartwatch and a prototype of wearable on key-based security systems such as ATM keypads and keyboards over a thirteen-month period. We show that our system can achieve 80% accuracy of inferring PIN sequences with only one try and over 90% accuracy with three tries without training and contextual information.

- We evaluate the performance of our system when inferring the PIN sequences with increased PIN length and under different sampling rates. We demonstrate that our system can achieve a good performance when inferring long PIN sequences (e.g., 6-PIN sequences) and under low sampling rate (e.g., 25Hz).

The rest of the chapter is organized as follows. We first put our work in the context of related studies in Section 3.2. In section 3.3, we investigate the feasibility of using wearables to obtain a user's PIN sequence of key-based services. We then describe the design of our PIN-sequence inference framework in Section 3.4. Next, we present two schemes of distance estimation and direction derivation to capture fine-grained hand movements via sensors on wearables in Section 3.5. The backward PIN-sequence inference algorithm to recover the complete user PIN sequence is described in Section 3.6. We present the detailed implementation of our framework in terms of pre-processing of the sensor data and coordinate alignment in Section 3.7. In Section 3.8, we perform extensive evaluation of our approach involving real key-based security systems. Finally, we discuss the relative issues and conclude our work in Sections 3.9 and 3.10 respectively.

3.2 Related Work

Recent studies show that embedded sensors on mobile devices, such as accelerometers and touch screens, can capture users' motion and leak their sensitive information [86, 104, 94, 41, 103, 129]. Recently, wearable devices, such as smartwatches and fitness bands, extend the sensing capability to limbs and enable many useful applications [77, 132, 90, 66]. These existing studies have shown the sensing capabilities of up-to-date mobile devices, which inspire us to explore the potential of using wrist-mounted wearables to recover fine-grained hand movements, and study to what extent the user's sensitive information could be leaked from their fingers.

Toward this end, we explore the possibility of recovering people's private PIN sequences through their wrist-worn mobile devices when they enter PINs on key-based security systems. Traditionally, key-based security systems could be breached by several methods, such as hidden cameras and skimmers [35, 7, 130]. For example, some ATM machines are attached by a hidden camera, which was used to record PIN sequences or body movements of entering PINs [81]. An adversary may also put a skimmer into the ATM machine card slot. When the customer slides their card, it will go through the skimmer first and then into the machine. A chip inside the skimmer device records information about the account without the knowledge of the customer [1]. These existing methods largely depend on installing dedicated devices in the restricted area.

In addition, researchers show that it is possible to recognize users' keystrokes by using acoustic approaches. Berger *et al.* [36] demonstrate that by using linguistic models and recorded typing sound on a keyboard, an attacker can successfully reconstruct the typed words. Zhu *et al.* [138] present a context-free and geometry-based approach to recover keystrokes by using multiple smartphones to record acoustic emanations from the keystrokes. Wang *et al.* [121] develop a system that extracts and optimizes the location-dependent multipath fading features from the audio signals and leverages the signal diversity resulted from the dual-microphone interface in a mobile device to identify key entries typed on a keyboard. Along this line, Jian *et al.* [76] demonstrate that mobile audio hardware in off-the-shelf mobile devices can be exploited to discriminate mm-level position differences, based on which they develop a system that can locate the origin of keystrokes by using only a single phone behind a keyboard [76]. Martinovic *et al.* demonstrate that the captured electroencephalography (EEG) signals from head-wearable EEG devices can reveal whether the presented stimuli (e.g., images) are related to the user's private information such as bank cards, area of residence and PIN numbers. [83]. Marquardt *et al.* develop an application that can utilize accelerometers in a smartphone to sense the vibrations caused by keystrokes from a nearby keyboard and further identify the keystrokes [82]. Their proposed technique relies on a linguistic model and labeled training data and the system is highly sensitive to environment noise (e.g., people moving around).

The most related work to ours are two concurrent studies, which analyze the leak of users' passwords or typed words from smartwatches [118, 78]. Wang *et al.* [118] devise a system that can infer typed words on a keyboard by utilizing motion sensors in smartwatches. The system assumes to know the fixed initial position of the smartwatch and relies on a linguistic model to infer typed words, which makes it hard to deal with non-contextual inputs, such as passwords and PIN sequences. Liu *et al.* [78] apply sensors in a smartwatch to infer users' inputs on a keyboard or POS terminal by utilizing machine-learning based techniques. Their approach requires training of hand movements between keystrokes, and it is unclear how the system handles changing positions of the wrist during typing. Moreover, both of the above work can only achieve moderate accuracy in deriving the user inputs given limited number of tries. Different from previous work, our key entry inference system is training-free, contextual-free and does not involve additional devices. Furthermore, our backward PIN-sequence inference framework is not subject to environmental noises, such as ambient noise, light interference and people walking around.

3.3 Attack Model and Feasibility Study

The positions of wearable devices on human bodies naturally enhance the devices' capability of the activity recognition and facilitate many applications based on the context of activities. However, such strong sensing ability brings up new security and privacy issues. In this work, we study the possible personal secret leakage in a very common scenario that people wear wrist-worn wearable devices while using key-based security systems, such as ATM machines, password secured door entries, and keypad-controlled enterprise servers. In this section, we describe the attack model and explore the feasibility of utilizing wearable devices to recover personal key entries in key-based security systems.

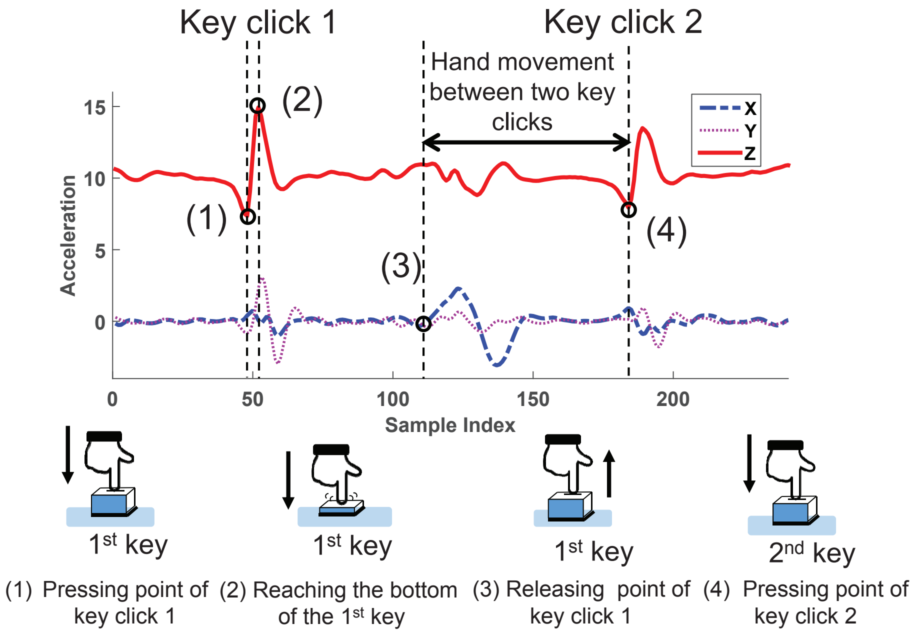


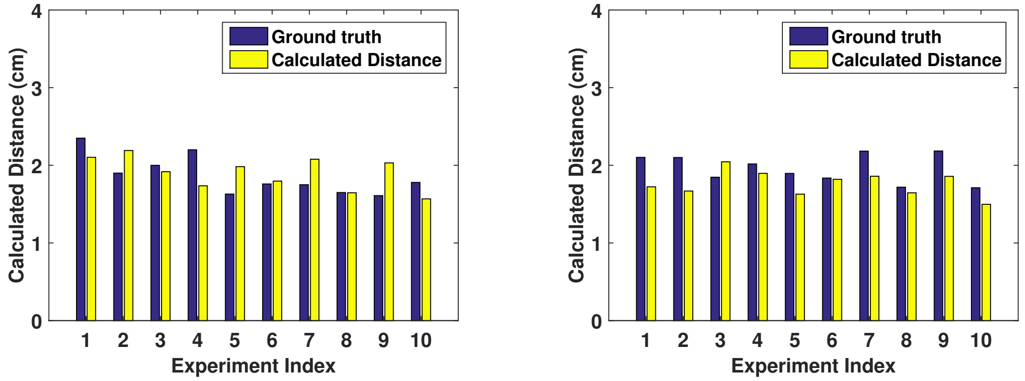
Figure 3.1: Acceleration patterns inherited from key entry activities, shown in the readings of a 3-axis accelerometer on IMU.

3.3.1 Attack Model

We consider an adversary aiming at recovering a person's secret PIN entries leveraging embedded sensors (e.g., accelerometer, gyroscope and magnetometer) in wearable devices worn on his/her wrist. The adversary has the knowledge of where the victim visits the key-based security system and can obtain the layout of the keypad. We assume that the adversary is able to access the sensor data and communicate over networks on the smartphone, but cannot observe the PIN entry activities visually by any means. The wearable device is usually paired with the user's smartphone via Bluetooth and constantly sends sensor data to the person's smartphone for logging purpose. Most wearables are using Bluetooth Low Energy (BLE) to transmit sensor data. With low energy, BLE comes with low security capability compared with Bluetooth. As a result, for example, the sensor data could be sniffed by the adversary by using Bluetooth sniffing techniques [31, 89].

But the adversary does not have access to training data, which is specific to a particular key-based security system. Particularly, we identify two representative attacking scenarios as follows:

Sniffing Attacks. An adversary can place a wireless sniffer close to a key-based security



(a) Distance estimation between keys 4 and 5

(b) Distance estimation between keys 8 and 5

Figure 3.2: Distance estimation of the number pad on the Dell keyboard based on IMU.

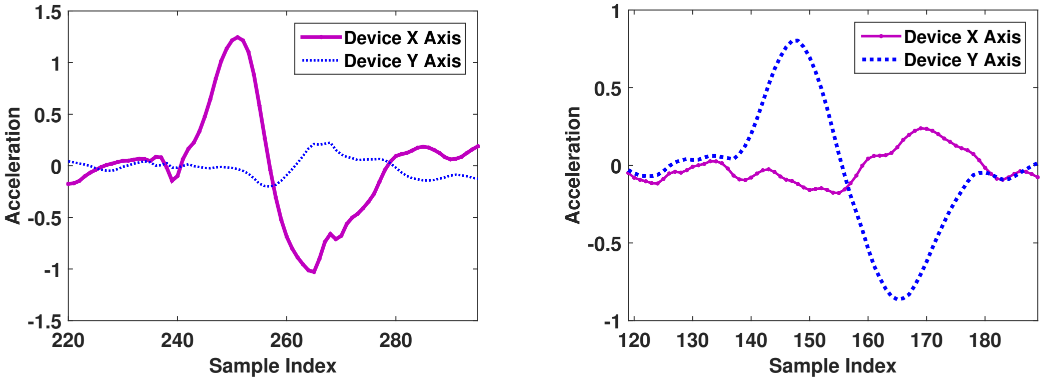
system (e.g., ATM machine or key-based security door) to eavesdrop sensor data from the wearable device, which is worn on the victim's wrist when he/she enters security PINs into the security system. The adversary utilizes the wireless sniffer to capture Bluetooth packets sent by the wearable device to its associated smartphone [108, 96, 44], and determines the victim's PIN sequence based on the sensor data extracted from Bluetooth packets.

Internal Attacks. An adversary can access the embedded sensors in the victim's wrist-worn wearable device by installing a malware app without the victim's notice[6]. The malware app waits until the victim accesses the key-based security system and keeps sending sensor data back to the adversary's server through the Internet. The adversary can aggregate the sensor data on the server to determine the victim's PIN sequence remotely.

3.3.2 Intuitions of Hand Movements behind Key Entry Activities

When accessing a key-based security system, a person's PIN sequence is entered through multiple key clicks. During each key click, there exhibits acceleration and deceleration of keys when pressed and released by the user. This simple information can serve as a guideline to discriminate different key clicks. The critical question we need to answer is that whether the sensors on wearable devices can discriminate between key clicks and capture the fine-grained movements between two consecutive clicks. In particular, we look for unique sensing patterns inherited from such acceleration and deceleration that could be used to facilitate the discrimination of key clicks and distance estimation of hand movement between two key clicks.

A key click can be separated into two consecutive time periods: *key pressing* and *key releasing*



(a) Moving along the X axis.

(b) Moving along the Y axis.

Figure 3.3: Accelerometer readings from IMU.

periods. The key pressing period starts when a person's finger touches the key and ends when the finger presses the key to the bottom of the keypad (denoted as *pressing point*). The key releasing period starts when the person's finger releases the key from the bottom of the keypad and ends when the finger stops moving after it is detached from the key (denoted as *releasing point*). Intuitively, the hand accelerates towards the keypad while pressing the key before the pressing point, and decelerates and stops quickly due to the reaction force from the key that touches the bottom of the keypad. When releasing the key, the hand accelerates towards the opposite direction to the keypad and stops after the finger is detached from the keypad. We illustrate the hand's acceleration/deceleration in the Z-axis caused by key pressing and releasing in Figure 3.1. We use the keypad's coordinate system with the Z-axis perpendicular to the keypad plane and pointing out from the keypad, and the X-axis aligned to the direction connecting the first and the second key.

Furthermore, in between two consecutive key clicks, the key entry activity involves the hand movement from one key to another. As shown in Figure 3.1, the accelerations on the X axis present an obvious up-and-down trend, while the accelerations on the Z and Y axes remain stable. The intuition behind this phenomenon is that the hand usually accelerates and moves relatively in parallel with the keypad on the shortest trajectory between the first and second keys. After passing the middle point of the trajectory, the hand decelerates to stop when it reaches the Key 2's position. Such unique up-and-down acceleration trend is very useful to help capturing the small distance of hand movement between two keys.

Feasibility Study. To study whether the sensors on wearables can capture such detailed acceleration patterns during key entry activities, we conduct two sets of experiments on the number pad of a Dell USB wired keyboard L100 with an Invensense MPU-9150 9-axis motion sensor (i.e., IMU), which is a prototyping alternative to a wearable device. The sensor uses a moderate sampling rate of 100Hz and contains an accelerometer, gyroscope and magnetometer that are comparable to embedded sensors in wearable devices. During the experiments, the participant wears the sensor on his wrist and keeps his hand in parallel to the keypad below so that the sensor's Z axis points out and is perpendicular to the keypad. The first set of experiments moves from keys 4 to 5, which is along the sensor's X axis, and the second set of experiments moves from keys 5 to 8 along the sensor's Y axis. The distance between keys 4 to 5 is only $1.9cm$, the same as that between keys 5 to 8. We use a camera on top of the keyboard to record the moving distance ground truth of the sensor. We note that these two experiment setups are special as the sensor's coordinate system is fully aligned with the keypad's coordinate system.

We estimate the sensor's moving distance by applying the double integration to the acceleration readings of the X axis and the Y axis from the accelerometer on the sensor. The details of the distance estimation scheme are presented in Section 3.5. Figure 3.2 compares the ground truth and the estimated distance in 10 runs of aforementioned settings, respectively. We find that overall the estimation errors are less than $1cm$, the mean error of the 10 runs of each experimental setting is as low as $0.27cm$ and $0.24cm$ on the X and Y axes, respectively.

Additionally, we find that there is an unique up-and-down acceleration pattern captured by the sensor, which can be utilized to determine the sensor's moving direction. Figure 3.3 shows that the up-and-down acceleration pattern (like a sine wave) appears on X and Y axes respectively when the sensor is moving along X or Y axes. The capability of accurate distance estimation of the small moving distance between keys and the moving direction determination are the foundation for recovering the user's secret PIN sequence. Thus, these observations are encouraging as they indicate the sensors on wearables have the capability to capture the fine-grained hand movements to facilitate PIN sequence recovery.

3.4 System Design

In this section, we discuss the challenges in our system design and provide an overview of our system.

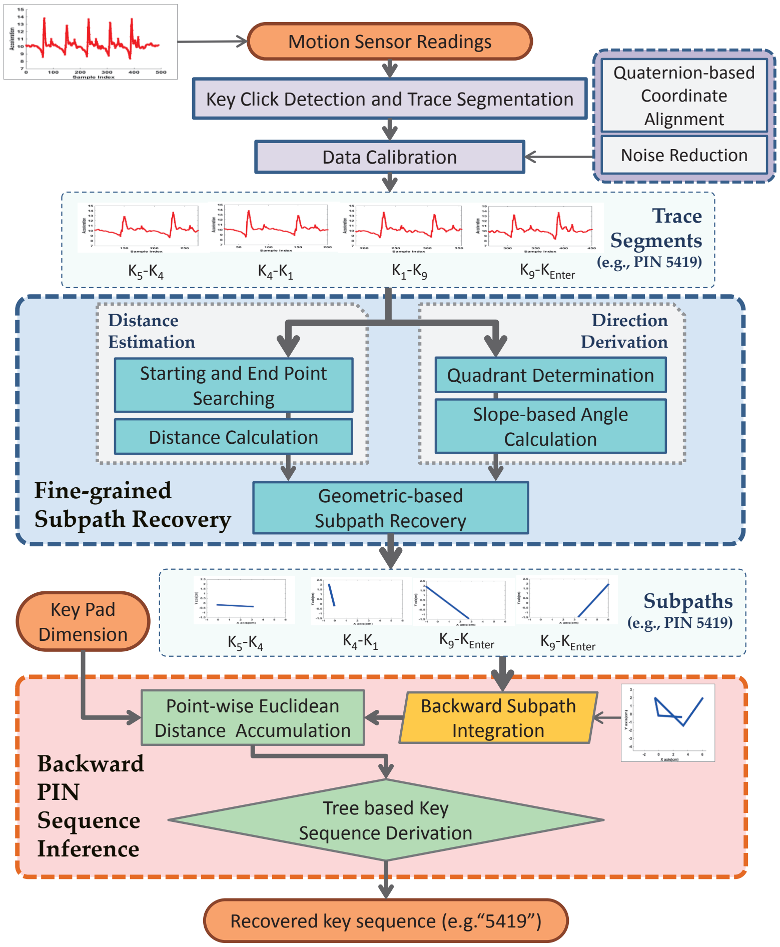

Motion Sensor Readings
Key Click Detection and Trace Segmentation
Quaternion-based Coordinate Alignment
Noise Reduction
Data Calibration
Trace Segments (e.g., PIN 5419)
K5-K4
K4-K1
K1-K9
K9-KEnter
Distance Estimation
Direction Derivation
Starting and End Point Searching
Distance Calculation
Quadrant Determination
Slope-based Angle Calculation
Fine-grained Subpath Recovery
Geometric-based Subpath Recovery
Key Pad Dimension
Subpaths (e.g., PIN 5419)
K5-K4
K4-K1
K9-KEnter
K9-KEnter
Point-wise Euclidean Distance Accumulation
Backward Subpath Integration
Backward PIN Sequence Inference
Tree based Key Sequence Derivation
Recovered key sequence (e.g."5419")

Figure 3.4: PIN-sequence inference framework.

3.4.1 Challenges

The goal of accurately recovering personal PIN sequences by using the embedded sensor of wearable devices worn on the victim's wrist is not trivial. Our system design and implementation need to overcome the following challenges:

Robust Fine-grained Hand Movement Tracking. Using embedded sensors in wrist-worn wearable devices to reconstruct the trajectories of hand movements in key-entry activities is challenging since the sensors not only capture the acceleration patterns of key clicks and movements from key to key, but also are affected by the users's unconscious hand vibration and rotation. Furthermore, due to the limited size of the keypad, the distance between keys is small, making it hard to estimate using the low-grade sensors on wearables. Thus, we need to design distance estimation and direction derivation schemes to accurately estimate the hand moving distance between keys and track the direction of fine-grained hand movements despite various interfering sensing factors.

Training-free Key Entry Recognition. Considering the attacking nature of our goal, it would be unlikely for the adversary to collect any training data (e.g., sensor data of hand movements) before recovering a user's PIN sequence. And it is also unlikely to have the user's cooperation during this process. Thus, we aim to infer the user's secret PIN sequence leveraging wearables without training efforts involving target users' participation.

Recovering PIN Sequence without Contextual Information. The target user's PIN sequences used in key-based security systems are usually consisted of numbers without contextual information or linguistic meaning. Our developed method should have the ability to recover sensitive information consisting of random combination of numbers. This requires our system to be able to recover PIN sequences without relying on linguistic model or dictionaries.

Sensing with Single Free-axis Wearable Device. Using a single wearable device to recover PIN sequence is necessary because usually there is only one wearable device available on the wrist of the hand that performs key entry activities. There is no reference point available besides the single wearable device. Furthermore, sensor readings are with respect to the wearable device's coordinate system, which is not stable and changes often according to the device's posture. In order to recognize key entry activities and derive fine-grained hand movement trajectories, it is important for our system to translate the sensor readings from the wearable device's coordinate system to a fixed coordinate system, such as the keypad's coordinate system.

3.4.2 System Overview

The main goal of our work is to demonstrate that using wearable devices could reveal people's secret PIN sequence to key-based security systems such as ATM machines, electronic-key based door entries, and enterprise servers. We design and implement a system that has the capability to reveal target user's secret PIN sequences through tracking the fine-grained hand movement trajectories related to key entry activities. The basic idea is to examine the acceleration of the user's hand movements when accessing key entry based security systems. Based on the feasibility study of two special cases in Section 3.3, wrist-worn wearables can capture the unique patterns of acceleration embedded in the hand movements caused by entering the secret PINs. Such unique patterns can be exploited to estimate hand moving distances and directions during the key-entry activities, which can be leveraged to reconstruct fine-grained moving trajectories of the user's hand and infer the PIN sequence traversed by the trajectories.

The flow of our system is illustrated in Figure 3.4. Our system takes as input the raw sensor readings, such as acceleration, rotation rate, and quaternion, from the wearable device worn on a target user's wrist. Then the system performs *Key Click Detection and Trace Segmentation* to detect each key click by examining accelerations and separate the sensor readings into segments containing consecutive key entries. The *Data Calibration* utilizes *Quaternion-based Coordinate Alignment* and *Noise Reduction* techniques to translate each segment of accelerations into the measurements with respect to the coordinate system of the keypad, and remove noise from readings by using the Savitzky-Golay filter.

The core of our system consists of two components, *Fine-grained Subpath Recovery* and *Backward PIN-Sequence Inference*, which first estimate the distance and direction of hand movements in each segment of acceleration collected between two consecutive key entries, and then integrate the estimated distance and direction of each segment to determine the entire PIN sequence based on the physical constraints of the keypad and temporal relationship of the key entering sequence. We define a *subpath* as the trajectory of the user's hand movement between two consecutive key clicks inside one segment. As shown in Figure 3.4, the Fine-grained Subpath Recovery consists of two subtasks: *Distance Estimation* and *Direction Derivation*. The Distance Estimation identifies the unique acceleration patterns embedded in the key pressing and releasing activities and perform distance estimation based on such patterns. Additionally, the Direction Derivation leverages the estimated distance together with the acceleration patterns caused by the hand movement in each subpath to derive the hand moving direction.

After obtaining the estimated moving distance and direction in each subpath, the system develops the Backward PIN-Sequence Inference to recover the user's PIN sequence. Specifically, our system first applies the *Backward Subpath Integration* to combine subpaths in a backward manner in time series. Then the system performs *Point-wise Euclidean Distance Accumulation* to calculate the accumulated Euclidean distance for each candidate of key sequence at each estimated key position (i.e., point-wise). Last, the *Tree based Key Sequence Derivation* generates a tree with the candidates of key sequence and their accumulated Euclidean distance. The key sequence candidate with the minimum accumulated Euclidean distance is chosen to be the output of the system, which is the inferred PIN sequence that the victim uses in the key-based security system. Note that, this work can be extended to identify keyboard typing or keyboard passwords by using the Bayesian model and dictionaries [118].

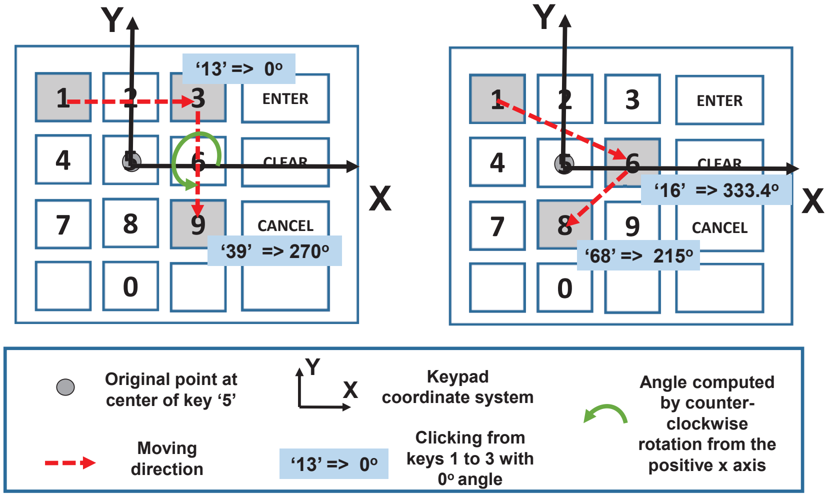

Y
X
'13' => 0°
1
2
3
ENTER
4
5
6
CLEAR
7
8
9
CANCEL
'39' => 270°
0
'16' => 333.4°
'68' => 215°
Original point at center of key '5'
Keypad coordinate system
Angle computed by counter-clockwise rotation from the positive x axis
Moving direction
'13' => 0°
Clicking from keys 1 to 3 with 0° angle

Figure 3.5: Illustration of the coordinate system on a typical key pad and examples of moving directions of key clicks, 13, 39, 16, and 68.

3.5 Distance Estimation and Direction Derivation Schemes

Our system requires tracking hand movement trajectories on small keypads accurately without training. Inspired by the basic dead reckoning technique, we seek to derive such fine-grained trajectories based on hand movement distances and directions. Particularly, we develop *Distance Estimation* and *Direction Derivation* schemes to estimate the distances and derive direction for each subpath (i.e., between two consecutive key clicks).

3.5.1 Distance Estimation

In order to accurately estimate the hand movement distance between two consecutive key clicks, we need to identify the patterns in the sensor data corresponding to the hand movement precisely. Therefore, our system needs to first search the starting and ending points of the sensor data caused by the hand movements based on pressing and releasing points of key clicks; then calculate the hand moving distance by utilizing the extracted patterns from the sensor data. In the rest of the section, we assume the system has performed the *Key-click Detection* and segmented the sensor data to traces that capture hand movements between two consecutive key clicks. The sensor data in each trace are translated into keypad coordinate system through *Coordinate Alignment*. The details of Key-click Detection and Coordinate Alignment will be discussed in Section 3.7. Figure 3.5 illustrates the coordinate system of a typical ATM keypad, where the center of key 5 is the origin; the directions of positive X and Y axes are in parallel with the direction from keys 5 to 6 and keys 5 to 2, respectively; and the Z axis is perpendicular to the X-Y plane, pointing out from the surface of the keypad. The four quadrants of the X-Y plane are defined as the standard quadrants in a two-dimensional Cartesian system. Figure 3.5 also shows some examples of moving directions of key clicks, e.g, 13 indicates clicking from keys 1 to 3.

Starting and Ending Points Searching based on Pressing and Releasing Points. The hand movements from one key to another happen after releasing the first key and end when touching the second key. Ideally, the hand movement distance can be calculated based on the acceleration (e.g., acceleration from the Z-axis) extracted between the releasing point of the first key click and the pressing point of the second key click. However, such coarse segmentation includes the sensor data resulted from hand vibrations usually result in large estimation errors. In Section 3.3, we find that the acceleration captured during the hand movements between consecutive key clicks has significant and unique patterns on X and Y axes (i.e., either up-and-down or down-and-up shapes due to different moving directions).

Apparently, such unique acceleration patterns include merely the dynamics of the key-to-key hand movements, and can be further utilized to facilitate accurate hand moving distance estimation. In order to determine the right segment of acceleration data corresponding to the unique acceleration pattern, we propose to further search the starting and ending points of the pattern based on the segment of sensor data. Specifically, we define the first zero-crossing point occurring before and after the unique acceleration pattern as the *starting point* and *ending point*, respectively. The intuition behind this is that when a hand moves from one key to another, its moving trajectory is mainly in parallel with the X-Y plane of the keypad. Therefore, the acceleration and deceleration of the hand during such movement dominates the acceleration on X and Y axes, and results in the acceleration that always experiences a pattern of $[0, a_{k,max}(a_{k,min}), 0, a_{k,min}(a_{k,max}), 0]$ as shown in Figure 3.6, where $a_{k,max}$ and $a_{k,min}$ denote local maximum and minimum of acceleration on X and Y axes with $k = x$ or y.

Thus, we design a strategy to locate the starting and ending points of the unique acceleration pattern so that we could estimate the distance between two key clicks accurately. Our strategy involves the following steps: 1) extract the acceleration on X and Y axes between the releasing and pressing points of two consecutive key clicks respectively; 2) examine the extracted acceleration to find the $a_{x,max}, a_{x,min}, a_{y,max}, a_{y,min}$; 3) determine the *dominated axis* by choosing the axis has the more significant unique acceleration pattern (i.e., a larger peak-to-peak value defined by $|a_{k,max} - a_{k,min}|, k = x$ or y); 4) find the starting point of the unique pattern on the dominated axis by searching the first time that acceleration crosses the axis (i.e., zero-crossing

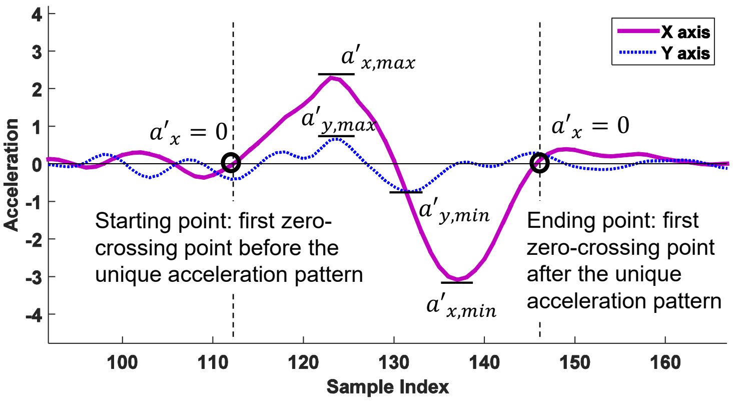


Figure 3.6: Searching for starting and ending points based on releasing and pressing points within an acceleration segment.

point) before $a_{k,max}$ or $a_{k,min}$, whichever occurs earlier; 5) similarly, find the ending point of the unique pattern on the dominated axis by searching the first zero-crossing point after $a_{k,min}$ or $a_{k,max}$, whichever occurs later. The accelerations within the starting and ending points derived above merely correspond to the hand movements between two consecutive key clicks and are utilized to calculate the hand movement distance and direction in our schemes.

Distance Calculation. The distance estimation between two consecutive key clicks is obtained by considering the movements in both X and Y axes. To perform accurate estimation, we compute the small movement between two samples in sensor data and then sum up to produce the distance estimation in one acceleration segment bounded by the identified starting and ending points. As the distance is two times integration of accelerations, we utilize trapezoidal rule to approximate each integration.

3.5.2 Direction Derivation

In order to recover the complete PIN sequence, our system needs to determine the moving direction of each subpath during the key-entry process in addition to the distance. We define the moving direction of a subpath as the angle between the positive X axis and the subpath with counter-clockwise rotation as shown in Figure 3.5. The moving direction is denoted as $\vartheta \in [0360$. The basic idea is to find the direction based on the ratio of distances on X and Y axis derived from hand movement acceleration. In particular, we design a two-step approach, including the *Quadrant Determination* and *Slope-based Direction Calculation*. The Quadrant Determination first leverages the unique acceleration patterns to determine which quadrant of X-Y plane that the hand moving direction belongs to. Then the Slope-based Direction Calculation examines the slope angle of the moving direction in a quadrant ranging from $0 to 90$ based on the hand movement distances on X and Y axes, and converts the slope angle to the moving direction ϑ.

Quadrant Determination. Intuitively, the hand movement acceleration projected on X and Y axes results in different combinations of the unique acceleration patterns in terms of the order of $a_{k,max}$ and $a_{k,min}$ on X and Y axes with $k = x$ or y. For example, when the hand moves towards 45°, the acceleration on X and Y axes both experiences the $a_{k,max}$ before the $a_{k,min}$, while the acceleration on the X axis experiences the $a_{x,max}$ after the $a_{x,min}$ and the acceleration on the Y axis experiences the opposite when the hand moves towards 135°. Therefore, we leverage the combinations of unique acceleration patterns on X and Y axes to determine the quadrant that a certain moving direction should belong to. Specifically, the quadrant of the moving direction can be determined by the following equation:

$$Q = \begin{cases} 1;\ if\ I_{a_{x,max}} < I_{a_{x,min}} \&\ I_{a_{y,max}} < I_{a_{y,min}}, \\ 2;\ if\ I_{a_{x,max}} > I_{a_{x,min}} \&\ I_{a_{y,max}} < I_{a_{y,min}}, \\ 3;\ if\ I_{a_{x,max}} > I_{a_{x,min}} \&\ I_{a_{y,max}} > I_{a_{y,min}}, \\ 4;\ if\ I_{a_{x,max}} < I_{a_{x,min}} \&\ I_{a_{y,max}} > I_{a_{y,min}}. \end{cases} \tag{3.1}$$

where Q is the quadrant index, $I_{a_{axe,max}}$ and $I_{a_{axe,min}}$ denotes the index of the local maximum and minimum on X and Y axes, respectively.

Slope-based Direction Calculation. After quadrant determination, we compute the slope angle of the moving direction within each quadrant based on the ratio of the distance on X and Y axes by utilizing the following equation:

$$\phi = \left| arctan\left(\frac{s_y}{s_x}\right) \right|. \tag{3.2}$$

Equation (3.2) returns the relative moving direction defined in a quadrant ranging from 0° to 90°, we further convert the ϕ to an absolute moving direction (i.e., the direction defined within keypad coordinate ranging from 0° to 360°). Given the quadrant index Q, the absolute moving direction ϑ can be derived as follow:

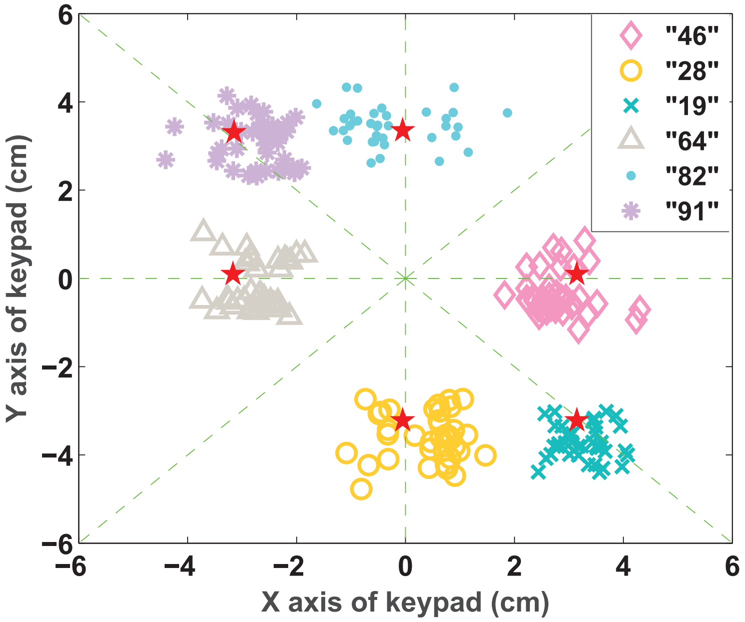


Figure 3.7: Illustration of the clustering results of distance estimation and direction derivation for 6 different subpaths $\{46, 28, 19, 64, 82, 91\}$ by treating the first key click as the origin. The red star is the ground truth.

$$\vartheta = \begin{cases} \phi; & if\ Q = 1, \\ 180^\circ - \phi; & if\ Q = 2, \\ 180^\circ + \phi; & if\ Q = 3, \\ 360^\circ - \phi; & if\ Q = 4. \end{cases} \tag{3.3}$$

Once we estimate the distance and derive the direction of a subpath, the relationship between two consecutive key clicks in the contained subpath is determined. Therefore, if the position of either key click is known, we can derive the position of the other key click according to the derived moving distance and direction. We show an example of distance estimation and direction determination for 6 subpaths $\{46, 28, 37, 64, 82, 73\}$. Figure 3.7 shows the clustering results in both distance and direction when treating the first click as the origin. We observe that each key-click combination is clustered together around the ground truth (shown as the red star) based on our distance estimation and direction determination schemes, indicating that our schemes have the capability to capture the fine-grained hand movement trajectories in key entry activities.

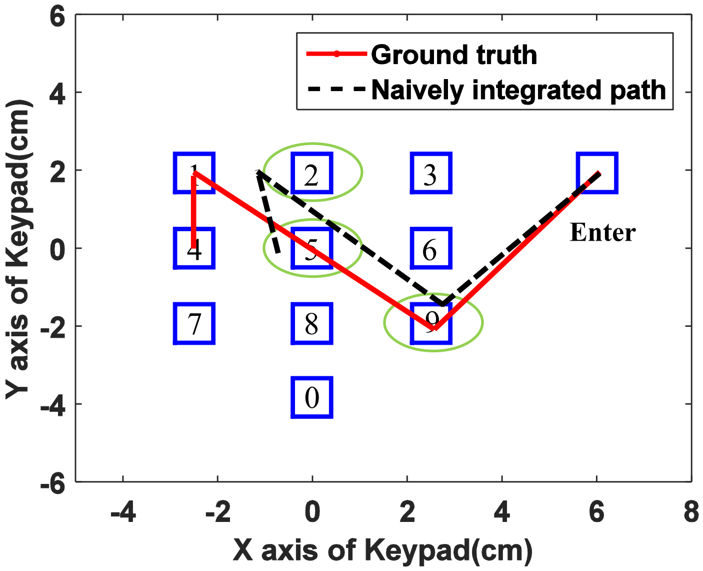


Figure 3.8: Example of the naively integrated trajectory having a large accumulated error cannot correctly map to the key positions of the PIN sequence ”419” (though the estimation error of distance and direction of individual subpath is small).

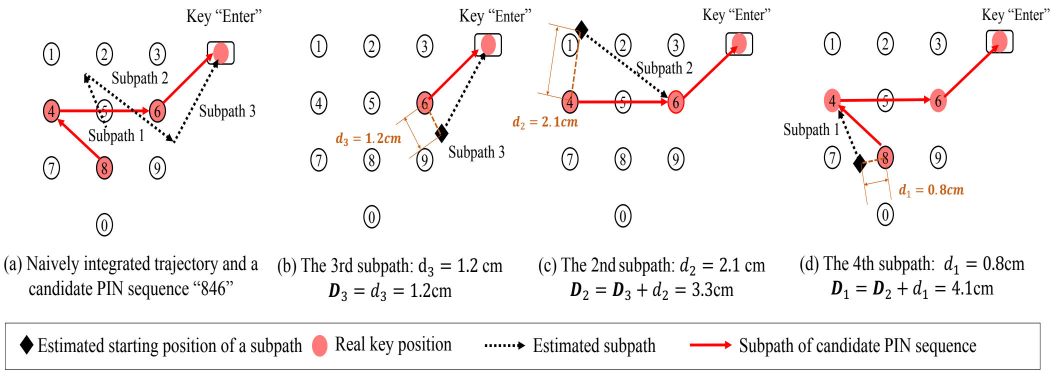
3.6 Backward Pin Sequence Inference Algorithm

After performing *Fine-Grained Subpath Recovery* grounded on distance estimation and direction determination, we next describe how to reconstruct the hand-movement trajectory using the estimated subpaths to infer the target user's PIN sequence.

3.6.1 Backward Subpath Integration

We notice that all key-based security systems require the user to execute the verification by pressing key *Enter* or *Confirm*, which is at a known position on the keypad. We can then utilize this information to reconstruct the hand-movement trajectory on the keypad by examining the subpaths in a backward time sequence. That is, the position of key Enter can be considered as a end of the last subpath, and the starting of the last subpath indicates the position of the last key clicked before key Enter.

More generally, we concatenate the estimated end of the $(j-1)^{th}$ subpath to the starting of the j^{th} subpath and continue to repeat this step until reaching the starting of the first subpath. By integrating all the derived subpaths in such a backward head-tail connecting way, we can obtain a trajectory roughly matching the hand movements during the key-entry process, called the *Naively Integrated Trajectory*. Ideally, the vertices on the Naively Integrated Trajectory



(a) Naively integrated trajectory and a candidate PIN sequence "846"

(b) The 3rd subpath: $d_3 = 1.2$ cm $\boldsymbol{D_3} = d_3 = 1.2$cm

(c) The 2nd subpath: $d_2 = 2.1$ cm $\boldsymbol{D_2} = \boldsymbol{D_3} + d_2 = 3.3$cm

(d) The 4th subpath: $d_1 = 0.8$cm $\boldsymbol{D_1} = \boldsymbol{D_2} + d_1 = 4.1$cm

Figure 3.9: Example of point-wise Euclidean distance accumulation for candidate PIN sequence "846", where the real PIN is "419".

should be mapped to real-key positions with the last vertex mapping to the center of Key Enter.

3.6.2 Point-wise Euclidean Distance Accumulation

Although we can recover each individual subpath based on the estimated distance and derived direction, each subpath contains small errors and the Naively Integrated Trajectory inherits and further accumulates such small errors in each subpath, resulting in mapping to the wrong-key positions on the keypad. Figure 3.8 shows an example that the naively integrated subpaths (i.e. in black dashed lines) cannot recover the correct target user's PIN sequence, e.g., "419", instead, they return "529" as a result. To reduce cumulative errors, we propose a *Point-wise Euclidean Distance Accumulation* approach. In this approach, instead of matching the Naively Integrated Trajectory directly to the keys on the keypad, we consider each subpath separately by comparing the closeness in terms of the Euclidean distance between the starting point of the subpath (i.e., point-wisely) and real key positions, while the ending point of the subpath is fixed on real keys.

In particular, each subpath j contains the estimated distance (S_j) and direction (ϑ_j). Given a real key's position as an ending point (assuming this key is clicked at this ending point), we can estimate the starting point $(\widetilde{x_j}, \widetilde{y_j})$ of each subpath. We conduct this effort in a backward manner starting from Enter key because we know the ending point in the last subpath is the

Enter key. The estimation of the starting point in the j^{th} subpath is obtained as following:

$$\begin{cases} \widetilde{x_j} = cos(\vartheta_j + 180) \times S_j + \mathcal{X}, \\ \widetilde{y_j} = sin(\vartheta_j + 180) \times S_j + \mathcal{Y}, \end{cases} \tag{3.4}$$

where $(\mathcal{X}, \mathcal{Y})$ are the coordinates of ten real number keys $\{1, 2, 3, ..., 9, 0\}$ on the keypad. Given that there are ten real number keys in the key pad, there will be ten estimation results of the starting points in subpath j. We note that, for the last subpath, $(\mathcal{X}, \mathcal{Y})$ is the coordinates of the key Enter. Once the starting point of the j^{th} subpath is estimated, our algorithm will recursively move to the previous subpath. By doing so, we introduce the concept of *accumulated Euclidean distance*, which is the sum of the Euclidean distances between the starting point of a subpath and the coordinate of a real key in the keypad, over all consecutive subpaths. We can recursively run the following equation to calculate the accumulated Euclidean distance:

$$\mathbb{D}_j = \mathbb{D}_{j+1} + d_j, \tag{3.5}$$

where $\mathbb{D}_j$ and $\mathbb{D}_{j+1}$ denote the accumulated Euclidean distance of two consecutive subpaths, respectively, and d_j is the Euclidean distance between the estimated starting point $(\widetilde{x_j}, \widetilde{y_j})$ of the j^{th} subpath and a real key in the keypad. The resulted final accumulated Euclidean distance measures the closeness of the real key combination, defined as *PIN sequence candidate*, to the estimated consecutive subpaths while leveraging the dimension of the keypad. The insight is that we would like to explore the possible candidate keys leveraging the estimation from each subpath without fixing to a particular key matching. In this way, we will not end up with only one Naively Integrated Trajectory, instead, we will obtain multiple key sequences as the candidates for PIN sequence recovery. Furthermore, by conducting the point-wise Euclidean distance accumulation for each candidate of PIN sequence, our algorithm balances the contribution of each estimated subpath and reduce the accumulated errors that impact the accuracy of PIN sequence inference.

Example. Figure 3.9 shows an example of how the Euclidean distance is accumulated point-wisely in backward for a specific candidate PIN sequence “846” (The real PIN entry in this example is ”419”). In the sequence of Figure 3.9, (a) we first generate the Naively Integrated Trajectory consisted of three consecutive subpaths, *subpath* 1, *subpath* 2, and *subpath* 3, which need to be point-wisely compared with the candidate subpaths: “84”,“46”, and “6*enter*” in the candidate PIN sequence “846”. The generation of naively integrated trajectory is based on the estimated distances and derived directions of each subpath. (b) then we start by mapping the ending point of subpath 3 to the key Enter and set $\mathbb{D}_4 = 0$, and utilize the estimated moving

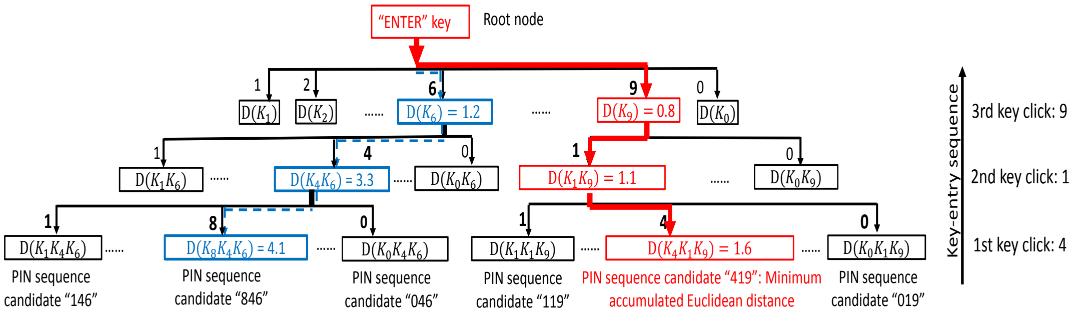


Figure 3.10: Illustration of the construction of the backward trajectory inference tree for recovering PIN "419".

distance and derived direction in the subpath to estimate its starting point on the keypad in a backward way. The Euclidean distance between the estimated starting point of subpath 3 and key 6 (i.e., the 3^{rd} key entry in the candidate PIN sequence "846") is found to be $d_3 = 1.2cm$, and the accumulated Euclidean distance for this subpath is $\mathbb{D}_3 = \mathbb{D}_4 + d_3 = 1.2cm$; (c) next, assuming the ending point of subpath 2 is mapped to key 4, we similarly estimate the starting point of the subpath and calculate the Euclidean distance between the estimated starting point and the position of key 4 (i.e., $d_2 = 2.1cm$). The accumulated Euclidean distance for the previous two supaths is $\mathbb{D}_2 = \mathbb{D}_3 + d_2 = 3.3cm$; (d) lastly, we assume the ending point of the subpath 1 to be key 8 and estimate the starting point of the subpath. We find the Euclidean distance between the estimated starting point and the position of key 8 to be $d_1 = 0.8cm$ and calculate the accumulated Euclidean distance for the entire candidate of PIN sequence "846" as: $\mathbb{D}_1 = \mathbb{D}_2 + d_1 = 4.1cm$. We note that our algorithm recursively calculates the accumulated Euclidean distance for every possible candidate of PIN sequence based on Equations (3.4) and (3.5) and select the candidate with the minimum accumulated Euclidean distance as the final result.

3.6.3 Tree-based Key Sequence Inference

To implement the Backward PIN-Sequence Inference algorithm, we develop a tree-based approach for PIN-sequence inference. Next, we discuss how to build and optimize the tree in our algorithm.

Building a Tree with PIN Sequence Candidates. In order to record and compare different candidates of PIN sequence, we seek to build a decimal tree according to the backward order of all PIN sequence candidates. Each node is defined as a 2-tuple structure containing its corresponding key entry and the Euclidean distance accumulated on the path from the root node to the node, denoted as $< NodeKey, AccuDist >$. Because the tree is built based on a backward order, nodes in the j^{th} level of the tree correspond to the $(N-j)^{th}$ key entries of all candidates of PIN sequences. The root node is always the last key entry (i.e., key Enter), while the leaf nodes are always the first key entry of the candidate of PIN sequence (i.e., number keys on the keypad). Each node (except the leaf nodes) has 10 child nodes corresponding to keys 0 to 9. The branches from one parent node to its child nodes represent the subpaths between the keys corresponding to the parent and child nodes. The leaves of the tree stores the final accumulated Euclidean distance of each candidate of PIN sequence. Our algorithm searches for the leaf node having the minimum accumulated Euclidean distance, and traces back to recover the path from the leaf node to the root node. The inferred PIN sequence is generated by recording the key entries corresponding to the nodes on the recovered path.

Figure 3.10 shows an example of a tree for inferring a PIN sequence of "419", where the accumulated Euclidean distance for one candidate of PIN sequence "846" is $4.1cm$, while another candidate of PIN sequence "419" has the accumulated Euclidean distance of $1.6cm$, which is the minimum over all candidates. Therefore, the candidate of PIN sequence "419" will be determined to be the inferred PIN sequence.

Subpath Calibration and Tree Pruning. In order to improve the accuracy of our system, we take the advantage of the keypad dimension to calibrate subpaths. Intuitively, the distance of a subpath should not exceed the dimension of a keypad. Therefore, if the estimated distance of a subpath exceeds the dimension of a keypad, our system replaces the estimated distance of the particular subpath with the possible longest distance on the keypad. In addition, since every non-leaf node in a PIN-sequence tree has 10 child nodes, the j^{th} level has 10^j nodes. Apparently, it is not necessary to store and calculate the Euclidean distance in every node or sort the accumulated distances of the PIN candidates for the entire PIN space. Our algorithm prunes the tree by keeping the nodes with the least m accumulated Euclidean distances for each tree level. In this way, leaf nodes are largely reduced from 10^N to m, where N is the length of the PIN sequence. In this work, we set $m = min\left(10^j, 100\right)$ in our algorithm for the tree level j, which balances the tree size and algorithm performance. Compared to our algorithm without tree pruning, the running time of our algorithm with tree pruning is reduced from $O\left(2^N\right)$ to $O\left(N\right)$ when N is greater than 2.

Viterbi and Hidden Markov Model based Implementations. We also study to apply the Viterbi algorithm and the Hidden Markov Model (HMM) to solve the PIN inference problem.

We implement two methods, Viterbi and HMM-Viterbi to infer the PIN sequences, both of which also have the running time $O\left(N\right)$, and we compare their performance with our algorithm. 1) In particular, by considering each key button as a state in the trellis diagram and expressing the cost of the path between any two states as $\left|\vec{RealKeyDistance} - \vec{EstimatedSubpath}\right|$, we can then utilize Viterbi algorithm to search the shortest path (i.e., the smallest summation of the path cost for sequential states) in the trellis to infer the PIN sequence. 2) Hidden Markov Model can be applied to model the PIN sequence inference problem, and the dynamic searching in HMM needs to be implemented by Viterbi algorithm [106]. The state transition probability between any two keys can be expressed as $exp\left(-\left|\vec{RealKeyDistance} - \vec{EstimatedSubpath}\right|\right)$, and the PIN sequence decoding problem becomes searching for the sequential states with the highest probability (i.e., the greatest multiplication of the transition probabilities of sequential states). Overall, we find that through performance evaluation in Section 3.8.8, the performances are comparable among the three methods when attacking with one PIN sequence. And the original back PIN sequence Inference algorithm outperforms the Viterbi and HMM-Viterbi when generating optimal PIN candidate list. This is because the Backward PIN Sequence Inference algorithm tests all the most possible PIN sequences and can reflect the best capability of the attack, especially when attacking with more than two PIN sequences on the key-based security system, which usually tolerates multiple tries.

3.7 Implementation

3.7.1 Key-click Detection

Given embedded sensor data from wearable devices, our system first performs key-click detection based on acceleration readings to find the key-click events and the number of keys in a PIN sequence and assist the trace segmentation. Key clicks usually cause significant changes of acceleration towards the keypad that has the potential to be distinguished from other hand movements. In particular, we calculate the magnitude of the composition of accelerations on three axes first, and apply a threshold to examine the normalized magnitude of the composed acceleration to detect key clicks. We empirically determine the threshold to be 0.6 based on our experiments with 20 participants in this work.

3.7.2 Key-click Trace Segmentation

After key-click detection, we roughly segment input sensor data into small chunks containing the data between two consecutive detected key clicks. After segmentation, the resulted small chunks contain the sensor data representing subpaths, which include the acceleration caused by hand movements from one key to another. In addition, to mitigate high frequency noise caused by hand vibration, we apply the $Savitzky - Golay\ filter$ [98] to each chunk of sensor data respectively.

3.7.3 Quaternion-based Coordinate Alignment

When recovering the user's PIN sequence from the wearables' embedded sensors, our system involves three different coordinate systems, namely, *wearable coordinate*, *world coordinate* [2] and *keypad coordinate*. The sensor readings from a wearable are defined within the wearable coordinate and thus cannot be used directly to represent hand movements because of the rotating wearable coordinate caused by frequently changed hand position. In this work, we employ quaternion to help convert sensor readings from the wearable coordinate to keypad coordinate for hand trajectory derivation.

Specifically, we first convert the sensor readings from the wearable coordinate to world coordinate by applying $\vec{a}_w = q_{dw}\vec{a}_d q_{dw}^{-1}$, where $\vec{a}_w$ and $\vec{a}_d$ are the sensor readings in the world coordinate and werable coordinate, respectively, and q_{dw} is the quaternion that represents the conversion from the werable coordinate to world coordinate. Then a_w will be further converted to the keypad coordinate via $\vec{a}_k = q_{wk}\vec{a}_w q_{wk}^{-1}$, where $\vec{a}_k$ denotes the sensor readings in the keyboad coordiante and q_{wk} denotes the quaternion that represents the conversion from the world coordinate to keypad coordinate. The quaternion q_{dw} can be extracted from wearables during hand movements, and q_{wk} can be derived from $q_{wk} = q_{kw}^{-1}$, where the quaternion q_{kw} can be collected by placing a sensor (i.e., smartphone, smartwatch, or IMU) aligned with the coordinate of the target keypad. We note that adversaries can utilize this method to obtain q_{kw} without attention at a time other than the user entering the PIN sequence.

3.8 Performance Evaluation

In this section, we present the experimental methodology and describe the evaluation metrics. We then present the most important results of our system with respect to PIN sequence recovery using the Backward PIN-sequence Recovery Algorithm. Finally, we show the performance of

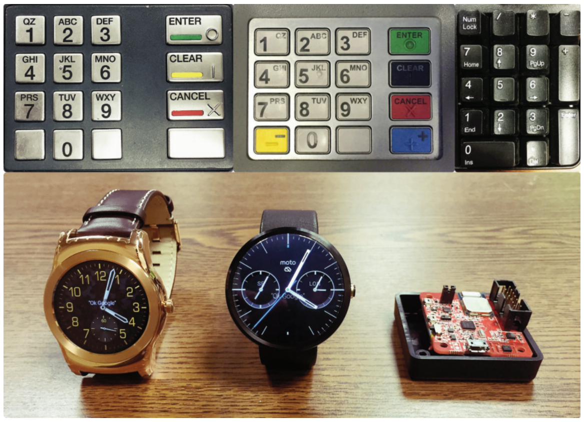


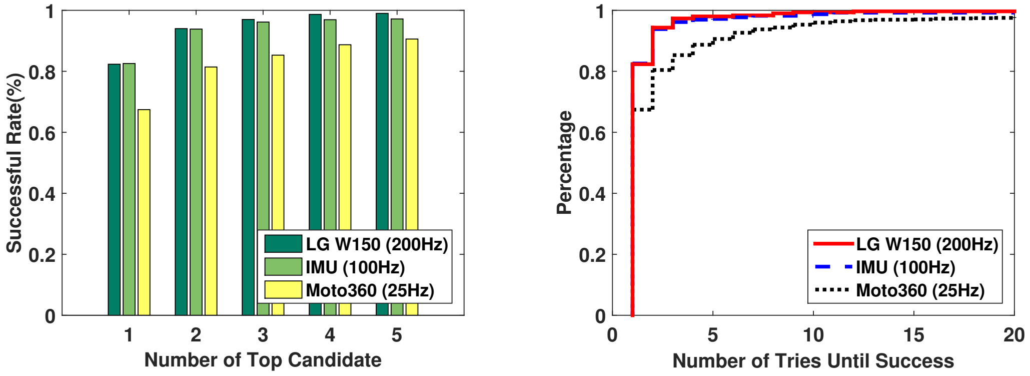
Figure 3.11: Experiments: three different kinds of keypads, detachable ATM pad, keypad on ATM machine, keyboard; and wearable devices.

two supporting schemes for PIN sequence recovery, distance estimation and direction derivation schemes.

3.8.1 Experimental Methodology

Keypads. We evaluate our system with three different kinds of keypads as shown in Figure 3.11: 1) A keypad on ATM machine (from PNC bank) with the dimension of $108mm \times 76mm$; 2) A real detached ATM keypad with the dimension of $127mm \times 95mm$, both 1) and 2) representing the use cases with different ATM pad sizes; and 3) A number pad of Dell USB wired keyboard L100 with the dimension of $77mm \times 97mm$, representing the use case of key-based security access to enterprise servers. The three keypads have different structures and key depths. It is important to evaluate their effects on our approach when capturing fine-grained hand movements. We focus on experiments on numbers to recover PIN-sequences.

Wearable Devices. In our experiments, we use three different types of wearable devices, including two smartwatches (i.e., LG W150 and Moto360) and an IMU (Invensence MPU-9150)[4]. These wearables represent different achievable maximum sampling rates (i.e., 200Hz, 25Hz and 100Hz, respectively). The LG W150 and Moto 360 are two commodity smartwatches running on Android Wear OS with Bluetooth LE. The IMU contains a 9-axis motion tracking sensor designed for consumer electronics. We use it as a prototyping alternative to a wearable device with its sampling rate set to 100Hz. During key-entry activities, the wearable devices collect acceleration and quaternion data and send them to a pre-associated storage device (i.e., smartphone via Bluetooth and laptop via an USB cable for smartwatches and IMU respectively).



(a) Success rate of recovering PIN sequence within top-k candidates.

(b) Cumulative distribution function of the number of tries until success.

Figure 3.12: Performance of Backward PIN-sequence Inference to infer 4-PIN sequences with three kinds of wearables on detachable ATM Keypad.

The ground truth of the hand moving distance and direction is computed through the video recorded by a camera set on top of the keypad. In particular, we use AutoCAD to connect two positions of the sensor in two captured video frames corresponding to the time points when the finger just leaves the first key and about to touch the second key, respectively. The measured distance and angle of the line (with the positive X axis of the keypad) connecting these two sensor positions are used as the ground truth of the distance and direction of the hand movement.

Data Collection. We conduct experiments of various key-entry activities with three different types of wearables on three kinds of keypads. 20 volunteers are recruited to performance key-entry activities over an 13-month period. The volunteers are asked to enter keys in three ways: 1) 4-digit PIN sequences consisting of five consecutive key clicks; 2) 6-digit PIN sequences consisting of seven consecutive key clicks (with "Enter" as the last click) and 3) a single subpath consisting of two consecutive key clicks. For each subpath, based on the keypad layout, we classify different subpath lengths into three representative scales: *short*, *medium* and *long*. Specifically, *short* covers subpaths between two adjacent keys with no keys in between (e.g., 45, 41 and 75); *medium* is for horizontal and vertical subpaths between two keys with one key in between (e.g., 46 and 82); and *long* contains subpaths of two keys neither horizontal nor vertical and with one or more keys in between (e.g., 10, 37 and 29). We collect 7000 PIN sequences from three keypads when having 20 volunteers wear three different kinds of wearables. For single subpath, we collect 3000 subpaths from three keypads including *long*, *medium* and *short* distances with volunteers wearing an IMU.

3.8.2 Evaluation Metrics

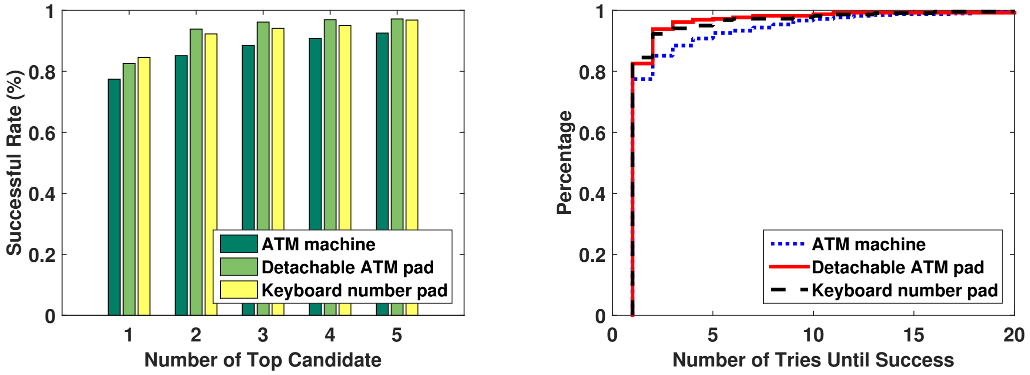
We develop the following metrics to evaluate our system with regard to the accuracy of distance estimation and direction determination schemes and the performance of our Backward PIN-sequence Inference Algorithm:

Distance Estimation Error. To evaluate the performance of our distance estimation scheme, we define the *Distance Estimation Error* as the difference between the estimated distance and the ground truth of the hand moving distance. The ground truth of the hand moving distance is computed through the recorded video during experiments. We study the Distance Estimation Error in two ways: *mean error* and *cumulative distribution function (CDF).*

Direction Classification Accuracy. To evaluate the performance of our direction derivation scheme, we divide the 360° on the X-Y plane into 16 groups (i.e., 5 groups in each quadrant excluding 4 overlapped groups) and examine whether the derived direction is classified into the same group as that of the corresponding ground truth. The ground truth of angles is also computed through the recorded videos. The *Direction Classification Accuracy* is $\frac{\tilde{N}_c}{N_c}$, where $\tilde{N}_c$ is the number of directions have been classified into the same group containing the corresponding ground-truth direction, and N_c is the total experimental runs of direction classification.

Top-k Success Rate. Given an experimental run of a key-entry activity, our algorithm could return multiple top candidates of key-entry sequence in an ascending order of the accumulated Euclidean distance. We define that the inference algorithm is a *Top-k Success Hit* if the first k candidates of key-entry sequence returned from our algorithm contain the target user's key-entry sequence. We further define the *Top-k Success Rate* as the ratio ($\frac{\widetilde{N}_s^k}{N_s}$) of the number of Top-k Success Hits ($\widetilde{N}_s^k$) over the total number of experimental runs (N_s) when applying key-entry sequence inference to recover the target user's PIN sequence. Specially, when $k = 1$, the ratio indicates the rate of our algorithm that can successfully determine the target user's key-entry sequence without ambiguity.

Tries Until Success. Since our system can provide multiple candidates as the result for key-entry sequence inference, the adversary has the chance to try out each key sequence returned in the candidate list to recover the target user's PIN sequence. We define the *Number of Tries Until Success* as the number of candidate key-entry sequence the adversary has tried (starting from the candidate with the smallest accumulated Euclidean distance) until he/she breaks the key-based security system, suggesting a success recovery of the target user's PIN sequence. Thus, the Number of Trails Until Success indicates the possible efforts that an attack needs to take to break the key-based security system.



(a) Success rate of recovering PIN sequence within top-k candidates.

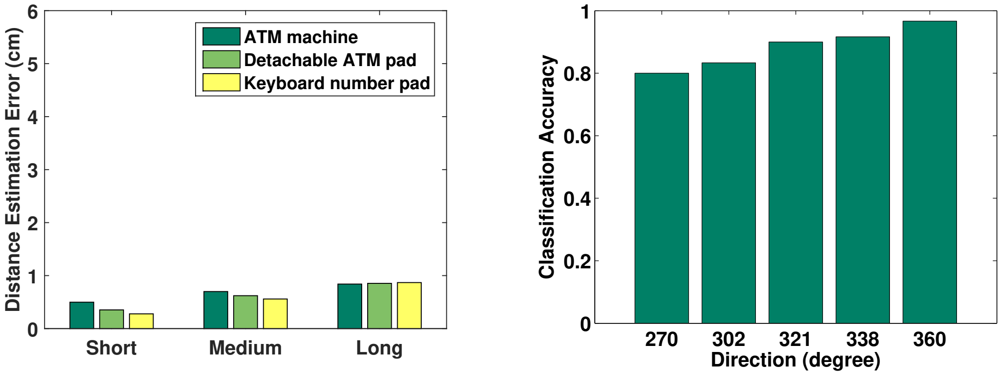
(b) Cumulative distribution function of the number of tries until success.

Figure 3.13: Performance of 4-PIN sequence inference on three different keypads by using medium sampling rate 100Hz (IMU).

3.8.3 Performance of Backward PIN-Sequence Inference

Wearable Devices. We first examine the performance of our Backward PIN-sequence inference algorithm to infer 4-PIN sequences on the detachable ATM keypad with three different wearable devices. Figure 3.12(a) shows the top-k success rate of our system from three different types of wearable devices. We find that our system can effectively recover 4-PIN sequences from all the three wearables, and higher success rate is achieved under higher sampling rates. In particular, by choosing the top-1 choice, our system can achieve over 82% success rate for the LG W150 and IMU, while the success rate is 67% for the Moto 360. Furthermore, the PIN sequences can be inferred with increasing success rates if the adversary utilizes more choices from the top-k candidate list. Specifically, when using the top-2 choices, the adversary can achieve about 94% success rate with the LG W150 and IMU, and the success rate for the Moto 360 is over 80%. Although the Moto 360 achieves lower success rates than the LG W150 and IMU due to its much lower sampling rate (i.e., 25Hz), an adversary can still achieve a high probability to reveal the PIN sequences based on top-2 or 3 choices. This indicates that our system can tolerate the insufficient information introduced by wearable devices with low sampling rates.

Figure 3.12(b) depicts the cumulative distribution of the number of tries until successfully recovering the user's 4-PIN sequence from three wearables. We find that the adversary can break over 97% PIN entries from the LG W150 and IMU within 5 tries, which is usually the maximum PIN tries on ATM machine. The number of PIN entries revealed increases to 99%, if the attacker conducts 10 tries. For Moto 360, the attacker can break 90% PIN entries within 5 tries and 96% within 10 tries. Therefore, regardless of the types of wearable, the attacker



(a) Distance estimation error of three kinds of keypads.

(b) Direction classification results of ATM machine.

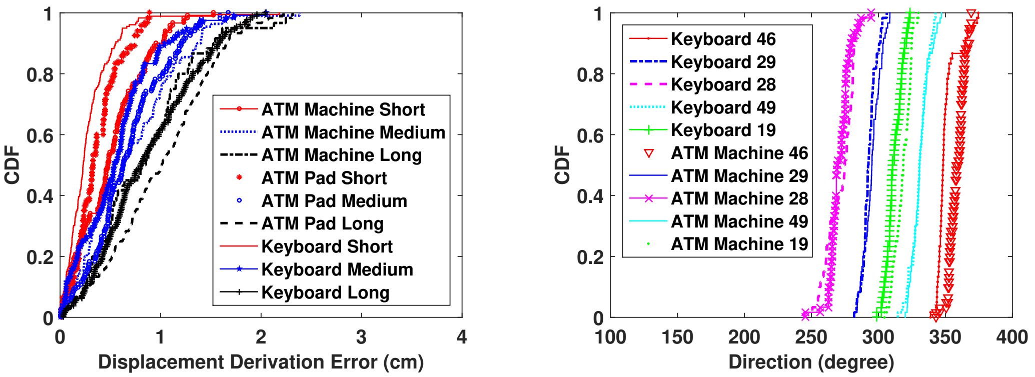
Figure 3.14: Distance estimation mean error and direction classification results between two consecutive key clicks under 100Hz sampling rate (IMU).

can break the user's PIN sequence with few tries. Although the LG W150 is set to use 200Hz sampling rate and generates the best performance, we find that using 100Hz sampling rate is enough to achieve comparable good results. Therefore, we present the results using the IMU for the rest sections.

ATM Keypads and Keyboard. Figure 3.13(a) shows the top-k success rate to recover 4-PIN sequences on three keypads. We observe that our system can achieve around 80% success rate for all three keypads with the top-1 choice. When using the top-5 choices, our system can achieve over 97% success rate with both of the detachable ATM pad and the number pad on keyboard, while on real ATM machine, the success rate is over 92.5%. Figure 3.13(b) confirms our observation in Figure 3.13(a). The results demonstrate that our Backward PIN-sequence Inference is effective when applied with keypads of different layouts and coordinates. The success rate is higher with both of the detachable ATM pad and the number pad on keyboard than that with the ATM machine. Our results suggests that the electronic magnetic field and the tilt angle of the ATM machine affect the PIN entry recovery result on ATM machine.

3.8.4 Distance Estimation of Different Kinds of Keypads

We next study the performance of two supporting schemes. The study of the distance estimation scheme is described in this subsection, and the results of the direction determination scheme is presented in the next subsection. We apply our distance estimation scheme to various subpaths across three different kinds of keypads. We compare the distance difference between ground truth (i.e., obtained from camera) and the estimated distance from sensor data. Take ATM



(a) Cumulative distribution function of distance estimation errors.

(b) Cumulative distribution function of direction derivation errors.

Figure 3.15: Performance of distance estimation and direction derivation on three kinds of keypads under 100Hz sampling rate (IMU).

machine as an example, the distances for *short*, *medium* and *long* are 2.5cm, 5cm and 6.4cm, respectively.

We observe that the mean error is proportional to the distance scale, i.e., short distance has relative smaller error compared with long distance, as shown in Figure 3.14(a). In particular, the mean error of ATM machine for short, medium and long distance are 5mm, 7mm and 8.5mm, respectively. For detachable ATM pad, the error of long, medium and short distance are 8mm, 6mm and 3.5mm, respectively. The mean error of long distance in keyboard number pad experiment is 8mm, 5mm for medium distance and for short distance the error is as low as 3mm. The experiment results from keyboard shows relative smaller distance error since the physical layout of keyboard number pad is smaller than ATM machine keypad and detachable ATM pad. We observe that such error difference is marginal and reveal the effectiveness of our scheme.

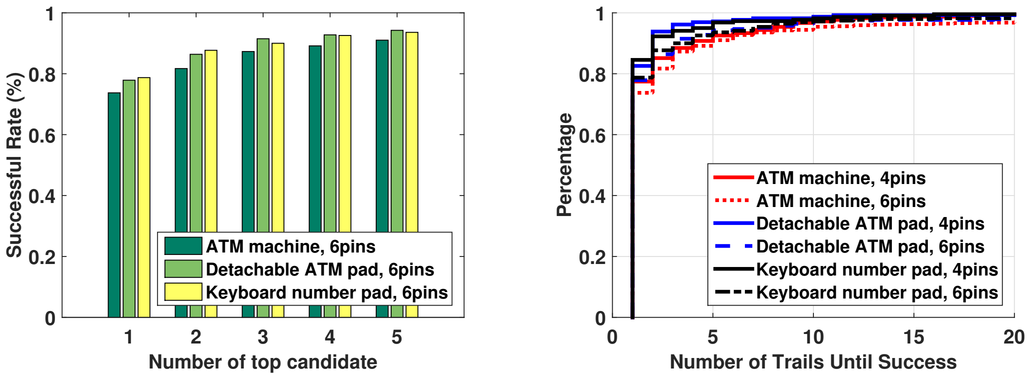
Figure 3.15(a) shows the cumulative distributive function of distance estimation errors. We observe that the 80th percentile errors are 8mm, 10mm and 12mm for short, medium and long distance of ATM machine, respectively. For detachable ATM pad the 80th percentile error are 5mm, 10mm and 13mm, receptively and the 80th percentile error of number pad experiment are 4mm, 8mm and 13.2mm respectively. The results also show the effectiveness and robustness of our scheme under various keypads.

3.8.5 Direction Derivation of Different Kinds of Keypads

Next, we evaluate our slope-based direction derivation scheme by showing the performance under three different kinds of keypads. According to the keypad layout, we select five representative directions in one quadrant. Take ATM machine as an example, the five directions within the fourth quadrant are: keys 2 to 8, keys 2 to 9, keys 1 to 9, keys 4 to 9 and keys 4 to 6. The corresponding direction angle for these subpaths on the keypad are: 270° , 302° , 321°, 338° and 360°. To evaluate our direction derivation scheme, we study the direction classification accuracy of classifying the directions of testing subpaths into the aforementioned 5 groups of directions angles. Figure 3.14(b) shows the direction classification accuracy with five directions on ATM machine. The X axis represents the ground truth direction between two keys on the ATM machine. We find that there are few subpaths mistakenly classified as incorrect direction. In particular, our scheme can achieve 80% classification accuracy for 270° and we observe that directions with larger angles have better accuracy, which is up to 97% accuracy for 360°. This may due to that when user performs vertical key clicks (e.g., key 2 to 8 with 270° on ATM pad), there might be a small inclined angle between hand moving direction and wrist moving direction. For keyboard and ATM pad, we have similar high classification accuracy. In addition, Figure 3.15(b) shows the cumulative distribution function of estimated five directions in the fourth quadrant. We find that all five directions obtained from our scheme only have small overlap for any two adjacent directions. Moreover, 90% of the derived direction are close to the ground truth direction within ±10°. The above results show that our system provides effective distance estimation and direction derivation schemes under various keypads and is robust in real environments.

3.8.6 Impact of PIN-sequence Length

Because longer-PIN-sequence inference is more likely to be affected by the errors of deriving hand movement trajectory, we examine the impact of the PIN-sequence length to the performance of the Backward PIN-sequence Inference Algorithm. Figure 3.16(a) shows the top-k success rate of recovering 6-PIN sequences on three different kinds of keypads by using the IMU collecting data at 100Hz. We find that our system achieves around 80% success rate of revealing 6-PIN sequences on all three keypads using the top-1 choice. When trying with the top-5 choices, our system achieves around 93% success rate on the three keypads. As we can see that the results are very similar to those of inferring 4-PIN sequences in Figure 3.13, indicating that the Backward PIN-sequence Inference algorithm is robust to different lengths of PIN sequences.



(a) Success rate of recovering PIN sequence within top-k candidates.

(b) Cumulative distribution function of the number of tries until success.

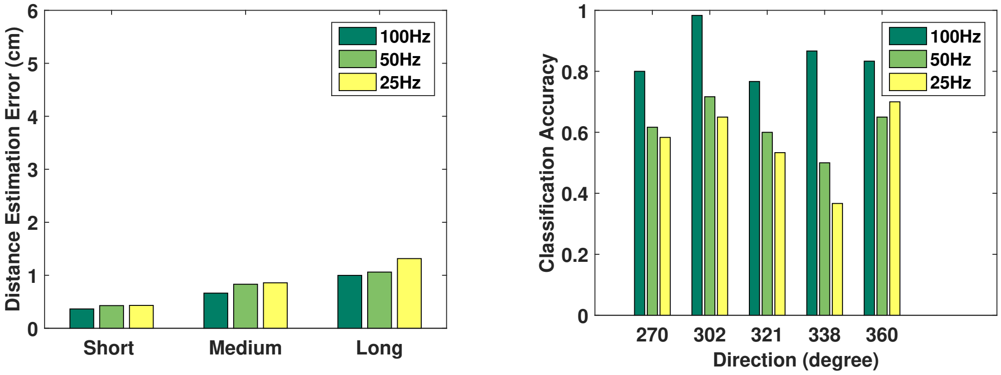
Figure 3.16: Performance of 6-PIN Sequence Inference on three different keypads by using medium sampling rate 100Hz (IMU).

Figure 3.16(b) depicts the cumulative distribution of the number of tries until successfully recovering the 6-PIN and 4-PIN sequences on the three keypads, respectively. We observe that our system can successfully break around 80% 6-PIN and 4-PIN sequences with one try and over 96% 6-PIN and 4-PIN sequences with 10 tries. The results show that the PIN inference performance of our system is consistently good for different PIN sequence lengths, because our Backward PIN-sequence Inference algorithm does not accumulate errors in recovering subpaths.

3.8.7 Impact of Sampling Rate

We then study the impact of the sampling rate of wearables to our system. Figure 3.17(a) shows the mean errors of the estimated distances between two consecutive key clicks on the detachable ATM pad with the IMU sampling at 25Hz, 50Hz and 100Hz, respectively. We find that higher sampling rates generate slightly smaller errors for the short, medium and long distances. In particular, the mean errors for short, medium and long distances are 4.3 mm, 8.5 mm and 13.1 mm when the sampling rate is 25Hz, And when the sampling rate is 50Hz, the mean errors are 4.2 mm, 8.3 mm and 10 mm, respectively. In addition, Figure 3.17(b) shows the direction classification results of our slope-based direction derivation scheme under various sampling rates. Although lower sampling rates cause lower accuracy of direction derivation results, our system still recovers over half of the moving directions correctly, which indicates that our system can recover hand trajectories with good performance at lower sampling rates.

We further evaluate the impact of different sampling rates to our Backward PIN-sequence Inference algorithm . Figure 3.18(a) depicts the performance of 4-PIN and 6-PIN sequence



(a) Distance estimation error of PIN sequence within top-k candidates.

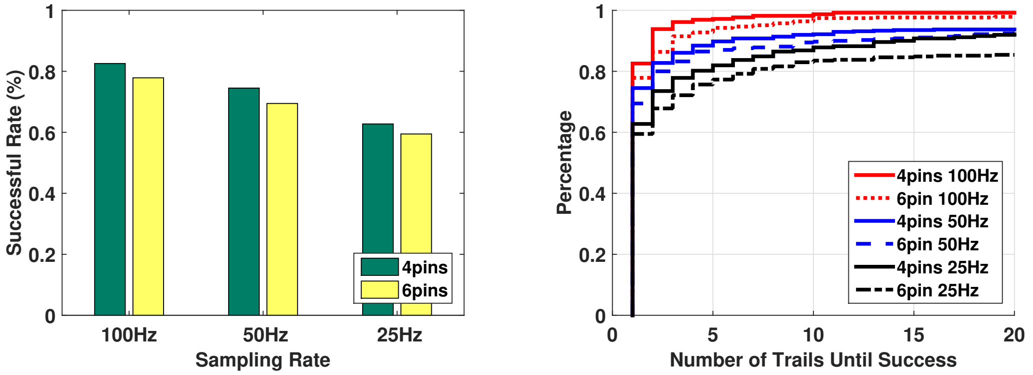
(b) Cumulative distribution function of the number of tries until success.

Figure 3.17: Distance estimation mean error and direction classification results between two consecutive key clicks with IMU under 100Hz, 50Hz and 25Hz sampling rate.

inferences with top-1 choice on the detachable ATM pad with the IMU sampling at 25Hz, 50Hz and 100Hz. We find that our system can achieve over 70% and 60% accuracy in inferring both 4-PIN and 6-PIN sequences at 50Hz and 25Hz sampling rate, respectively. Figure 3.18(b) further confirms our observation in Figure 3.18(a). Moreover, we find that adversaries can achieve very high accuracy of revealing PIN sequences when trying more choices under low sampling rates. Specifically, our system can successfully break over 90% 6-PIN and 4-PIN sequences with 5 tries under $50Hz$ sampling rate and around 80% with 5 tries under 25Hz. The results demonstrate that our system can effectively reveal PIN sequences when using various sampling rates. Even the sensor data sampled at 25Hz has very high probability to leak the user's PIN sequences.

3.8.8 Performance Comparison among Three Algorithms

Finally, we compare the performance of the three methods Backward PIN Sequence Inference, Viterbi algorithm and HMM-Viterbi. Figure 3.19(a) shows the top-k success rate of the three algorithms in revealing 6-PIN sequences. As expected, Viterbi algorithm and HMM-Viterbi are efficient to find the optimal PIN sequence (i.e., top-1 candidate) from the estimated PIN entry trajectory. In particular, Viterbi algorithm and HMM-Viterbi achieve the same accuracy of 82% as the Backward PIN Sequence Inference algorithm in finding the top-1 result. Moreover, the Viterbi algorithm and HMM-Viterbi have much lower accuracy than the Backward PIN Sequence Inference algorithm when trying more than one PIN candidates. The results show that HMM-Viterbi and Viterbi are not as good as Backward PIN Sequence Inference algorithm for investigating the multi-try problem in practical attack, and thus cannot fully reflect the



(a) Success rate of recovering PIN sequence within top-1 candidate.

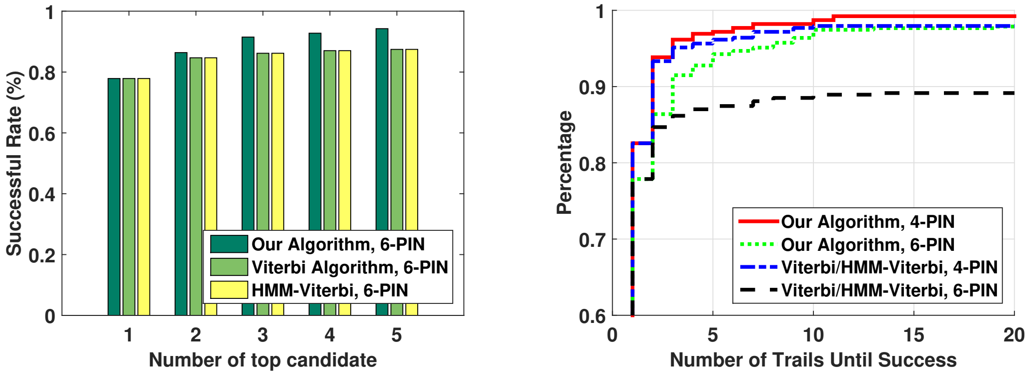
(b) Cumulative distribution function of the number of tries until success.

Figure 3.18: Performance of 4-PIN and 6-PIN Sequence Inference on detachable ATM pad by using IMU under 100Hz, 50Hz, 25Hz sampling rate.

attackers' capability. Note that the HMM-Viterbi and Viterbi algorithm have the similar performance, which is because that Viterbi algorithm is applied in HMM to solve the decoding problem of HMM [106]. We further compare the Tries Until Success among the three algorithms. Figure 3.19(b) shows that Backward PIN Sequence algorithm outperforms the other two algorithms for both revealing 4-PIN and 6-PIN sequences and shows much more success rate enhancement to reveal PINs of longer lengths (e.g., 6-PIN). The above comparisons show that the Backward PIN Sequence algorithm not only finds the optimal PIN sequence efficiently but also provides the optimal candidate list, which provides more comprehensive understanding of the PIN leakage issues on the key-based security system from wearable devices.

3.9 Discussion

Wearing the Wearable Device on the Left Hand or Right Hand. Our training-free approach does not require mirroring the derivation from sensor data when applied to either the left-handed or right-handed user since the inherent physics of key entry activities will be preserved regardless of either case. We assume the victim use either hand wearing a wearable (i.e., a smartwatch or fitness tracker) to access key-based security systems. While it is very difficult to know the exact number of how many people sharing this style, we instead discuss the population of the potential wearable user victims. We take the right-handed user for discussion as the left-handed user share the same conclusion. Wearable devices are usually designed in a way that allows users to comfortably wear them on either wrist (e.g., smartwatches no longer



(a) Success rate of recovering PIN sequence within top-k candidates.

(b) Cumulative distribution function of the number of tries until success.

Figure 3.19: Performance Comparison between different algorithms to infer PIN sequence with IMU under 100Hz on detachable ATM Keypad.

necessarily have crowns as traditional watches do). There are many smartwatch users [9, 5] claiming that they wear smartwatches on their right wrists. Furthermore, for those wearing traditional watch on the left wrist, they tend to wear fitness tracker on the right wrist for health-related applications. Naturally, the right-handed people use their right hand to perform key entry and the sensors in their smartwatches or fitness trackers can be utilized by our approach to reveal PINs. Given the growing cheaper price of these wearable devices, many people wear both a smartwatch and a fitness tracker on separate hand to better serve their work and health applications, which further increases the number of potential victims. Lastly, the increasing popular usage of wearables leaves adversary great chances to recover the user's sensitive information, making it vulnerable irrespective of the hand on which it is worn.

Using Sensor Moving Direction as Hand Moving Direction. We discuss the rationality of using sensor moving direction as hand moving direction. The current system is designed for recovering a PIN sequence by reconstructing hand movement trajectories. We leverage embedded sensor readings from wearable devices on a user's wrist to determine the direction. We use the sensor movement to represent the hand movement since the hand and the wrist are moving together. During our extensive experimental study, we observe that sensor movement and hand movement share similar moving trend. Therefore such a representation is reasonable.

The Trend of Sniffing Attacks. Based on the survey of over 15 wearable devices, we understand that the smartwatches can transmit raw sensor data to the mobile device. The fitness trackers transmit aggregated or simplified data to synchronize with mobile devices, because current fitness trackers only aims at providing coarse-grained information for the applications, such as step counting and activity tracking. The findings make the sniffing attack possible to obtain the sensor data directly. Furthermore, with the growing demands for more powerful body sensors to enable pervasive applications, such as health care, activity recognition, and human computer interaction, we envision that the fine-grained sensor data from fitness trackers is both necessary and useful to support these applications, and the sniffing attacks will remain as unneglectable threats.

Defending Strategies. Existing studies suggest to decrease embedded sensors' sampling rates (e.g., under 50Hz) to mitigate the attack through smartwatches [118]. Our system shows that users PINs can still be revealed from wearables with such low sampling rate (e.g., 50Hz and lower). The reason is that the strong distinct motion during the PIN entry can be captured by the wearable sensor even under low sampling rates. Moreover, our system can recover the fine-grained PIN input trajectory to reveal the PIN sequences. Furthermore, we show that using longer PINs (e.g., 6-PIN sequences compared to 4-PIN sequences) cannot diminish the possibility of leaking the PIN information from wearables regardless its greater password strength [3]. Future countermeasures may aim at camouflaging the sensitive sensor data transmitted from wearables to host devices. For instance, a wearable can inject a certain type of noise to its sensor data (e.g., quaternions and accelerations) so that the data cannot be used to derive fine-grained hand movements while still effective for many common applications (e.g., activity recognition and step counting). Moreover, more secure schemes can be designed to protect the access and transmission of sensor data. That is, advanced encryption schemes are necessary to protect the raw sensor readings in wireless communication, and the access to sensor data should be regulated according to difference scenarios and applications by the wearable or its host device's operating system to avoid leakage.

3.10 Summary

In this work, we showed that the embedded sensors on wrist-worn wearable devices (i.e., smartwatches and fitness trackers) can be exploited to discriminate mm-level distances of the user's fine-grained hand movements during key-entry activities, exposing the user to a serious security breach. We presented a PIN-sequence inference framework to recover the user's secret key entries when the user accesses key-based security systems such as ATM keypads and regular keyboards. The implemented system does not require any training or contextual information, which makes it applicable in real world adversarial contexts. In particular, our system exploited the physics phenomenon and unique patterns of key entry activities from the sensor data and developed distance estimation and slope-based moving direction derivation schemes to capture the small hand movement between two consecutive keys. Our system further applied the Backward PIN-sequence Inference algorithm to reveal the user's complete PIN sequence, leveraging both the spatial and temporal constraints of the key entry to achieve a high success rate. Extensive experiments involving 20 volunteers on three different types of keypads over 13 months showed that our system can achieve 80% accuracy in revealing the user's PIN sequences with one try, and over a 90% success rate within three tries, while recovering the hand movement trajectory has a mean error as low as $6mm$. Such a technique kept a consistent performance in revealing long PIN sequences (e.g., 6-PIN sequences) and could still achieve a very high accuracy under very low sampling rate of embedded sensors(e.g., 25Hz).

Chapter 4
Securing Voice Assistants using Wearable Devices

4.1 Background

In recent years, smart devices (e.g., Google Home and Amazon Alexa) have incorporated advanced speech recognition technologies that enable the devices to understand natural language and take voice commands. By using voices as inputs, users can smoothly and conveniently interact with their voice assistant (VA) systems to accomplish numerous daily tasks. In particular, such a convenient function has been quickly adopted by users and widely used in various applications (e.g., playing music, managing calendar events, shopping online and controlling smart home appliances). As a result, VA systems have already been widely used in various scenarios, such as home, workplace and even public places.

While the VA systems bring immense flexibility and convenience to users, the highly sensitive information collected by these systems could attract an adversary's interests and put the user's privacy under high risks. For instance, the adversary can easily learn the user's schedule such as when to pick up his/her kid from daycare by asking the VA system "What is my schedule to pick up my son". Similarly, the user's private travel schedule such as when to attend a conference can also be easily revealed by requesting "Remind me which day to attend the Machine Learning conference". Therefore, both of the user's family and personal sensitive information can be obtained by simply asking one question. Furthermore, the adversary can even request the voice assistant to execute commands that are against the user's will. For example, the adversary can place online orders through the user's associated account without knowing the credit card information by telling the VA system "Order a MacBook from Prime Now", and then he can wait at the user's address to pick up the delivery. When the adversary can access the VA system at home remotely (e.g., through a hacked Smart TV), he can even use the voice command "Unlock the exterior door" to unlock the door's smart locking system and gaining entry into the house.

Existing VA systems have deployed the voice biometric technology (at least to authenticate "wake words"), however, this approach identifies users based on their unique acoustic features solely in the *audio domain* (i.e., extracting information from the data captured by microphones).

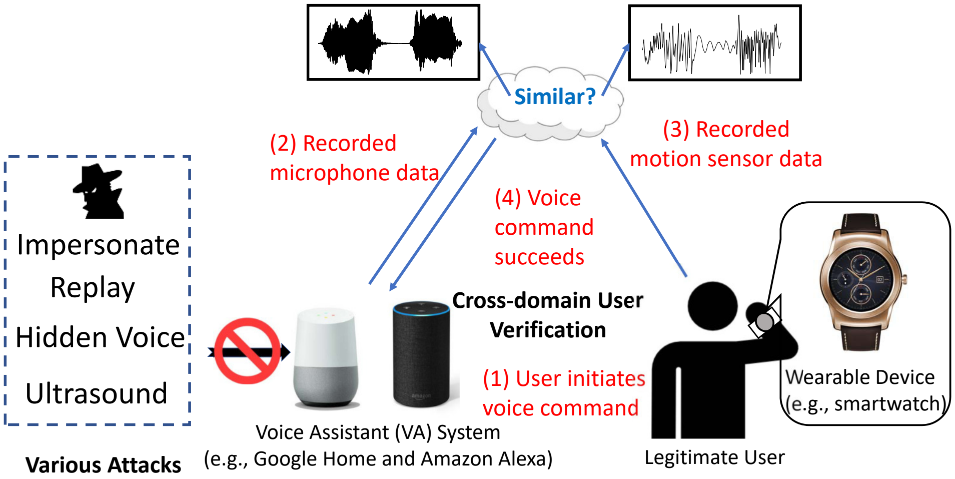

Similar?
(2) Recorded microphone data
(3) Recorded motion sensor data
(4) Voice command succeeds
Cross-domain User Verification
(1) User initiates voice command
Impersonate
Replay
Hidden Voice
Ultrasound
Various Attacks
Voice Assistant (VA) System
(e.g., Google Home and Amazon Alexa)
Legitimate User
Wearable Device
(e.g., smartwatch)

Figure 4.1: Proposed architecture of WearID.

The acoustic features are known to be vulnerable to *impersonation attacks* and *replay attacks*, where the adversary can fool the systems by imitating the legitimate user's voice [112] or via a simple record-and-replay of the user's voice commands [75]. Moreover, recent studies show that hidden voice attacks and ultrasound attacks could access VA systems surreptitiously even when the legitimate user is present near the VA device [43, 134].

To address the above vulnerabilities underlying the VA systems, we propose a wearable-assisted VA user authentication system, *WearID*, which performs cross-domain authentication on the user's voice commands by leveraging the wearable device as an additional factor. This is motivated by the wearable's nature of being used as a security token [88] and the already huge wearable user number (i.e., reaching 593 million in 2018 [109]). WearID utilizes the low-cost motion sensors embedded in the user's wearable device to capture the unique voice characteristics in the vibration domain, which is compared to the traditional audio domain voice captured by the VA device's microphones to verify the voice commands from various audio attacks. The flow of WearID is illustrated in Figure 4.1. Our system simply uses the regular *wake word* (e.g., "OK Google") to trigger the authentication process. The voice command is then captured by the microphone of the VA system in the audio domain and the accelerometer of the wearable device in the vibration domain, respectively. We develop a training-free algorithm to perform the cross-domain comparison and the software processing component is deployed in the VA system's cloud service to process the sensor data for user authentication. If the similarity is high, the system accepts the voice command as from the legitimate user. Otherwise, the system rejects the voice command and sends a warning message to alert the user. Our solution can be easily integrated into existing VA systems and wearable devices and does not need special hardware or modifications to VA systems. Moreover, our system verifies the user automatically and does not require cumbersome user operations such as virtual buttons on wearables [53].

Recent studies show the initial success of using motion sensors on the smartphone to capture the speaker's voice. For instance, Gyrophone [85] presents that gyroscope can capture the acoustic signals from an external loudspeaker and reveal the speaker information (e.g., gender and identity). Accelword [135] uses the smartphone's accelerometer to detect human voice for *wake word* recognition. Speechless [33] identifies the condition of using the smartphone motion sensor to capture sound: the shared hard surface between the external speaker and the smartphone. However, implementing WearID in practical scenarios using motion sensors in wearable devices to enhance the security of VA system is a challenging task. *First*, the vibration domain information provided by the motion sensor and its unique acoustic characteristics remain unclear. *Second*, the high-sampling-rate microphone data (e.g., 8kHz and 44.1kHz) and the low-sampling-rate motion sensor data (e.g., 200Hz) are not directly comparable, the relationship between two distinct sensing modalities must be determined for a reasonable comparison. *Third*, the synchronization of the two data sets from totally different hardware is difficult. *Fourth*, the proposed system should defend against various audible impersonation and replay attacks [112, 75] and inaudible attacks [43, 134].

Toward this end, we explore the feasibility of leveraging the wearable's motion sensor to harness the aerial voice vibrations corresponding to live human speech. To ensure reliable cross-domain comparison, WearID develops a spectrogram-based method to convert the microphone data into low frequency aliased signals, making it comparable to the real motion sensor readings. We extensively study the unique response distance and characteristics of the motion sensors in wearable devices and identify the complex relationship between the two sensing modalities to facilitate the data comparison. Our system is designed to maximize the usage of motion sensors' response in the frequency domain and focus on the acoustic signals with the frequencies and amplitudes that are perceivable to motion sensors during the microphone spectrogram conversion. WearID leverages the VA system's wake word to trigger the verification process and start data collection on the VA and wearable devices simultaneously. To trigger the data collection on the wearable device, WearID utilizes two alternative approaches based on WiFi communication or accelerometer-based wake-word detection and coarsely synchronize the two different sensing modalities. We develop a shift 2D-correlation method, which shifts the spectrogram of the two sensing modalities' readings within a short time window to reduce the residual synchronization errors and obtains the maximum 2D correlation to describe the cross-domain similarity. In addition, WearID calibrates the data to remove the vibration noises (e.g., hand motions) and identify precise command sound segment during data preprocessing. This proposed system eventually reveals the unique relationship between the two types of signals, which presents a voice command crossing two domains making it hard to be forged by adversaries in various attacks including audible impersonation and replay attacks and inaudible attacks.

Our Contributions:

- We find that human voices can be captured over the air by the motion sensors embedded in wearable devices. This could serve as an additional domain (i.e., vibration domain) to the original audio domain to verify the user and secure the VA system.
- We propose a unique cross-domain user verification system, WearId, which can be easily integrated with the existing VA systems and wearable devices without making any hardware modifications and requires minimum user effort.
- We leverage the motion sensor's short response distance to voice to effectively prevent the impersonation and replay sounds from accessing the wearable. We derive the unique spectrogram relationship between two sensing modalities (i.e., microphones and motion sensors) to provide enhanced user verification using wearable devices.
- We conduct extensive experiments and user studies with different models of smartwatches and participants, which result in 600 human voice segments. The results show that WearID can authenticate user's voice commands with 99.8% accuracy in the normal situation and detect 97% of various impersonation and replay attacks with a low false negative rate of 2%. When under the hidden voice and ultrasound attacks [134], WearID achieves close to 100% accuracy of verifying the users.

4.2 Related Work

Audio-domain Voice Authentication and Security Issues. The traditional user authentication methods designed for voice access systems mainly extract each individual's voice features in the audio domain to identify users [126, 65, 63, 115, 42, 95]. Mel-Frequency Cepstral Coefficients (MFCCs) [87] and Spectral Subband Centroids (SSCs) [69] describe a voice's timbre and vocal-tract resonances and are widely used as unique voice features to distinguish users. The modulation frequency [34] capturing formant and energy transition details of a voice sound contains speaker-specific information for user identification. However, only relying on the audio-domain features has been shown to be vulnerable to acoustic-based attacks. For example, an adversary can spoof the legitimate user to pass a voice authentication system by recording and replaying a user's voice sound [75]. Moreover, the adversary can study the user's daily speech to impersonate or synthesze the user's voice to pass the voice authentication [112, 75, 54, 54].

WearID Versus Other Authentication Methods. To defend against the replay and impersonation attacks, researchers show that advanced speaker models, Gaussian Mixture Model and i-vector models [32, 62], and the speech features, relative phase shift and modulation features [55, 131] could be used to secure the voice authentication systems. However, these solutions solely use the features from the audio domain, which are still vulnerable to audio-based attacks because the attackers could easily gain the knowledge of these features for forgery. Rather than voice features, more researchers propose to determine the liveness of the sound source by exploiting the physical features of human speeches [45, 137, 136]. Specifically, Chen *et al.* [45] examine the unique magnetic field patterns generated by electro-acoustic transducers to detect loudspeaker generated voice. VoiceLive [137] and VoiceGesture[136] detect the dynamic acoustic characteristics (via time-difference-of-arrival and Doppler shifts) that only occur in human voices to identify liveness. However, these approaches are focusing on smartphone and require much user effort to place the smartphone microphone close to mouth. Thus they are not applicable to the VA systems (e.g., Google Home and Amazon Alexa) that allow users to give voice commands freely from distance. Feng *et al.* [57] develop a user verification system for the VA systems by capturing the user's facial vibrations via an accelerometer embedded in a pair of glasses. The vibrations are then compared with the voice recorded by the VA system to verify whether the voice command is given by the legitimate user wearing the glasses. However, this approach requires the user to wear a dedicated device with a high sampling-rate accelerometer and needs to modify the VA device hardware.

Vibration-domain Voice Recognition. Recent studies show that the MEMS motion sensors (e.g., accelerometer and gyroscope) are able to capture acoustic sounds [85, 135, 50]. Gyrophone [85] utilizes the gyroscope in a smartphone to recognize the speaker's information (e.g., gender and speaker identity) from the speech played by a loudspeaker. Accelword [135] leverages the accelerometer in a smartphone rather than a microphone to recognize the user's wake word sound(e.g., Siri), which reduces the energy consumption. Speechless [33] further analyzes the speech privacy leakage including the speech content from the smartphone motion sensors under various attacking scenarios. These works require much effort to train the system with motion sensor data and do not reveal the relationship between the sensor readings and real voice recorded by microphones. Moreover, the acoustic impact to the motion sensors in wearable devices attached to human bodies is still unexplored.

4.3 Vulnerability of Voice Assistants

4.3.1 Potential Security Breaches in VA Systems

While VA systems bring great convenience and flexibility to users, the open nature of voice access allows anyone to use VA systems via sounds, causing serious security breaches. We separate commodity VA systems into two categories, the personal VA systems (integrated into mobile devices) and the family/community shared VA systems (standalone VA devices). The details of these two categories are provided in Section 4.3.2. In this work, we focus on the standalone VA systems such as Google Home and Amazon Alexa, which can be deployed in home or office and accessed by multiple people. We find that an adversary can obtain the user's private information (e.g., personal schedules and email contents) or conduct unauthorized operations (e.g., control smart appliances like lights and doors) as introduced in Section 4.3.3.

4.3.2 Two Types of VA Systems

Based on whether the VA system is shared among a group of users or not, we divide the current commodity VA systems into two types: *Personal VA Systems* are designed only for personal use. They are usually integrated into users' mobile devices, such as smartphones (e.g., Google Now and Siri). Differently, *Family/community Shared VA Systems* are designed to be used in the home or office environments. They are usually built into a stand-alone device and shared by multiple users. The typical commodity products of this type are Amazon Alexa and Google Home. The differences between these two types of VA systems are that the personal VA system usually takes voice commands using the microphone of the user's mobile device, which is in the proximity to the user, while the family/community shared VA system is usually designed to pick up users' voice commands from a distance in a house or office. The family/community shared VA systems are considered to be of higher risk because adversaries could easily access the VA systems without being noticed. Thus, this type of VA system is the primary focus of this work.

4.3.3 At-risk Information/Operations in VA Systems

VA systems are usually linked to the users' personal information and even their family/community information. When an adversary has access to the VA system, he can easily get the user's private information. For example, the adversary can get the user's shopping information by asking "What is in my shopping list?". The adversary can also get the personal email content by saying “Read me my email”. Furthermore, the adversary can get the user’s family schedule via the voice command ”List all events for January 1st”. Moreover, recent VA systems are usually deployed as a hub connecting various smart appliances at home or in the office. In that case, the adversary can use voice commands to control the smart appliances without permission. For instance, an adversary can put the user in danger by saying “Unlock the exterior door”. Along with this direction, we investigate the privacy-sensitive voice commands, related to issues of private information leakage and unauthorized operations.

Limited Defense Methods in VA Systems. Most of the off-the-shelf VA systems require a pre-defined voice command (known as *wake word*) to wake up the system, such as “Alexa” and “OK Google”. These wake words could be used to verify the identity of the speaker by comparing with the pre-recorded sounds in the user profile, which is built when the user enrolls the system. However, such audio-based speaker verification in the current VA systems is not trustworthy, because the acoustic features they depend on can be easily spoofed. To warn the VA system users about this issue, Google Home particularly notes that *“A similar voice might be able to access this info, too”* [26]. Furthermore, current VA systems can only verify users based on the wake words, leaving the voice commands unprotected from the attacks. Thus an adversary only needs to focus on attacking the wake words, which makes the attack much easier. In addition, we find that the VA system stays in the listening mode for a long time (e.g., 30 seconds for Google Home) to capture voice commands after being woken up. During this time period, the VA system is defenseless to any adversary. Moreover, research shows that the audio-based VA systems are vulnerable to various audio attacks, including imitation and replay attacks.

4.3.4 Attack Model

We consider an adversary who is interested in obtaining the user’s private information or exerting an unpermitted operation from the standalone VA device at office or home, which may involve multiple users. We assume the adversary can not physically break the VA device, take control of the VA cloud service or get the possession of the user’s wearable device. We also assume that the VA user wears a wearable when using the VA system, which is normal given the huge number of wearable device users [109]. We summarize the potential attacks in two major categories:

Attack on User’s Absence. This type of attacks can only be launched without causing notice when the user is away from the VA device, and an adversary needs to get close to the

VA device:

- Random Attack. An adversary who does not know the user's voice characteristics can try to fool the VA system by using his own voice. Because the adversary only needs to attack the single wake word, such attack still has high success rates.

- Impersonation Attack. An experienced adversary who knows the user's voice characteristics can attack the VA system by imitating the user's voice. The adversary can also synthesize the user's voice by using an audio editing software and playback the synthesized sound via a loudspeaker to launch the attack.

- Replay Attack. An adversary who has the opportunity to observe the user's voice command can use a microphone to record the it and play it back via a loudspeaker to fool the VA system.

Co-location Attack. The type of attacks can still be launched surreptitiously even when the user is present near the VA device:

- Hidden voice attack. An adversary may embed the recorded user's voice commands into the background of music or video streams or directly generate hidden voice commands according to the knowledge of the underlying VA system [43]. The generated voice commands can be recognized by VA systems but not perceptible to human. Moreover, an adversary can control the volume or mute the VA device via hidden commands to avoid being noticed from the audible reply.

- Ultrasound Attack. An adversary may modulate the recorded user's voice commands onto the ultrasound frequency band (i.e., $\geq 20KHz$), and use such modulated sound to fool the VA system. Although human ears can not hear the modulated voice commands, they can still be recognized by existing VA systems due to the non-linearity of the microphone [134].

4.4 User Verification Design

4.4.1 Why Wearable? Why Motion Sensor?

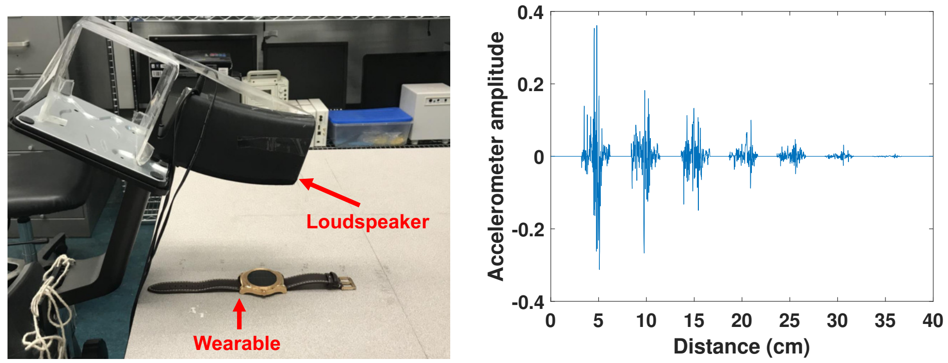
Why Wearable? While the number of wearable users has reached half billion worldwide [109], it is natural for us to leverage such pervasive wearable devices in our design, which could benefit a huge number of users without causing an additional cost. Moreover, the wearable's nature of being usually worn on the user body and rarely left unattended make it eligible for a trusted device. For example, people have been using the ID wristband for years [16], and the wearable devices have been considered a valid security token in various payment systems [88]. Many recent studies further develop continuous authentications on the wearable to guarantee the security of using the wearable as a trusted ID [116].

Why Motion Sensors? Why not a Second Microphone? There are three reasons why we choose motion sensors. First, a motion sensor provides a new way to examine acoustic signals in the vibration domain, which captures the unique frequency and amplitude responses caused by its hardware components. Compared to the approach using a second microphone to verify the voice command in the same acoustic domain, our design crossing two domains reveals more inherent signatures from the voice command. Thus, it can shield against more advanced acoustic attacks, such as the ultrasound and hidden command attacks [38], which cannot be prevented with a second microphone. Second, the low-frequency motion sensors require low energy and less computation, which is favorable to the resource constraint wearable device, while a microphone drains battery fast [135]. Third, most wearable devices are equipped with motion sensors [72, 40] because of their design purpose to track fitness/activities, but many wearables do not contain a microphone.

WearID versus Traditional Methods on Wearables. We compare WearID with three traditional wearable-based solutions, a virtual button [53] on the wearable device to access VA systems, proximity detection using Bluetooth/WiFi and liveness detection [45]. All the three approaches have no or limited capability to verify the voice source and thus suffer from various acoustic attacks such as hidden command attack and ultrasound attack. Furthermore, the liveness detection requires the user's mouth near the VA device and a virtual button to access the VA system. It requires the user to unlock the wearable, find the App (or button), press the button and wait. Differently, WearID verifies the voice commands across two domains and requires low user effort.

4.4.2 Acoustic Response in Vibration Domain

Distinct Acoustic Characteristics. The foundation of our cross-domain user authentication is using the low-cost accelerometers in wearable devices to capture human voices with distinct acoustic characteristics. Although the accelerometers in wearable devices are not designed for capturing sounds, the mechanism of capturing accelerations is based on the Micro Electro Mechanical System (MEMS) technology [113], which is the same technology that enables traditional microphones to capture sounds. More specific, the microphone uses the vibrations of its membrane to capture sounds [123, 59], while the accelerometer uses the subtle movements of



Experiment setup

Response distance test

Figure 4.2: Experiment setup for feasibility study and the accelerometer response distance test.

its inertial mass to capture accelerations. Therefore, when there is a sound source close to an accelerometer, the accelerometer can also capture the sounds by measuring the movements of its inertial mass resulted from the sound pressures. However, due to different MEMS implemented in the accelerometer and microphone, the accelerometer captures the unique amplitudes and frequencies of the sounds, which are distinct from those captured by the microphone.

To study the feasibility of using the accelerometers on wearable devices to capture distinct characteristics of sounds, we conduct an experiment using the setup illustrated in Figure 4.2 (a). Specifically, we play an audio signal that sweeps from 0Hz∼ 22kHz by using a Logitech loudspeaker and use the accelerometer on a wearable device (i.e., Huawei Watch 2) and a microphone to record the sound. As shown in Figure 4.4, we find that the accelerometer can capture the sound frequency between 400Hz and 3200Hz, whereas the microphone can capture the sound frequency between 80Hz and 15kHz. Although the accelerometer captures a shorter range of sound compared to the microphone, it is sufficient to cover the major human voice frequencies (i.e., 1000Hz∼ 4000Hz) [14]. Furthermore, we find that the responses of the accelerometer and microphone to the same frequencies are different in the amplitudes, indicating the accelerometer captures distinct characteristics of the sound compared to the microphone.

Aliased Signal. Moreover, we find that the audio signals captured by the accelerometer are aliased. The signal aliasing is a phenomenon when the frequency components shift to a new frequency point and overlap at that frequency [85]. Figure 4.5 compares the spectrograms (introduced in Section 4.5.1) of the microphone and the accelerometer under a single chirp sound, where the accelerometer's spectrogram shows a "Zigzag" curve. This indicates that the acoustic response of accelerometers at a single frequency could correspond to multiple

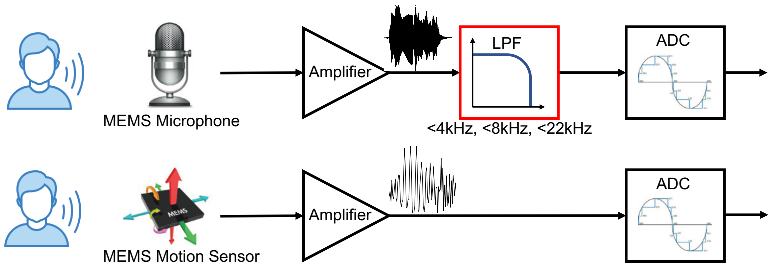
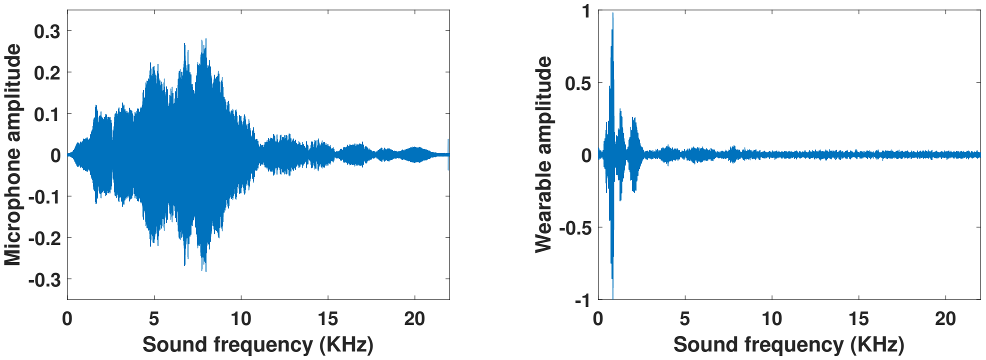


Figure 4.3: Hardware flow of microphone and motion sensor.



Amplitude of microphone

Amplitude of wearable accelerometer

Figure 4.4: Responses of the microphone and the accelerometer to a chirp from 0Hz to 22kHz (in time-domain amplitude).

frequencies of the sound. If these frequencies appear at the same time, their resulted responses would be overlapped (i.e., aliased). This is because accelerometers usually do not contain a low-pass filter between the amplifier and the analog-digital converter (shown in Figure 5.1) to set the frequency limit to its digitization, as the microphone does. Thus, the sounds when sampled by the low sampling rate accelerometer would generate aliased signals due to violating Nyquist Sampling Theorem [91]. Particularly, we model the relationship between the aliased accelerometer signal and the original audio signal as:

$$f_{alias} = |f - Nf_s|, N \in Z, \tag{4.1}$$

where f_{alias}, f and f_s denotes the aliasing signal frequency, original audio signal frequency and sampling rate of the accelerometer. We discuss more about this relationship based on a single tone signal in Appendix Section **??** and show an example in Figure **??**. Figure 4.5 (b) also suggests that when using the accelerometer to record the human voice, the accelerometer data is the combination of multiple aliasing signals of higher-frequency voice signals, which is very hard to interpret. Thus, one major task of this work is to explore the complex relationship between the microphone data and the accelerometer data to facilitate the cross-domain comparison.

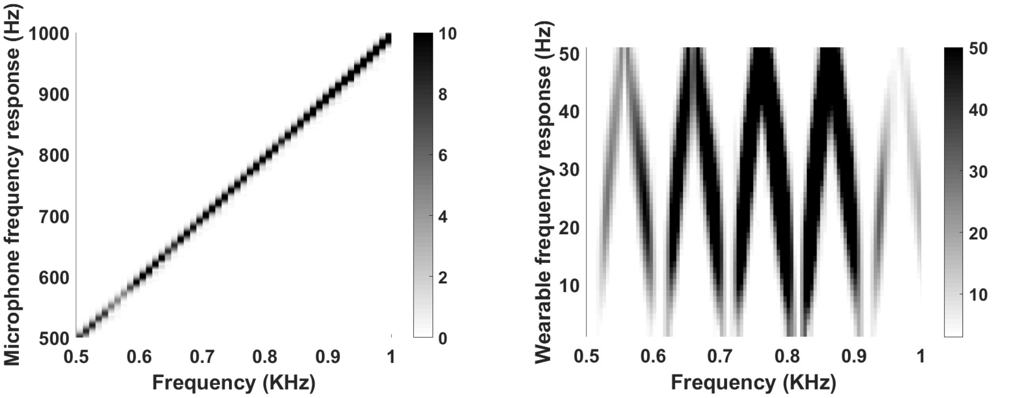
Short Response Distance. We also test the capability of the wearable's accelerometer on picking up voice under various distances. We play a recorded voice command (i.e., ”one”) with the fixed volume using a loudspeaker as shown in Figure 4.2 (a) and utilize the smartwatch LG Urbane W150 to record the sound under distances from $5cm$ to $35cm$. In Figure 4.2 (b), we can observe that the amplitude response of accelerometer decreases with the distance, and over the distance of $25cm$, the responses can be barely observed. Such short response distance of the accelerometer can further assist to shield against many long-distance acoustic attacks.

4.4.3 System Purpose and Challenges

Our system checks whether the legitimate user is the voice source when his/her voice commands are received by the VA system. The basic idea is to compare the command sound across the audio domain (i.e., via the VA device's microphone) and the vibration domain (i.e., via the motion sensor of the user's wearable). If the command sound matches across the two domains, the voice command is verified to come from the legitimate user (i.e., owner of the wearable device). Existing VA systems only focus on verifying wake words and neglect the protection of more sensitive voice commands, which opens more opportunities to adversaries. In comparison, our system requires confirming whether the voice commands come from the right user. There are many challenges to design such a system: 1) It is challenging to match the command sound from the sensors working in two different domains, which exhibit distinct response characteristics and have a considerable gap in sampling rates (e.g., 8000Hz versus 200Hz). When re-sampling to fill such gap, a slight noise in the data can be amplified to greatly impact the comparison results. More discussion about the difficulties of such cross-domain comparison are presented in Appendix **??**; 2) It is unknown whether the motion sensors on wearables provide sufficient information to characterize human voice sounds, given their low fidelity and design purpose of capturing motion instead of sound; 3) How to trigger and synchronize the authentication process on the VA device and the wearable device need to be explored; 4) The proposed authentication system should defend against various attacks regardless the user's presence or absence to the VA device.

4.4.4 System Flow

Toward this end, we develop a novel VA system, WearID, which verifies the authenticity of voice commands through capturing the user's voice from the vibration and audio domains. Figure 5.2 illustrates the flow of WearID. Our system exploits the *Audio Domain Data Collection* and



VA's microphone

Accelerometer on Huawei watch 2 sport

Figure 4.5: Spectrogram of the frequency responses at the microphone and the accelerometer under a chirp signal ($500Hz \sim 1000Hz$).

the *Vibration Domain Data Collection* to collect the microphone data from VA device and accelerometer data from the wearable, which describe the user's voice interaction with the VA system in two different domains. *Coarse-grained Synchronization* aims at triggering the motion sensors to record the voice commands at the right time (i.e., after the wake word) and provides coarse synchronization between the two domain data. The insight is that the wake word for initiating the VA device could be utilized to trigger the data collection on both devices. We propose two alternative approaches to achieve the coarse-grained synchronization. The *WiFi Communication-based approach* only requires the VA device to detect the wake word and trigger the wearable to start data collection through the WiFi communication when both devices are in the same WiFi network [93]. We note that emerging wearables are in a trend of having standalone WiFi modules that can connect to WiFi networks directly. In the case of wearables not having WiFi modules, they still can connect to WiFi networks through the paired smartphones. The alternative approach *Parallel Wake-word Detection-based approach* uses the motion sensor in the wearable device to detect the wake word independently based on voice recognition in vibration domain. To achieve this, WearID reuses the motion sensor data from the ongoing fitness tracking App, which continuously counts the user's walking steps. After the synchronization, the wearable and VA device start to collect the accelerometer and microphone readings respectively for the voice command. The data will be uploaded to a cloud server that is running our processing algorithm to compare the cross-domain data for user authentication.

WearID exploits *Vibration Domain Feature Derivation* and *Audio Domain Feature Derivation* to derive reliable time-frequency features from the data collected by the wearable's motion sensor and the VA device's microphone, respectively. The derived features are converted to

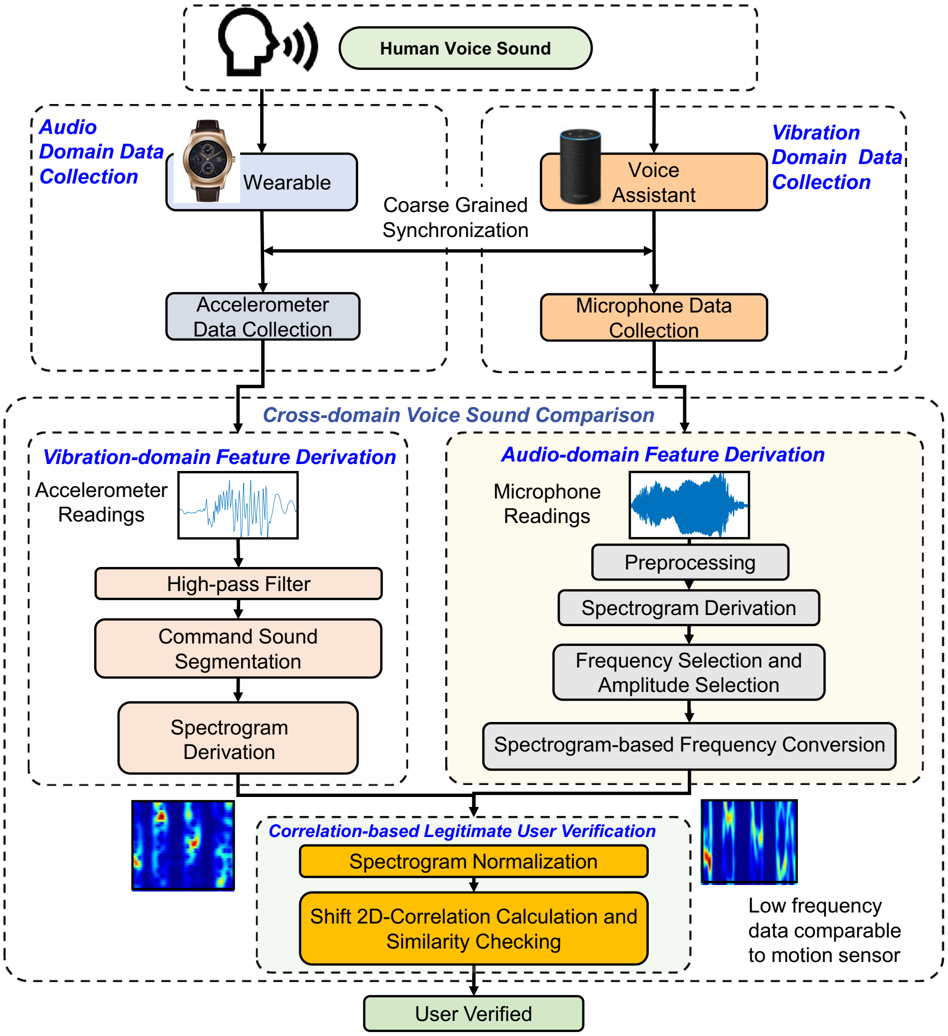
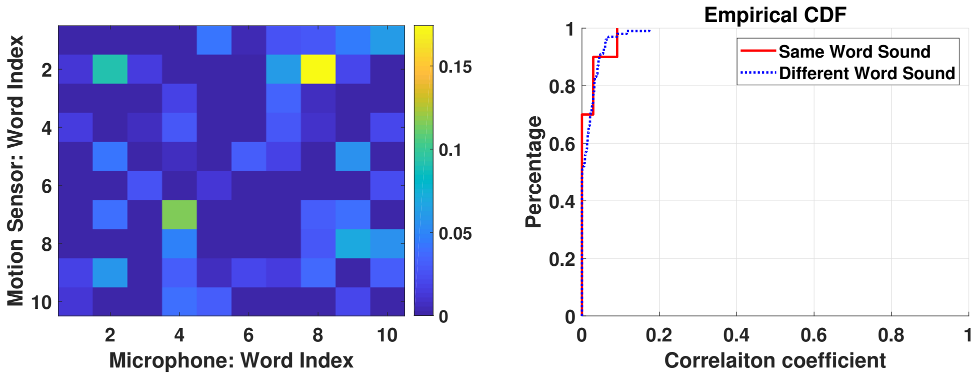

Human Voice Sound
Audio Domain Data Collection
Wearable
Vibration Domain Data Collection
Voice Assistant
Coarse Grained Synchronization
Accelerometer Data Collection
Microphone Data Collection
Cross-domain Voice Sound Comparison
Vibration-domain Feature Derivation
Accelerometer Readings
High-pass Filter
Command Sound Segmentation
Spectrogram Derivation
Audio-domain Feature Derivation
Microphone Readings
Preprocessing
Spectrogram Derivation
Frequency Selection and Amplitude Selection
Spectrogram-based Frequency Conversion
Correlation-based Legitimate User Verification
Spectrogram Normalization
Shift 2D-Correlation Calculation and Similarity Checking
Low frequency data comparable to motion sensor
User Verified

Figure 4.6: User verification overview.

comparable spectrograms based on a complicated unique relationship between the audio and vibration domains. The *Correlation-based Legitimate User Verification* calculates the similarity between the derived spectrograms for user verification.

In particular, the *Vibration Domain Feature Derivation* removes the mechanical noise (e.g., due to hand movements) from the motion sensor data by using a high-pass filter and extracts the voice command segmented from the motion sensor readings by examining the moving variances. The two-dimension time-frequency description of signal, spectrogram, is then derived from the identified motion sensor segment. Similarly, the *Audio Domain Feature Derivation* pre-process the microphone data to remove the acoustic noise and identify the command sound segment, which is utilized to derive the spectrogram. The next is to convert the microphone spectrogram to the low-frequency form comparable to the motion sensor and maximize the



(a) Correlation matrix

(b) CDF of the correlation

Figure 4.7: The time-domain correlation between the microphone data and motion sensor, which are resampled to the same sampling rate level (Illustrated with 10 words on Amazon Echo and Huawei watch 2).

intersection of the two distinct sensing modalities' acoustic responses. This is done by the *Spectrogram-based Frequency Conversion* and the *Frequency Selection and Amplitude Selection* , while the later guides the conversion according to the identified unique acoustic characteristics of motion sensor. The *Correlation-based Legitimate User Verification* first performs the *Spectrogram Normalization* to normalize the time lengths and magnitudes of the spectrogram in two domains. *Shift 2D Correlation-based Similarity Calculate* computes the 2D-correlation between the spectrograms to check the similarity and shift one spectrogram over time during calculation to address the synchronization errors. The resulted maximum 2D-correlation coefficient is compared with a threshold to determine whether the voice command received by the VA device is from the legitimate user (i.e., wearable owner).

4.5 Prevent Privacy Leakage from Voice Assistant Attacks

Different from the microphone, the accelerometer shows its unique characteristics when responding to sounds. In order to match the voice commands captured by the two different sensing modalities to verify the user, our basic idea is to convert the high-frequency microphone data into the low-frequency data that describes the ”equivalent” acoustic responses to the accelerometer. In this section, we introduce our approaches to exploit the complex relationship between the two domains to verify the voice command.

4.5.1 Cross-domain Voice Command Comparison

Difficulty of Comparing Microphone Data with Motion Sensor Data. Figure 4.7 illustrates the difficulty of comparing the microphone data with the motion sensor data, where a participant speaks ten words to both a microphone and a accelerometer, and both data are re-sampled to the same sampling rate for similarity comparison. Particularly, Figure 4.7 (a) shows the time-domain correlation coefficient between the microphone recorded sound (i.e., X axis) and motion sensor data (i.e., Y axis) by cross-comparing ten words. We observe that the correlations at the diagonal (i.e., same word sound) and non-diagonal (i.e., different word sounds) are indistinguishable. The results indicate that the re-sampling technique and the time-domain analysis are insufficient to address the similarity comparison of the two different sensing modalities. Figure 4.7(b), CDF of the correlation coefficients, further depicts the challenge of matching the sound across the two domains, where the sound of the same word and those of different words all show low correlation values (i.e., less than 0.1). Thus, we need to investigate the inherent unique relationship between the two sensing modalities to facilitate their similarity comparison.

Spectrogram Derivation. As the time-domain analysis is shown to be limited to describe the complex relationship between the high frequency microphone and the low-frequency motion sensor data (Appendix **??**), we resort to the time-frequency analysis and derive the spectrogram (i.e., two-dimensional representation of the signal) to analyze their unique responses to sound. Particularly, we compute the Discrete Time Short Time Fourier Transform (DT-STFT) of the microphone/accelerometer readings $x(n)$ using a sliding window function as expressed in equation 4.2.

$$DTSTFT(m,\omega) = \sum_{n=m}^{m+N-1} x(n)w(n-m)e^{-j\omega n}, \tag{4.2}$$

where m and ω are the time index and frequency index of the two dimension signal description, $w(n)$ is a window function, and N is the DT-STFT size of the data in the sliding window (e.g., 2048 for microphone data and 64 for accelerometer readings). We then compute the magnitude squared of the DT-STFT $P(m,\omega) = |DTSTFT(m,\omega)|^2$, which is the power spectrum at time m. Next, we slide the window by step of size p and obtain the spectrogram, the time series of the squared DT-STFT $S = [P(0,\omega), ..., P(\frac{M-N}{p}, \omega)]$.

Spectrogram-based Frequency Conversion. To convert the spectrogram of high-frequency

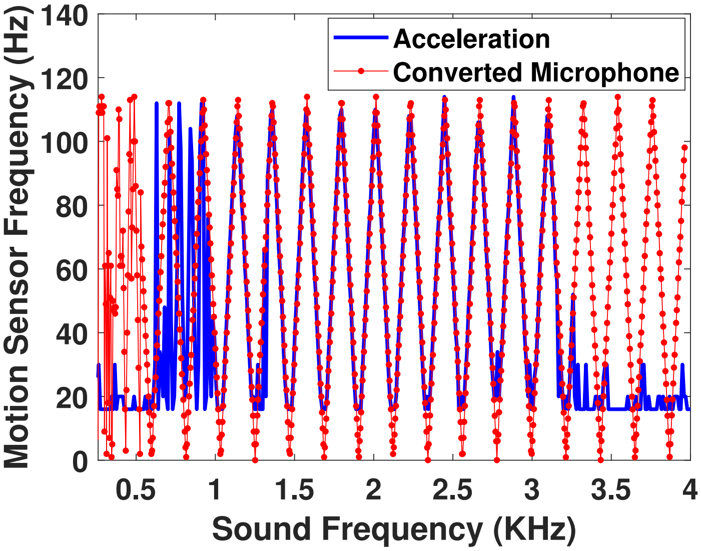
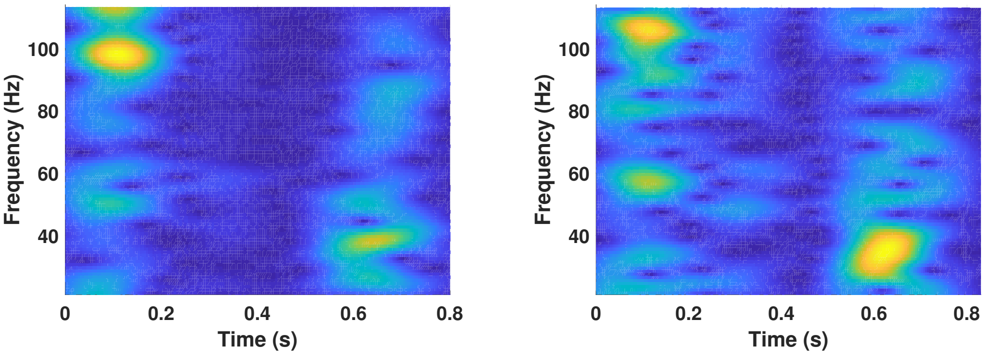


Figure 4.8: Converting the microphone data of a frequency chirp ($0 \sim 4KHz$) into the low frequency data based on spectrogram.

microphone data to the low-frequency one that is comparable to the accelerometer spectrogram, we develop a spectrogram-based frequency conversion method. The high-to-low frequency conversion takes as input the microphone spectrogram point $P_{mic}(t_n, \omega_m)$ and calculates its new position (t_n, ω_w) in the converted low-frequency spectrogram. The original microphone frequency point ω_m is then mapped to the low-frequency point ω_w based on Equation 4.5, while the time point is unchanged. The resulted new spectrogram is computed as $\hat{P}_{mic}(t_n, \omega_w) = \sum_{n=-\inf}^{\inf} P_{mic}(t_n, win(|\omega_m + n \times \omega_{ws}|))$, where $win()$ is a window function with non-zero value for $[0, \omega_{ws}]$ and ω_{ws} is the sampling frequency of accelerometer. This new spectrogram reflects the acoustic responses of accelerometer under the same voice command in the microphone data. As shown in Figure 4.5(b), we can generate the similar "accelerometer" spectroram given a single frequency chirp in Figure 4.5(a). Due to the complexity of human voice, we study the unique characteristics of accelerometer for generating more "precise" low-frequency spectrogram.

Frequency and Amplitude Selection. Except the low sampling rate, the accelerometer's sensor structure, the vibrations of other electric components in the wearable device and the low sensor fidelity all cause the accelerometer to respond to sounds differently from a microphone. In particular, these factors cause the accelerometer to subdue some frequencies while in favor of responding to other frequencies with higher amplitudes. Moreover, being not dedicated for recording sounds, the accelerometer shows lower sensitivity to sounds and some small volume



(a) Motion sensor data

(b) Converted Microphone data

Figure 4.9: Comparison of the accelerometer spectrogram with the converted microphone spectrogram under ”Alexa”).

sounds may not result in readings in the accelerometer but can be easily recorded by microphones. To reveal the complex relationship between the two sensing modalities when recording sounds, we compare the accelerometer's spectrogram with the converted low-frequency microphone spectrogram for every frequency point of a chirp signal. In particular, we extract the maximum amplitude on each spectrum (i.e., column of the spectrum) at every time index and obtain a clear *frequency sweeping curve* for the two sensing modalities as shown in Figure 4.8. We observe that the accelerometer only responds to a small frequency range (700Hz - 3300Hz) clearly (i.e., blue zigzag curve). Moreover, only for this frequency range, the two sensing modalities' acoustic responses match well. Besides frequency characteristics, we also analyze the amplitude of the sound that could generate a response on the wearable's accelerometer. In particular, we find that when the sound is greater than 70dB, the sound can leave obvious readings on the wearable, which is consistent with the observations in Accelword [135]. Therefore, in order to facilitate the similarity comparison between the two sensing modalities, wearID converts the microphone data to the low frequency data as well as following the frequency and amplitude characteristics of the accelerometer, which maximize their acoustic response intersections. Figure 4.9(b) illustrates an example of the converted microphone spectrogram for the word ”Alexa”, which shows an ”equivalent” low frequency form as that of the accelerometer in Figure 4.9 (a).

Spectrogram-based Conversion Algorithm. We develop the spectrogram-based conversion algorithm, which integrates the spectrogram-based frequency conversion with the frequency and amplitude selection. The algorithm details are introduced in Appendix Algorithm 1, the transformation algorithm takes microphone spectrogram S_{mic} and the sampling rate of the

Algorithm 1 Spectrogram-based Conversion Algorithm

```
function CONVERSION($S_{mic}$)
    Input: $S_{mic}$-original microphone spectrogram
           $f_{ws}$-sampling rate of accelerometer
    Output: $\hat{S_{mic}}$-converted microphone spectrogram
    $|\hat{S_{mic}} = zeros(T, F)|, \omega_{ws} = 2\pi \times f_{ws}$
    for $t = 1 : T$ do
        for $f_mic = 700 : 3300$ do
            // Frequency selection
            for $N_{shift} = -10 : 10$ do
                $f_w = |f_{mic} - N_{shift} \times f_{ws}|$
                if $|S_{mic}(t_n, f_m)| > 70dB$ & $f_w \leq f_s$ & $f_w > 0$ then
                    // Amplitude selection
                    $\hat{S_{mic}}(t_n, f_w) = \hat{S_{mic}}(t_n, f_w) + |S_{mic}(t_n, f_m)|$
                    // Spectrogram-based frequency conversion
                end if
            end for
        end for
    end for
end function
```

accelerometer f_{ws} as input and calculates the new spectrogram $\hat{S_{mic}}$ that locates within the low-frequency range (e.g., $0 - 200Hz$) as the output. Specifically, the algorithm only selects the power spectrum point within frequency $700Hz$ to $3300Hz$ and with the magnitudes greater than 70dB for conversion. Spectrogram-based frequency conversion is then performed to the selected spectrogram points based on equation 4.1. If multiple spectrogram points are mapped to the same point in the new spectrogram, their magnitudes are added together.

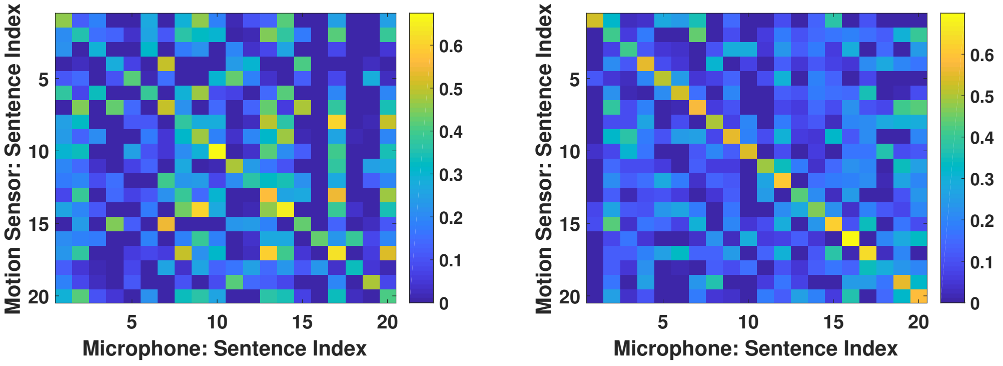
4.5.2 Legitimate User Verification

Spectrogram Normalization. The scales of amplitude are greatly different in accelerometer and microphone readings. Therefore, we develop the 2D-interpolation scheme and the 2D-normalization scheme to normalize the length and magnitude of the two spectrograms. In particular, the 2D-interpolation scheme performs row-based interpolation to align the two spectrograms. The 2D-normalization resolves the scale differences of the two spectrograms and conduct column-based normalization using Equation 4.3:

$$S_{norm}(t_n, w_m) = \frac{S(t_n, w_m) - S_{min}(t_n)}{S_{max}(t_n) - S_{min}(t_n)}, \tag{4.3}$$

where $S(t_n, w_m)$ is a power spectrum point at time t_n.

Cross-domain Comparison based on Shift 2D-Correlation. Next, we match the voice commands across the audio domain and the vibration domain and calculate the 2D-correlation coefficient between the microphone and accelerometer spectrograms using equation: $Corr(S_{mic}, S_{acc}) = \frac{A \times B}{\sqrt{A^2 \times B^2}}$, where A, B represent two spectrogram matrices. Note that the microphone data and the accelerometer data are coarsely synchronized in Section 4.5.3. To further reduce the synchronization error and improve the similarity comparison accuracy, we



(a) Voice sound of words

(b) Voice sound of sentences

Figure 4.10: The spectrogram correlation based on our method.

perform the *Shift 2D-Correlation* during the correlation calculation between the two domain spectrograms. In particular, we fix the microphone spectrogram and shift the spectrogram of accelerometer one index by one index along time axis to calculate the similarity. More specifically, we use a sliding window with a fixed size and shift it to left or right on the accelerometer's spectrogram within time T (e.g., $500ms$). The 2D-correlation is calculated for each shift and the maximum 2D-correlation coefficient is found as the similarity score of the two domain information. A threshold-based method is then applied to examine the similarity score and verify the user. Figure 4.10(a) illustrates the similarity comparison result of the above method to differentiate 20 different words. Clearly, the diagonal comparisons (i.e., same words) show much higher correlation coefficients. Figure 4.10(b) further confirms the efficiency of our method to differentiate a user's 20 voice commands (i.e., sentences), which are distinguished better, because sentences contains much more voice information than single words.

4.5.3 Data Preprocessing

Coarse-grained Synchronization

Coarse-grained synchronization triggers the VA device and wearable to collect the voice command and coarsely synchronize the two devices. The existing VA system requires the user to speak a wake word such as "OK Google" and "Alexa" to wake up the VA device before taking any voice commands. WearID integrates such method to trigger the verification process and start the data collection on both devices. In particular, we develop two alternative approaches, the WiFi communication-based method and the parallel wake-word detection method. 1) WiFi communication-based method leverages the existing setting where the wearable device and the

VA device are connected to the same WiFi network [93]. The waked VA device sends a message through the connected WiFi network to the wearable device to trigger its data collection. While standalone wearables directly receive WiFi packets, the wearables that work with a paired smartphone can receive the message relayed by the smartphone's Bluetooth. In addition, the time lag between the microphone and accelerometer data is usually less than 40ms, which is mainly caused by the network delay and the system time differences. Both devices' readings under the WiFi communication-based method are shown in Appendix Figure 4.11(a) and (b), which are coarsely synchronized. 2) As an alternative method, the parallel wake-word detection method requires the wearable to detect the wake word in parallel with the VA device. The wearable reuses the accelerometer data from an ongoing fitness APP to recognize the wake word. Our study shows that a wake word can be recognized based on accelerometer from 10 words with 83% accuracy by Random Forest. The wake-word detection on the wearable can be further improved if assisted with a hand motion detection scheme.

Noise Removal

We note that the accelerometer data on the wearable contains the hand movement noises as shown in Figure 4.11(b). While these mechanical noises are in low-frequency compared to acoustic signals, we apply a high-pass filter (e.g., cutoff frequency $30Hz$) to remove the noise and obtain the accelerometer data that describes the voice command more precisely as shown in Figure 4.11(c). Besides, to obtain the microphone data that precisely captures voice command and remove the acoustic noises, we apply a bandpass filter (e.g., $300-4000Hz$) to filter out the acoustic sounds beyond the human voice frequency range.

Figure 4.11(a) and (b) show the accelerometer readings of the WiFi communication-based method, where the accelerometer starts recording after receiving the waked VA device's message. We can find that the data on the microphone and the accelerometer are roughly synchronized. In addition, WearID exploits a high-pass filter to reduce the noise introduced by hand movements or other mechanical vibrations. The resulted accelerometer data (Figure 4.11(c)) shows a slightly similar shape to the microphone data (Figure 4.11(a)).

Voice Command Segmentation

We next search for the *starting point* and *ending point* of the voice command on both the microphone and the accelerometer data to identify the voice command segments respectively. In particular, we analyze the moving variance of the data amplitudes and extract the envelope that covers the voice command. We then apply a threshold-based method to search for the

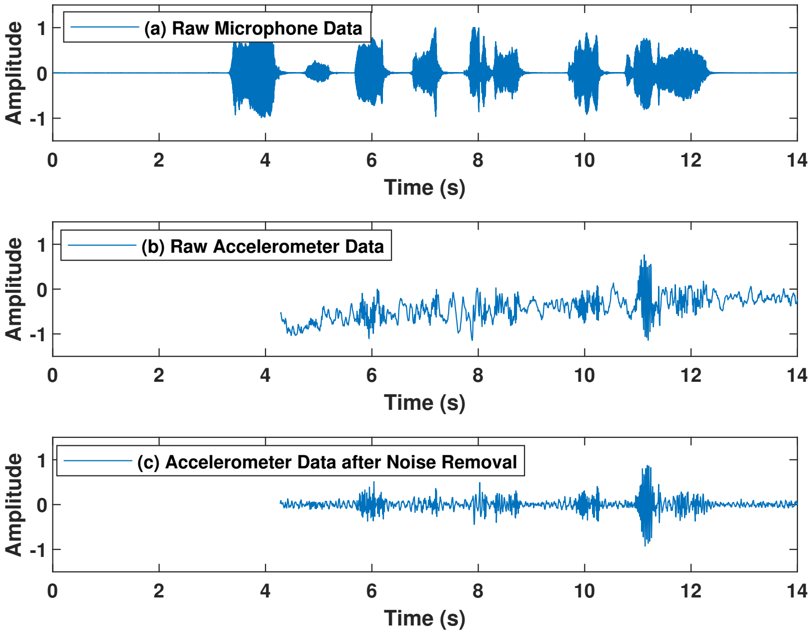


Figure 4.11: Synchronization of the microphone data (8000Hz) and accelerometer data (200Hz) and the hand vibration noise removal from accelerometer data.

starting/ending points. The voice command segmentation on the microphone data is easy and accurate. But on the accelerometer, the segmentation errors may be large due to its low sensitivity to sound and low sampling rate. To address this issue, we apply the microphone segmentation results to assist the voice command segmentation on the accelerometer. Because both data are coarsely synchronized, we search for the starting point on accelerometer within a window W_T after the starting point of the microphone segment. We then calculate the ending point on accelerometer data based on the microphone segment length.

4.6 Performance Evaluation

4.6.1 Experimental Methodology

Device. To evaluate WearID, two smartwatch models, Huawei 2 sport (100Hz) and LG W150 (200Hz) are involved to collect accelerometer readings. The accelerometer specifications of the two wearable devices are listed in Table 4.1. The two smartwatches run Android Wear OS 2.0 with Bluetooth LE. Although the sampling rate of the accelerometers could reach 4000Hz, the vendors constrain the sampling frequencies to be under 200Hz to ensure low power consumption. We record the voice commands leveraging a typical VA device, Google Home, which supports

Table 4.1: The specifications of the accelerometers in the tested wearable devices.

Model	**Accelerometer**	**User Programmable Range**	**Sensor sampling rate**	**System sampling rate**
LG Urbane watch 150	Invensense M6515	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$	4-4000Hz	200Hz
Huawei watch 2 sport	STMicroelectronics LSM6DS3	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$	4-1600Hz	100Hz

$8 \sim 96kHz$ audio recording. We use a Logitech S120 speaker [79] to conduct replay attacks and hidden voice commands. To perform ultrasound attacks, we use a function generator (i.e., Keysight Technologies 33509B [111]) and a tweeter speaker [56] which could generate sounds with frequency from $2kHz$ to $25kHz$.

Experimental Setup. We evaluate the performance of WearID in a typical office environment, where regular ambient noises (e.g., air condition, people walking) are presented. The participant speaks voice commands to a VA system placed 1 meter way while wearing the wearable device. Because the direction of the sound wave hitting the wearable affects the vibration energy received by the accelerometer, we test two typical directions with the wearable device being held horizontally or vertically to the user's mouth, which covers the best and worst cases. Note that the users can hold the wearable in any way to use WearID. To imitate the hidden voice commands and the ultrasound attack, we use the experimental setup as shown in Figure 4.2. We examine an extreme case where the loud speaker and the tweeter speaker are placed at $25cm$ distance to the wearable, which is hard to achieve in practical attacking scenario.

Data Collection. We involve 10 participants to test WearID under the normal situation and various attacks over a six-month period. The participants are asked to speak 20 representative voice command sentences as listed in Table 4.2 while wearing different wearables. From each participant, 80 voice command sound samples are collected. Besides, 100 hidden voice commands of 10 types and are utilized to evaluate WearID against hidden voice command attack [21], and a frequency sweeping signal from $15kHz \sim 25kHz$ is used to evaluate WearID against ultrasound attack. In total, 1000 data samples are collected from microphone and motion sensor respectively.

Evaluation Metrics. We define the following five evaluation metrics: true positive rate (TPR) is the percentage of legitimate voice commands being correctly verified; false positive rate (FPR) is the percentage of the adversaries' voice commands that pass the verification system; receiver operating characteristics (ROC) curve is generated by plotting the TPR against the FPR under thresholds from 0 to 1 with a step of 0.01; False Negative Rate (FNR) equaling to $1 - TPR$ is the percentage of legitimate users' commands being incorrectly rejected.

Table 4.2: Example of privacy leakages from voice assistant systems.

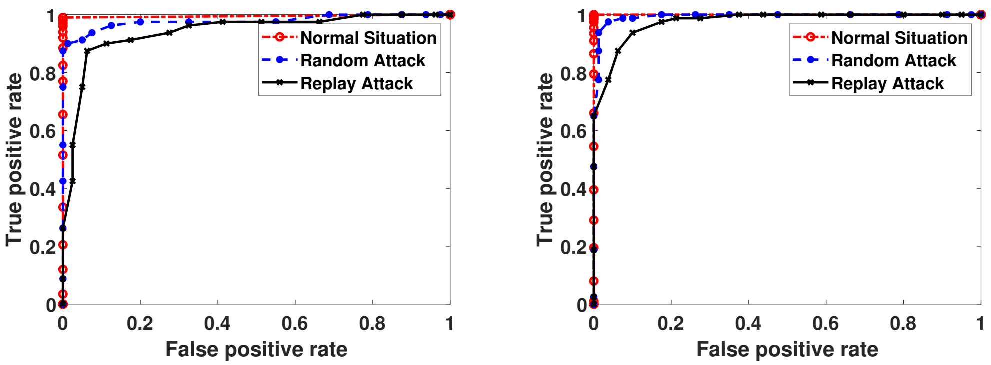
Security issues	Category	Voice Command Examples	Words
Privacy leakage	**Event schedule**	”What’s on my calendar for tomorrow”	6
		”Where is my next appointment”	5
		”List all events for January 1st”	6
		”How much is a round-trip flight to New York”	9
	Reminder	”Remember that my password is ’money’”	6
		”What is my password”	4
		”Add ’go to the grocery store’ to my to-do list”	10
	Shopping account information	”What’s on my shopping list”	5
		”Track my order”	3
	Contact	”Read me my email”	4
		”Call my mother”	3
Unauthorized operation	**Neighborhood location**	”Find me a Italian near my home”	7
		”What is the traffic to my home”	7
	Unauthorized purchase	”Add paper towels to my cart”	6
		”Order all items in my cart”	6
	Voice assistant	”Answer the call”	3
		”Delete all my reminders”	4
		”Play my favorite music on Spotify”	6
	Access smart home devices	”Show the living room camera”	5
		”Clear all Bluetooth devices”	4

4.6.2 Normal Situation

We first evaluate WearID in the normal situation when the user accesses the VA device while wearing a wearable and the attacker does not present. The VA device records the user’s voice command sound and the wearable device records the same sound simultaneously if it is worn by the user. But if the wearable is not worn by the user, it only records the environmental noises (e.g., acoustic noises and mechanical noises), because the wearable is not presented. The red curves in Figure 4.12(a) and (b) present the ROC curve of WearID to verify the legitimate users using Huawei watch 2 under the normal situation when the user uses the wearable in two typical ways. In particular, WearID achieves 99.8% TPR and 0% FPR to recognize the legitimate users with Huawei watch 2 for both holding ways. The red curves in Figure 4.13(a) and (b) shows the ROC obtained by LG W150 smartwatch. We find that WearID recognizes 99.6% legitimate users’ commands for both holding ways while FPR is 0%. The false negative rate in the normal situation for both smartwatches is 0.02% $\sim$ 0.04%, which indicates that WearID is robust and accurate to support the users’ daily usage of the VA device.

4.6.3 Attack on User’s Absence

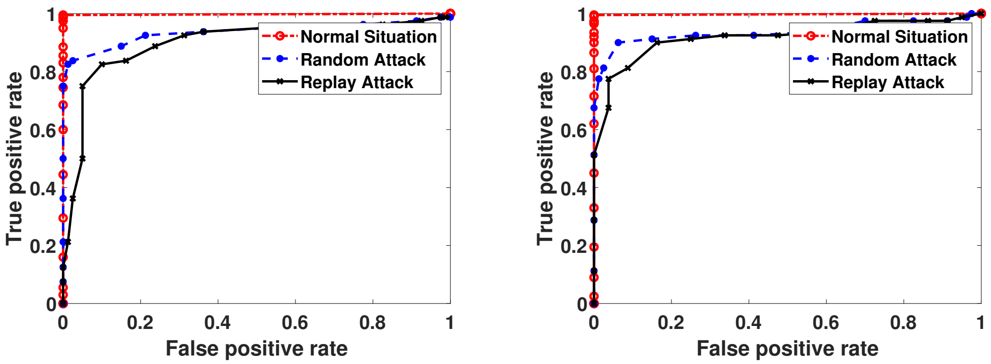
When the user is not present to the VA system, an adversary can reach to the VA device and perform a random attack or more sophisticated impersonate/replay attacks. During such attacks,



(a) Horizontal

(b) Vertical

Figure 4.12: Average ROC curve of verifying the user using Huawei watch 2 under normal situation, random attack and impersonate/replay attacks.



(a) Horizontal

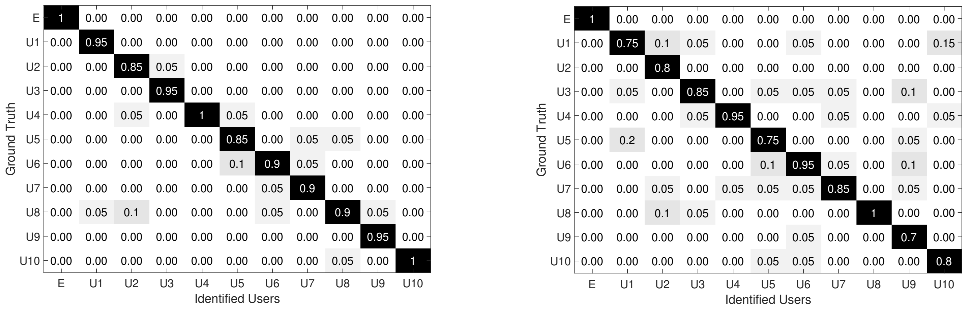
(b) Vertical

Figure 4.13: Average ROC curve of verifying the user using LG Urban W150 under normal situation, random attack and impersonate/replay attacks.

the VA device's microphone picks up the attacking sound and the legitimate user's wearable, while in a different place with the user, may record the owner's sound and the environmental noises.

Differentiating People's Voices.

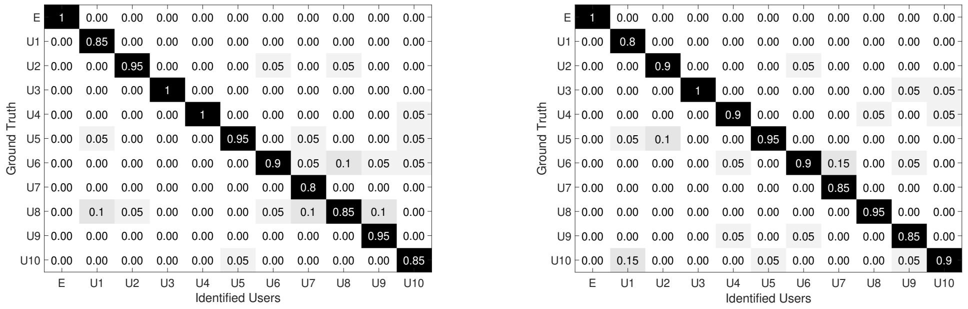
We first evaluate WearID's capability to differentiate people's command sounds across two domains, because when the user is absent with his/her wearable, the voice sounds received by the VA device may be different from that on the wearable. We consider an extreme case where we ask participants to speak the same voice commands to either the VA device or the wearable and evaluate WearID's capability to differentiate people's voices when fixing the speech



(a) Huawei watch 2

(b) LG Urban W150

Figure 4.14: Confusion matrix for distinguishing people's voices based on the same commands with horizontal holding way.



(a) Huawei watch 2

(b) LG Urban W150

Figure 4.15: Confusion matrix for distinguishing people's voices based on the same commands with vertical holding way.

content. In particular, each participant's accelerometer data is compared with other participants' microphone data to calculate speech similarity. Figure 4.14 shows the confusion matrix to differentiate people's voices between the VA device's microphone and the two smartwatches when they are held horizontally. We observe that WearID can accurately detect voice sounds received by the microphone and accelerometer to be from the same or different people. In particular, Huawei Watch 2 shows an average of 96% and the LG Urban W150 achieves 86% accuracy. Figure 4.15 further confirms our observation by showing the results of differentiating people's voice across two domains when the two smartwatches are held vertically. Specifically, Huawei Watch 2 obtain 91% average accuracy while LG Urban W150 achieves 94% accuracy. The results indicate that even under such extreme cases when people speak the same command, WearID can distinguish them correctly.

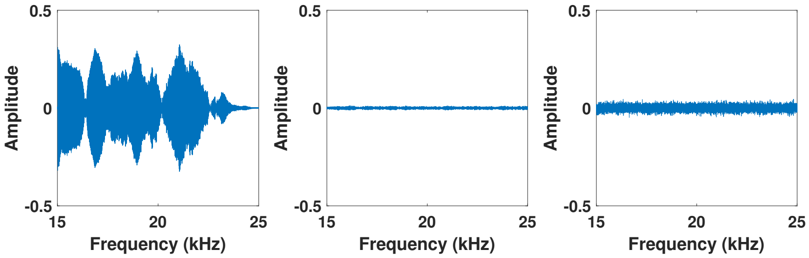


Figure 4.16: The frequency responses of the VA system and the wearables (i.e., microphone, Huawei watch 2, LG Urban W150 from left to right) under ultrasound attacks.

Against Random Attack

Under the random attack, an adversary does not have prior information about the user's voice sound and try to use his/her own voice to bypass the VA system. Since the user is absent from the VA system, the voice sound received by the user's wearable (e.g., the user's voice sound) would be different from that recorded by the VA system. To evaluate the performance of WearID under random voice attacks, we let each participant alternatively performs as the legitimate user and be attacked by other participants. The wearable records the user's voice sound and the VA system's microphone records the adversaries' sound. Figure 4.12 and Figure 4.13 show the average ROC curves of WearID (e.g., blue) to verify the users with two different wearables under the two typical holding ways. We observe that WearID can verify the user and reject random attacks with high accuracy. In particular, the AUCs for Huawei watch 2 and LG Urbane W150 are 94.46% and 88.85% under the horizontal holding way. In addition, the vertical holding way shows slightly higher AUC, which are 96.81% and 91.34% for Huawei watch 2 and LG Urbane W150 respectively. Moreover, given a FPR of 5%, WearID can achieve high TPRs of 95.21% and 98.47% for Huawei watch 2 held in horizontal and vertical directions respectively. The results indicate that WearID is effective to protect the VA system and verify the users with high accuracy under random attack. Moreover, in the practical scenarios, a legitimate user does not always speak and the wearable device usually records environmental noises. Thus the performance under random attack would approach to the normal situation (i.e., red curves).

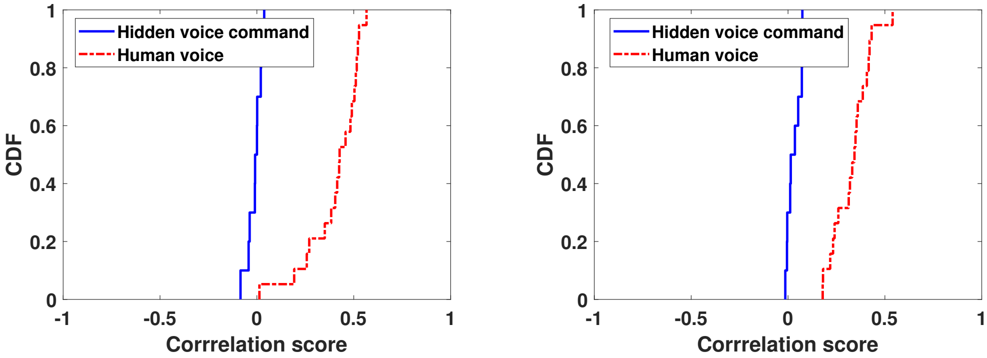
Against Impersonation and Replay/Synthesis Attack

We now consider the more sophisticated attacks on user's absence, which imitate/synthesize or just replay the legitimate user's voice commands to break the VA system. Ideally, an adversary could generate the voice sound which is exactly the same as the legitimate user. But the wearable is associated with the absent user and out of the adversary's control and it seldom happens when the two non-collocated devices (i.e., VA device and the wearable) receive the exactly same voice sounds from two independent sources. Because the wearable may still be possible to record the user's voice but with other speech content, we evaluate WearID in a more challenging scenario where the legitimate user's voice sounds are directly used for the attacking sounds of impersonation and replay attacks.

Figure 4.12 and Figure 4.13 show the average ROC curve (i.e., black curves) when verifying the user under impersonation/replay attacks. We find that WearID successfully verify the user by using both Huawei watch 2 and LG Urbane W150 under both horizontal and vertical holding ways. In particular, WearID achieves 89.12% and 86.78% for Huawei Watch 2 and LG Urbane W150 under horizontal holding way. The AUCs are 91.23% and 88.34% under vertical holding way. For a FPR of 10%, WearID can obtain the TPRs of 91.25% and 93.29% when Huawei watch 2 is held in horizontal and vertical directions. We find the performance of WearID under impersonation and replay attacks are slightly lower than those obtained under random attacks. This is because the adversary has obtained the additional knowledge about the user's voice to improve the attack. But WearID still effective on protecting the user's privacy. Moreover, in the practical scenarios, a legitimate user's wearable device does not always records the user's voice sounds, which make the performance approaching to that under normal situation.

4.6.4 User Verification under Co-location Attack

Against Hidden Voice Command. Under hidden voice attack, an adversary hides the recorded user voice sound into the noise sound, which is unintelligible to human but can be interpreted as commands by the VA system devices [43]. The adversary then plays back such noisy sound using a loudspeaker to control the VA system without causing the user's notice. In such scenario, both the VA system's microphone and the user's wearable receive the hidden commands. Figure 4.17 depicts the CDFs of the 2D-correlations between the microphone data and motion sensor data under hidden voice commands, where the loudspeaker is placed 25cm away to the two wearables and the volume is set to the maximum. We observe that the 2D-correlations between microphone and motion sensor are low for the hidden voice commands, which can be differentiated well from the legitimate user's voice commands. In particular, the median of the 2D-correlation coefficients for the hidden voice commands is around 0 for Huawei watch 2 and 0.05 for LG Urban W150. In comparison, the median 2D-correlation coefficients for the legitimate user's voice commands are around 0.5 for Huawei watch 2 and 0.4 for LG Urban W150. The reason is that motion sensors on the wearable has short response distance and unique response characteristics to sound. An adversary is hard to fool the system which is



(a) Huawei watch 2 (b) LG Urban W150

Figure 4.17: CDF of the cross-domain 2D correlations to distinguish the hidden voice commands and the legitimate user's voice commands.

developed based on verifying the sound from two domain information. The hidden voice attacks can thus be defended.

Against Ultrasound Attack. Under the ultrasound attack, an adversary modulates the recorded user voice command to an inaudible frequency and plays back it using an ultrasound speaker. Such inaudible sounds can be recognized by the VA system but they are hardly heard by the user [134]. In this scenario, both the VA's microphone and the user's wearable device is exposed to this inaudible sound. We thus evaluate WearID to see whether the ultrasound could leave similar responses on both devices. In particular, we use a function generator (i.e., Keysight Technologies 33509B [111]) to generate a nearly inaudible chirp of $15kHz \sim 25kHz$ and play the chirp using a tweeter speaker [56], which is placed $25cm$ away from the wearable. Figure 4.16 shows the frequency responses of VA microphone and the two smartwatches' accelerometers. We can find that the microphone show responses from $15kHz$ $24kHz$. But we do not observe any responses on the two smartwatches. The experimental results show that the wearable's motion sensors could shield the VA system from ultrasound attacks.

4.7 Summary

In this paper, we present WearID, a wearable-assisted verification system for Voice Assistant (VA) systems (e.g., Amazon Echo and Google Home). WearID verifies whether the voice command received by the VA system comes from the legitimate user based on examining the command sound recorded in two domains (i.e., audio and vibration). In particular, WearID compares the voice command recorded by the VA device's microphone with that of the legitimate user's wearable motion sensor to calculate the cross-domain speech similarity. We show that the motion sensors of the wearable have a short response distance to sounds and exhibit different response characteristics from microphones. We further identify their complex relationship, which is hard to forge in various audible and inaudible acoustic attacks such as replay attacks and ultrasound attacks. Moreover, we develop spectrogram-based conversion method and shift 2D correlation to facilitate the comparison of the voice commands across two domains under a huge sampling rate gap (e.g., 8000Hz vs. 200Hz). WearID is easy to be deployed on the off-the-shelf wearable devices and does not require any hardware changes to the VA systems. Extensive experiments with two commodity smartwatches and 1000 commands show that WearID can verify the command sound with 99.8% accuracy in the normal situation and detect 97% fake voice commands under various audible and inaudible attacks.

Chapter 5

Protecting Public Security Using Commodity Wi-Fi Devices

5.1 Background

The portable dangerous objects such as lethal weapons, homemade bombs, and explosive chemicals have posed an increasing threat to public security. In 2013, two homemade bombs detonated near the finish line of the annual Boston Marathon, causing 3 people dead and estimated 264 injured. In 2017, a gunman opened fire on a crowd of concertgoers at Harvest music festival on the Las Vegas Strip in Nevada, resulting in 58 people dead and 546 injured. In the above terrorist attacks, it is easy for the attackers to hide dangerous objects in small baggage without drawing any attention in public places. Due to the safety concerns following the recent shooting at a Florida high school, which left 17 people dead in April 2018, this high school now only allows the students to carry clear and transparent backpacks on campus [84]. But such measures also infringe the privacy of students, and may not be effective on preventing future attacks. To reduce such threats while preserving personal privacy, it is highly demanded of a wide deployment for non-intrusive security checks at the public places (e.g., museums, theme parks and schools).

Traditional in-baggage suspicious object detection involves either manual examination (e.g., setting up checkpoint at every entrance) or dedicated equipment (e.g., surveillance camera, X-ray machine, ultra-wide-band scanner) [49, 114, 92] and incurs high cost and deployment overhead, making them hard to scale. Recently, RF signals (e.g., WiFi and 60GHz radar) have shown their great potential in many non-intrusive sensing applications. For example, WiFi signals can be utilized to recognize human activities behind the wall [125] or perform coarse-grained imaging [64]. The 60GHz radar can be utilized to differentiate the objects (but cannot categorize the objects by material types) or perform imaging with two drones [133, 139]. However, these existing RF-based approaches involve high overhead by requiring a large antenna array or specialized signals. When a target object is placed in RF environments, both the object's inner (i.e., material content) and external (i.e., dimension and shape) properties contribute to the change of the wireless signals. Although the existing work can detect, track and image objects using RF signals, none of them separates the two influencing factors or applies them to fine-grained sensing applications, such as material detection and shape imaging of the small objects in baggage.

Intuitively, most dangerous objects such as weapons, homemade bombs and explosives, are usually metal or liquid, which have significant interference (e.g., absorption, refraction and reflection) to wireless signals, while baggage is usually made of fiber, plastics or paper that allow wireless signals to pass through. Such different impacts to wireless signals suggest that it is possible to use wireless signals for detecting and identifying suspicious objects hidden in baggage. In this work, we leverage the fine-grained channel state information (CSI) that is readily available in low-cost WiFi devices to detect and identify suspicious objects hidden in baggage without intrusion (e.g., opening the bag). The basic idea is to examine the rich information of CSI complex, which includes both amplitude and phase information of wireless signals, to capture the various wireless interference caused by the materials and shapes of objects. Our system can be easily deployed to many places that still have no pre-installed security check infrastructures (e.g., airport) and require high-manpower to conduct security check such as theme parks, museums, stadiums, metro/train stations and scenic locations (e.g., Time Square). It uses the commodity WiFi to enable a low-cost and easy-to-scale solution, which provides the first-line of defense for detecting hidden suspicious objects. Our solution is timely as it demonstrates the possibility to reuse the prevalent WiFi technology to perform suspicious objects detection at every public area vulnerable to adversarial activities without introducing the high-cost security-checking infrastructure. In order to ensure that no dangerous item is carried through the entrances, our system requires to achieve low false negative rate of suspicious object detection. We focus on detecting the in-baggage suspicious objects defined as metal and liquid objects, which cover common dangerous items, and certain materials that could be confused with the dangerous items.

In particular, to identify different materials, we exploit the WiFi signals transmitting through or bypassing the object, which result in different characteristics (i.e., absorption, refraction and reflection) in the CSI complex values from antennas and their differences. Additionally, we extract the signal reflected by the object from CSI to estimate its shape (e.g., width and height) or volume based on the finding that the strength of the reflected signal is proportional to the reflection area of the object. Compared to existing work, our approach uniquely separates the wireless interference caused by two influencing factors of objects (i.e., material and shape) by exploiting different signal beams contained in the CSI complex. Our system only requires a

WiFi device with 2 to 3 antennas and can be integrated into existing WiFi networks with low costs and deployment efforts, making it more scalable and practical than the approaches using dedicated instruments (e.g., X-ray and 60GHz radar).

A number of challenges need to be addressed to achieve the proposed system using off-the-shelf WiFi. First, the measured CSI from WiFi signals can be affected by a set of object's physical properties (e.g., material, shape, size and position), thus it is difficult to distinguish the different influences and identify the object's material and shape separately. Second, WiFi signals are not very suitable for object imaging due to its relative long wavelength comparing to the size of the target objects, which causes strong diffraction resulting in low imaging resolution. Third, detecting hidden objects in baggage needs to mitigate the effects of various types of bags. To address these challenges, we develop two system approaches specially designed for separating the refraction signals and the reflection signals from the CSI complex, and recognizing the object's material and shape, respectively. Our system eliminates the raw phase noise in CSI and reconstruct the CSI complex, which can robustly capture the dominant interference caused by material of suspicious objects even when the objects are hidden in the baggage. We also derive the reflection channel from CSI complex, which enables us to estimate the object's shape and volume at a finer level using the long-wavelength WiFi signals.

We summarize the main contributions of this work as follows:

- We demonstrate that the readily available WiFi signals from low-cost devices can penetrate vision-blocked baggage and facilitate suspicious object detection and identification without dedicated devices or signals.

- We exploit the rich information in CSI complex to detect suspicious in-baggage objects and identify their categories (i.e., metal and liquid).

- We develop reflection-based risk level estimation method to determine the risk level of suspicious objects based on the estimated volume for liquid and the shape imaging for metal. We show that the pure reflection from the object can be extracted from the imperfect CSI (affected by unpredicted shift) in the WiFi device without requiring large antenna array or modifying the transmissions.

- Extensive experiments with 15 representative objects, 6 types of bags/boxes are conducted over a 6-month period. We show that our system can achieve over 95% and 90% accuracy for identifying the suspicious object and determining its material type and achieve an average error of 16ml and 0.5cm for estimating liquid volume and metal object's shape.

5.2 Related Work

Recently, there have been increasing security concerns at many public scenarios (e.g., security checkpoint of entrances) where object detection is urgently required. As traditional approaches, the vision-based techniques [49, 122] use infrared or regular cameras to identify objects according to their color, shape, texture, and temperature. These approaches, however, are sensitive to the environmental light intensity and either require a clear line-of-sight (LOS) between the object and cameras or require the target objects to have a relatively high temperature to be detected.

Moreover, a couple of studies adopt dedicated devices (e.g., [114, 58, 133]) to recognize target objects when the LOS is blocked. For instance, X-ray imagery [114] and CT volumetric imagery [58] have been used to obtain a 2D and 3D image of the baggage/parcel item for dangerous objects (e.g., firearms) detection, respectively. RadarCat [133] uses Frequency Modulated Continuous Wave (FMCW) radar operating in 60 GHz band to recognize different objects. Ultra-wide band phased array radar can also be used to image objects by seeing through the wall [92]. However, these approaches rely on expensive and specialized equipment, which do not facilitate the wide deployment in practice. Recently, RF-based sensing has drawn considerable attention. TagScan [120] deploys cheap RFID tags to identify the material type and image the horizontal profile of a target, but it requires a specialized tag reader, and it is not known whether it can be applied to in-baggage object detection. RF-Capture [30, 29] could capture the human figure (i.e., a coarse skeleton) leveraging the reflected RF signals through a wall with specialized devices, but it is dedicated for large human body and is questionable on identifying the materials of small objects. Dinesh et.al. [37] aims to utilize everyday commodity radios (i.e., smartphone) to detect and locate hidden objects leveraging the backscatter signal measurements, but it is hard to separate the influence of the object's material and size only from backscatter signal.

Due to the prevalence of WiFi devices, a recent study [64] explores the feasibility of achieving computational imaging by leveraging WiFi signals. The researchers operate Universal Software Radio Peripheral (USRP) at 2.4 GHz band to image objects such as leather couches and metal shapes. But this method requires a large antenna array and is not sufficient to identify objects in a fine granularity manner, such as distinguishing the material of the objects. Furthermore, a set of studies use WiFi signals to sense minute human body movements to recognize/track human activities [125] and walking directions [128]. While these approaches mainly focus on exploiting the changes of fine-grained WiFi measurements (i.e., Channel State Information (CSI)) to sense human body movements, using WiFi signals to recognize small objects (e.g., water bottles, beverage cans, and knives) and different materials remains open.

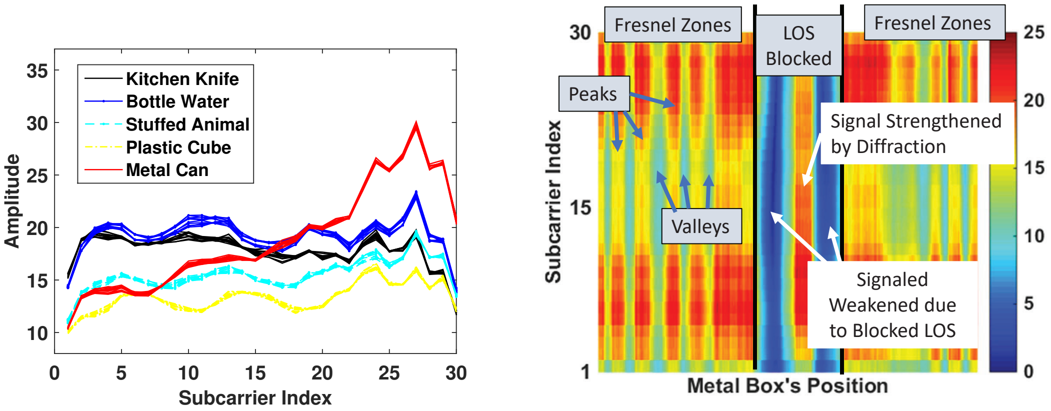
In this work, we conduct the first study to explore the feasibility of using low-cost off-the-shelf WiF devices to differentiate materials and types of the objects hidden in personal luggage or package boxes, which involves more challenges such as the different small objects in unknown positions of various bags or boxes. By exploring the rich context of CSI affected by the target object, we demonstrate that our approach can accurately estimate the inner nature (i.e., material) and outline properties (i.e., dimension/shape) of the hidden objects.

5.3 Preliminaries & System Design

5.3.1 Preliminaries

Existing work has shown that the wireless channel of a stable WiFi environment could be easily changed by adding an object, for instance, a person, a bag or a cup. The intuition behind this is that interferences caused by the additional object, including absorption, reflection, and refraction of WiFi signals, largely change the multi-path effect of the existing WiFi environment and result in a different wireless channel. In this work, we find that such wireless channel changes caused by the additional object could be different due to different materials and shapes of the objects. To illustrate this intuition, we conduct some preliminary studies by respectively placing 5 common objects (i.e., a kitchen knife, a bottled water, a stuffed animal, a plastic cube, and a metal can) at the same position between a WiFi transmitter and a receiver that are one meter apart. Figure 5.1(a) presents the CSI amplitudes across 30 subcarriers corresponding to these objects. We can see that the CSI amplitude at each subcarrier is affected by the objects differently due to the object's different physical properties (e.g., material, size and shape). However, we find it is difficult to further distinguish the materials, shapes or sizes of different objects by examining the CSI amplitudes. Thus it is necessary to separate the wireless channel changes caused by objects' materials, shapes and sizes and explore more useful information from CSI in addition to its amplitude.

In addition, we notice that moving the object to multiple positions with a single-antenna setup can imitate the large antenna arrays [30], which could be exploited to perform object imaging. We illustrate this potential by conducting an experiment in which we move a metal box along a rail that is perpendicular to the line of sight (LOS) between a pair of single-antenna WiFi transmitter and receiver. Figure 5.1(b) shows the CSI amplitudes of 30 subcarriers collected while we move the metal box. We find that the metal box causes the strongest decrease in the amplitude when it blocks LOS, mainly because metal hardly let WiFi signals go through it.



(a) Static objects' interference

(b) Moving objects' interference

Figure 5.1: Different objects' interference to the Wi-Fi signal (in CSI amplitude).

Such signal attenuation could be exploited to determine one dimension of the object (e.g., width or height). In addition, the repetitive peaks and valleys at all subcarriers on both sides of the LOS show the Fresnel Zones [119], which correspond to an object's reflection capability and can be utilized to estimate its' reflection surface area (related to both height and width). Ideally, we can estimate the dimension of an object by moving it crossing the LOS of a wireless channel like this. However, the strongest attenuation area due to the blocked LOS could be interfered by the diffraction of the WiFi signal at the small object (the strengthened signal in blocked LOS in Figure 5.1(b)). And estimating the dimension of an object directly using the peaks and valleys in Fresnel zone is not reliable because they are largely affected by the object's position and multi-path signals. Thus we need to seek solutions to extract the real reflection signal and reduce the influence of diffraction caused by the object to facilitate imaging the object.

5.3.2 Threat Model

Our work targets an adversary who intentionally or unintentionally carries dangerous items (e.g., lethal weapons, home- made bombs, combustibles) to public venues. Unlike tight security-checking areas (e.g., airports), there are two major types of areas vulnerable to adversarial activities: Places not having pre-installed security check infrastructures and employing high-manpower to perform security checks, such as theme parks, museums and stadiums, and the other kind even not having regulated checking process in place such as metro/train stations and scenic locations (e.g., Time Square). To launch an adversarial activity, the attacker usually hides the dangerous item in his bag or metal/plastic container to avoid being easily detected. In this

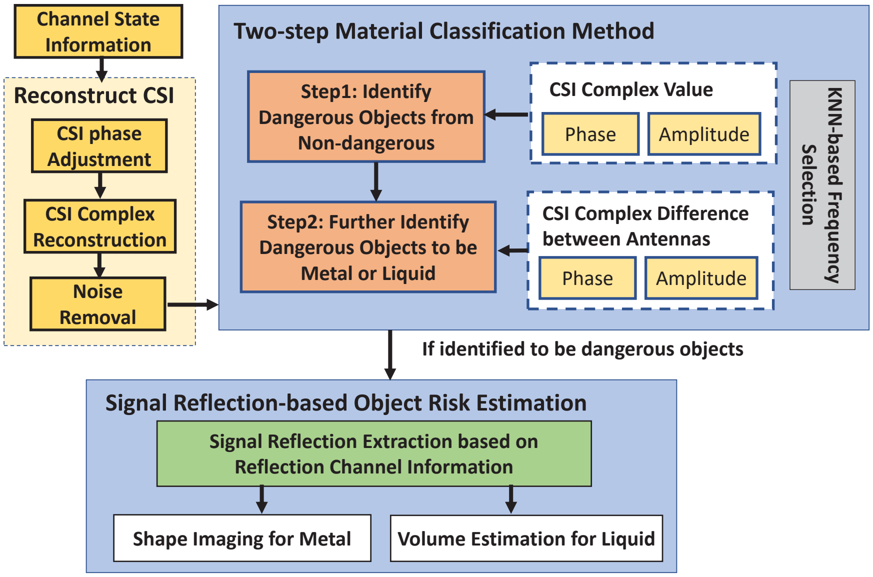

Channel State Information
Reconstruct CSI
CSI phase Adjustment
CSI Complex Reconstruction
Noise Removal
Two-step Material Classification Method
Step1: Identify Dangerous Objects from Non-dangerous
CSI Complex Value
Phase
Amplitude
Step2: Further Identify Dangerous Objects to be Metal or Liquid
CSI Complex Difference between Antennas
Phase
Amplitude
KNN-based Frequency Selection
If identified to be dangerous objects
Signal Reflection-based Object Risk Estimation
Signal Reflection Extraction based on Reflection Channel Information
Shape Imaging for Metal
Volume Estimation for Liquid

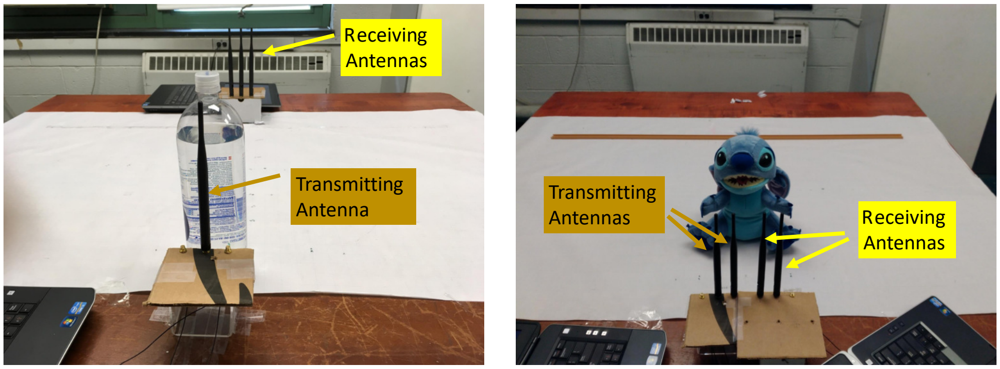
Figure 5.2: System overview.

work, we focus on detecting the suspicious objects including metal and liquid objects, which cover most of the dangerous objects that people could carry in baggage. More specifically, the metal objects such as aluminum cans, laptops, batteries and metal boxes can be used for homemade bombs, while the kitchen knives, guns and steel pipes can be directly used as weapons. Moreover, the liquids such as water, acid, alcohol and other chemicals in retainers might cause explosions.

5.3.3 System Design

System Requirements. Our system aims to automatically detect the suspicious objects in the aforementioned places. To achieve this goal, the design requirements of our system include: 1) A low false negative classification rate of suspicious objects in order to ensure adversaries cannot carry dangerous objects passing the security check; 2) A low system cost that is necessary to enable wide deployment at the places, which is lack of pre-installed security check infrastructures (e.g., museums, schools, stadiums, and train stations);3) Capability of identifying small objects that could be hidden in baggage; 4) Identifying both material and shape simultaneously.

System Overview. To facilitate the suspicious object detection and identification, we design a novel system leveraging CSI measurements readily available in existing WiFi devices. As illustrated in Figure 5.2, our system takes the CSI from a pair of WiFi transmitter and



(a) Setup1: The Tx and Rx are placed apart to identify material

(b) Setup2: The Tx and Rx are placed closely to capture reflected signal

Figure 5.3: Two experimental setups for object material identification and risk level estimation.

receiver as input. The system then performs *CSI Phase Adjustment* and *Complex CSI Reconstruction*, which correct the CSI phase drifting and reconstruct the CSI complex including amplitude and corrected phase to describe the channel in an appropriate manner. Our system then performs *Noise Removal* to mitigate the interference of environmental noises. After that, the preprocessed CSI measurements would go through two main components: 1) *Two-step Material Classification* focuses on analyzing the material type to detect the suspicious objects in the black box while decreasing the influence factors including the object's size, shape and position; 2) *Signal Reflection-based Object Risk Estimation* can extract the reflected signal off the object from the CSI to perform shape imaging and volume estimation to estimate the risk level of the suspicious objects.

More specifically, *Two-step Material Classification Method* is performed to first identify existence of the suspicious objects by leveraging the CSI complex values and then derive the CSI complex difference between antennas to further distinguish the suspicious objects to be metal or liquid by capturing their minute differences. *KNN-based Feature Selection* is performed to select the good subcarriers for the CSI complex and CSI complex difference. Given the material identified, *Signal Reflection-based Object Risk Estimation* is performed to further estimate the suspicious object's risk level based on extracted reflections from the CSI complex. In particular, the object's risk level is determined by performing the shape imaging for the metal and the volume estimation for liquid in containers. This is because the liquid would have a higher risk level if its volume exceeds the permissible limit and metal piece is more suspicious if it has a similar shape to weapons.

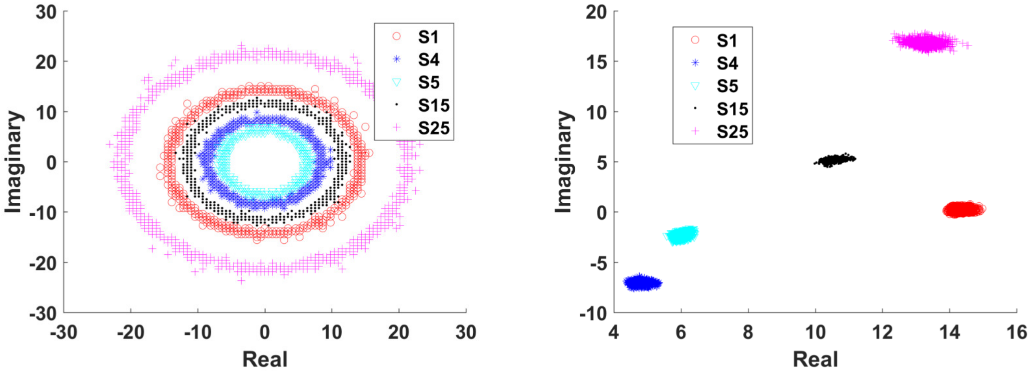
Two WiFi-antenna Setups. Two uniquely setups (as shown in Figure 5.3) are designed for *Material Classification* and *Object Risk Estimation* respectively, by meeting the various requirements of the two different goals. When identifying the object's material, our system requires to focus only on the material influence on the CSI and reduce the influencing factors caused by the object's shape, size and position. In setup one (Figure 5.3(a)), the object is placed close to the transmitting antenna, while the receiving antenna is placed apart. By blocking much more spherical area of the transmitting signal, the object close to the antenna heavily affects the transmitting signals. Thus the signal beams passing through the object or bypassing the object' surface dominate the signal beams arriving at the receiver (except the multi-path from permanent furniture), which are more related to the object's material influence. Moreover, due to the transmitting antenna's small elevation angle (e.g., 40 degree for 6dbi omni-antenna), the signals are more focused to a small area on the object, which reduces the influence caused by object's size and shape. Additionally, the object blocks more inner Fresnel zones near the transmitter [119], which further weakens the arriving diffraction and reflection signals and reduce the influence of sizes, shapes and positions. Thus we can focuses on the object's material influence to CSI. Differently, the setup two (Figure 5.3(b)) amplifies the influence caused by the object's shape and size by placing the object away from the closely settled transmitter and receiver. It is good for imaging object's front face based on reflection and avoid the reflection from the short object's upper face. Note that these two setups can be combined in practical scenarios. For example, we can deploy two WiFi device pairs along a conveyor belt in most entrance check points to facilitate material identification and shape imaging in sequence automatically.

5.4 CSI Complex Value Reconstruction

To facilitate the object detection and identification leveraging WiFi signals, we exploit CSI, the fine-grained description of the wireless channel, to capture the minute differences of the channel state change introduced by different objects. Specifically, the CSI with respect to each subcarrier is expressed as a complex value as follows:

$$H(f_k) = |H(f_k)|\, e^{j\angle H(f_k)}, \tag{5.1}$$

where $H(f_k)$ describes the channel response for the subcarrier with central frequency f_k, $|H(f_k)|$ and $\angle H(f_k)$ denote the corresponding amplitude and phase, respectively. It describes how the signal propagation is affected and reveals the impact of multipath effects between a pair of transceivers. The wireless channel will experience various impacts such as absorption, reflection



(a) Raw CSI complex value (b) Reconstructed CSI complex value

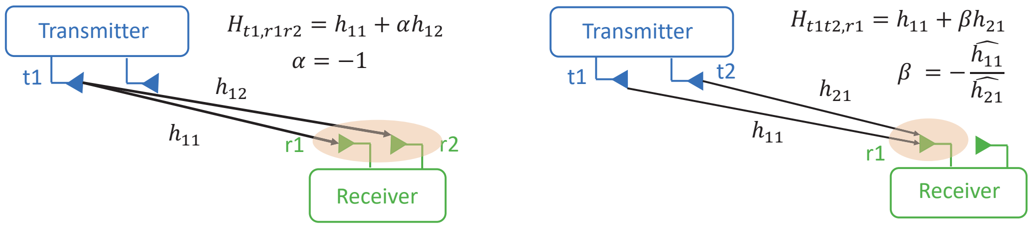
Figure 5.4: The CSI before and after phase adjustment in the complex plane.

and refraction by any object in the surrounding wireless environment, resulting in the changes of the CSI amplitude and phase at each subcarrier. However, the raw CSI extracted from WiFi signals could be distorted by the unpredicted phase shift and time lag caused by the non-synchronized transmitter and receiver [101]. Most studies thus only use the CSI amplitude instead of the complex CSI value to characterize the wireless channel. Figure 5.4(a) shows the raw CSI complex values for 5 randomly chosen subcarriers across 1000 packets. We find that the raw CSI complex show the "doughnut" shape for each subcarrier because their amplitudes keep constant but the phases are much random. Thus the CSI phase needs to be adjusted for a more accurate description of the wireless channel.

Existing studies utilize the phase difference between adjacent subcarriers [117] or antennas [71] to remove the unknown phase shift, which may lose some useful information from the original CSI phase. In this work, we adopt the phase unwrapping [61] and the linear transformation method (similar to [101]) to adjust the raw CSI phase. In particular, we first unwrap the raw phase across all the subcarriers of each packet, which is wrapped within the range $[-pi, pi]$. Then a linear transformation is applied to the unwrapped phase to remove the phase shift offset at each subcarrier and thereby derive the adjusted phase $\angle\hat{H}(f_k)$ as:

$$\begin{cases} b = \frac{\angle H(f_{30}) - \angle H(f_1)}{f_{30} - f_1}, \\ a = \frac{1}{30}\sum_{k=1}^{30} \angle H(f_k), \\ \angle\hat{H}(f_k) = \angle H(f_k) - bf_k - a \end{cases} \tag{5.2}$$

where $k, k = 1, 2, ..., 30$ is the index of the 30 subcarriers and $f_k, f_k = -28, -26, ..., 28$ is the frequency point index of the real OFDM subcarrier [110](Table 7-25f).


Transmitter
$H_{t1,r1r2} = h_{11} + \alpha h_{12}$
$\alpha = -1$
t1
h_{12}
h_{11}
r1
r2
Receiver
Transmitter
$H_{t1t2,r1} = h_{11} + \beta h_{21}$
$\beta = -\frac{\widehat{h_{11}}}{\widehat{h_{21}}}$
t1
t2
h_{21}
h_{11}
r1
Receiver

(a) Combined channel perceived by two receiving antennas

(a) Reflection channel at each receiving antenna to extract WiFi reflection signals

Figure 5.5: Combined channel and reflection channel.

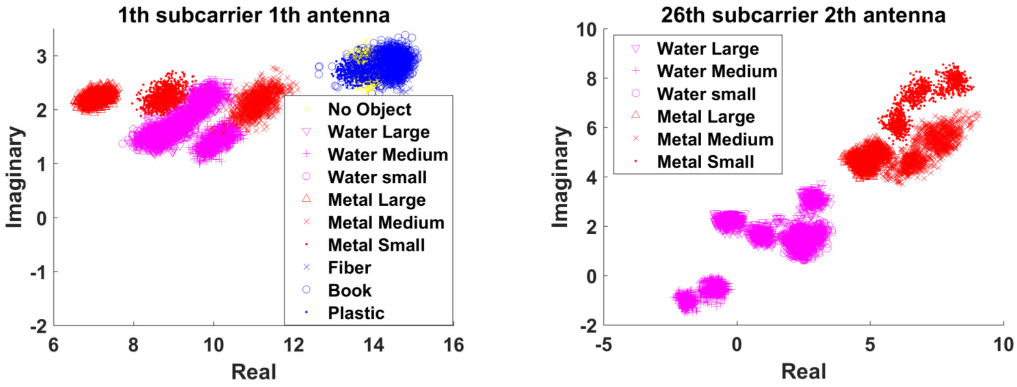
Given the adjusted CSI phase, we reconstruct the complex form of CSI as $\hat{H}(f_k) = |H(f_k)|\, e^{j\angle\hat{H}(f_k)}$, where $\angle\hat{H}(f_k)$ is the adjusted CSI phase. The reconstructed CSI complex $\hat{H}(f_k)$ accurately depicts the frequency response of each subcarrier in term of both amplitude and phase as shown in Figure 5.4(b), where the CSI complex of different subcarriers form their respective clusters in the complex plane. In a static wireless environment, both the CSI phase and amplitude maintain constant accordingly, which thus facilitates our two major system components to analyze the channel state changes introduced by the target objects with different materials, shapes and sizes.

5.5 Two-step Material Classification based on CSI Complex Value

In this section, we focus on the materials identification with our two-step method with the reconstructed CSI complex in Section 5.4, because the material (i.e., metal, liquid and unsuspicious) directly reflects whether the target object is suspiciously dangerous or not. The basic idea is to capture the wireless channel differences caused by different materials of target objects leveraging the CSI information. Different materials have different attributes on absorbing and refracting the WiFi signal, and such differences are reflected as the changes on CSI measurements. For example, 1) paper, cloth and plastics allow large portion of signal to penetrate; 2) the metal objects reflect a large portion of wireless signal and have the rest of signal scattered along its surface; 3) the liquid such as water has medium reflection but in the meanwhile allow a portion of signal to pass through.

5.5.1 Examining the Material's Impact on Channel State

We first examine how different materials influence the CSI complex. Figure 5.6 (a) shows the CSI complex values with respect to one subcarrier with 9 different objects in Setup One



(a) Differentiate dangerous objects from non-dangerous based on CSI

(b) Differentiate metal from liquid based on CSI difference

Figure 5.6: Two-step material identification method based on CSI complex and CSI complex difference.

(Figure 5.3(a)), where each object was tested three times with slight position and orientation changes. We can observe that the suspicious objects such as metal and water have their CSIs clustered together. In comparison, the CSIs corresponding to other objects such as fiber, books and plastics form another different cluster overlapped with the cluster when there is no object present (i.e., yellow dots). This is because these unsuspicious objects have little interference to the wireless channel due to their electric-insulated attributes and low density. Moreover, the metal objects and the water containers of different sizes are all significantly different from the unsuspicious objects in term of CSI complex. Therefore, regardless of the sizes and shapes, the suspicious objects can be distinguished effectively from the unsuspicious objects based on the reconstructed CSI complex. Note that the most bags/boxes showing at the theme park, museum entrance are made of the non-dangerous material such as fiber, paper and plastics, and thus they have little impact to the wireless channel. Accordingly, the hidden suspicious objects could dominate the interference to CSI complex and be easily detected.

5.5.2 CSI Complex Difference between Receiving Antennas

With the capability to tell suspicious materials from unsuspicious ones, the CSI complex alone is still hard to further distinguish the different types of suspicious materials. For example, as shown in Figure 5.6(a), the CSI clusters corresponding to liquid and metal objects are close to each other. This is because these suspicious materials all heavily interfere the wireless channel. Thus we need to further distinguish their minute difference by resorting to more in-depth information such as the relative spatial information from multiple antennas. For example, different materials have different scattering effects on the RF signals when passing through the object. Therefore, we propose to leverage the CSI complex differences between any two receiving antennas to capture the minute difference of the signal scattering at multiple antennas. Assuming that the transmitter emits a symbol x at antenna $t1$, the symbols received by the two antennas $r1$ and $r2$ of the receiver would be $h_{11}x$ and $h_{12}x$ (as shown in Figure 5.5(a)), where h_{11} and h_{12} are the CSI for the $t1$-$r1$ and the $t1$-$r2$ antenna pair. Then the combined input y1 at the two receiving antennas could be defined as $y1 = (h_{11} + \alpha h_{12})\, x$. By choosing $\alpha = -1$, we define the combined channel $H_{t1,r1r2}$ between t1 and r1,r2 as,

$$H_{t1,r1r2} = h_{11} - h_{12}, \tag{5.3}$$

Under the presence of an object, the combined channel $H_{t1,r1r2}$ measures the difference between the two channel states, which removes the common factors (e.g., permanent furniture influence) at two receiving antennas, and also amplifies the minute differences on scattering effects caused by different materials. As illustrated in Figure 5.6(b), the metal and water could be differentiated by the CSI complex difference regardless of their sizes. We then utilize the CSI complex difference to identify the types of suspicious materials.

5.5.3 Two-step Method Implementation

Based on the above observations, we develop a two-step material identification method to classify the object's material within Setup One. In particular, 1) we first differentiate the suspicious objects from unsuspicious ones by leveraging the reconstructed CSI complex values as features to perform classification; 2) we next identify whether the material of the dangerous objects is metal or liquid by deriving the CSI complex differences between two receiving antennas as the features for further categorization. At each step, we apply a learning-based method to build the material profiles. During the training phase, we first apply the KNN-based feature selection method to choose CSI-based features from good subcarriers and antenna pairs. In particular, we cluster the CSI-based features with respect to each subcarrier based on KNN; then k-fold cross validation is applied to the KNN-based clusters to determine the good subcarriers and antenna pairs which show lower K-fold loss ratio than a predefined threshold when differentiating the materials at each step. Next, a learning method, such as SVM or deep learning, is adopted to train the material profile at each step. Note that, to identify the object within different baggage, we pick several representative types of bags/boxes with the target objects enclosed to build the CSI profiles. During the testing process, the CSI and CSI complex difference of target

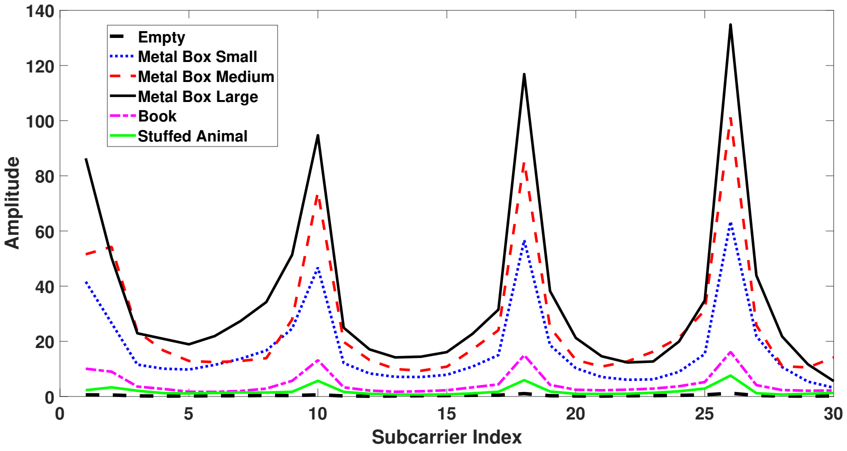
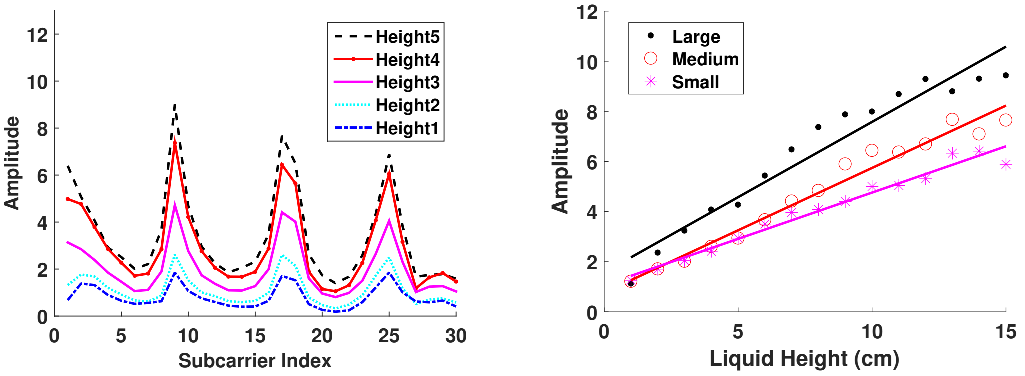


Figure 5.7: The reflection channel state information in response to different objects' reflections.

objects are compared with the pre-defined profiles for classification. As long as their material belongs to the three types (i.e., metal, liquid and unsuspicious), our system can identify them accurately. Moreover, most bags/boxes are made of unsuspicious material such as fiber, the hidden dangerous objects, if any, could dominate the impact on the CSI, which can be easily captured by our proposed system. Therefore our system can differentiate the materials of hidden target objects wrapped by various bags/boxes.

5.6 Object Risk Estimation leveraging Signal Reflection-based Object Imaging

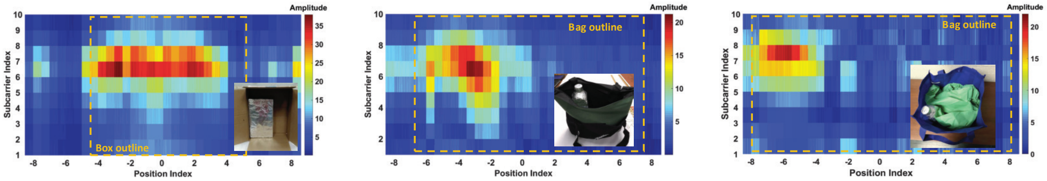
It is not sufficient to determine the risk of the suspicious objects by identifying the material only. For instance, the volume of the liquid less than a certain limit (e.g., $100ml$) is less risky and is usually allowed to be carried on flights; the metal pieces with similar shapes as the weapons (e.g., kitchen knife and soda-can bomb) are usually more dangerous. WiFi signals from off-the-shelf devices are not specifically designed for the small object imaging due to its long wavelength (e.g., $12cm$ for $2.4GHz$ and $6cm$ for $5Gz$), which would induce strong diffraction and thereby significantly decrease the imaging resolution [139]. To mitigate the effects of signal diffraction for better imaging resolution, we focus on the signals reflected from the target object to perform metal object imaging and liquid volume estimation.



(a) Increasing liquid height in the same container

(b) Increasing liquid height in the containers of different sizes

Figure 5.8: CSI amplitude changes with increasing volume of liquid.



(a) Tinfoil box in a package box

(b) Tinfoil bottle in a handbag

(c) Tinfoil bottle in a tote bag stuffed with clothes

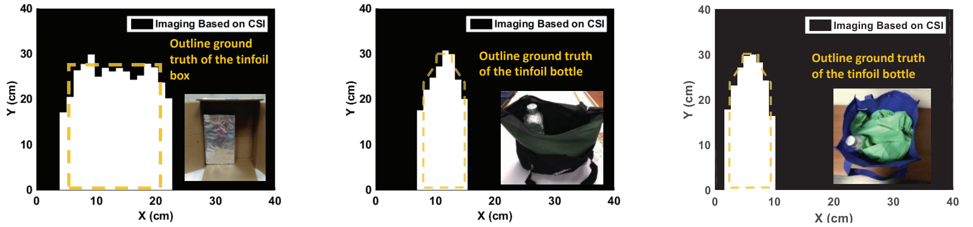
Figure 5.9: The amplitude of reflection channel state information in response to the metal objects in a baggage.

5.6.1 Extracting Reflected Signals from CSI Complex

We first introduce how to extract the signal reflected by the target object from the CSI complex based on Setup Two (i.e., Figure 5.3(b)). As shown in Figure 5.5(b), two transmitting antennas (i.e., $t1$ and $t2$) and one of the receiving antennas (e.g., ri) are considered for illustration. The channel response capturing the signals reflected from the target object only, defined as *Reflection Channel* $H_{t1t2,ri}$, can be represented as:

$$H_{t1t2,ri} = h_{1i} + \beta h_{2i}, \quad \beta = -\frac{\hat{h}_{1i}}{\hat{h}_{2i}}, \tag{5.4}$$

where h_{1i} and h_{2i} are the estimated channel states (i.e., CSI) for two antenna pairs (i.e., from transmitting antenna $t1$ and $t2$ to receiving antenna ri respectively). The weight $\beta = -\frac{\hat{h}_{1i}}{\hat{h}_{2i}}$ is calculated by $\hat{h}_{1i}$ and $\hat{h}_{2i}$, which are the channel states with no target object presented in the area of interest. When no object is placed, the signals from the transmitting antenna $t1$ and



(a) Tinfoil box in a package box (b) Tinfoil bottle in a handbag (c) Tinfoil bottle in a tote bag stuffed with clothes

Figure 5.10: Using the WiFi reflections extracted from CSI to image the metal objects in baggage.

$t2$ are combined linearly to null the reflection paths to the receiving antenna ri. Therefore the LOS and the reflected paths from the permanent furniture [74] are eliminated in the channel state information. But when an object is placed in the area, the reflected paths will become in-negligible, and the amplitude of reflected channel information $H_{t1t2,ri}$ implies the object's reflecting capability. Figure 5.7 shows an example of the amplitudes of the reflected channel state information (reflection CSI) perceived by receiving antenna $r1$ with different objects presented. In particular, empty environment renders close to zero amplitude for all subcarriers of $H_{t1t2,ri}$ amplitudes (i.e., black dash line), whereas the unsuspicious objects such as book and stuffed animal result in none zero amplitudes but much lower than metal objects. Moreover, we also find the sizes of the metal objects are proportional to the reflected CSI amplitudes of all subcarriers, and different subcarriers also have different sensitivity when they are reflected from the objects. The above observations confirm the effectiveness of our proposed method on capturing the signals reflected from target objects by eliminating the LOS and multipath signals. We next leverage the captured reflected signals to estimate the liquid volume and perform metal object imaging.

5.6.2 Volume Estimation for Liquid Objects in Baggage

To estimate the liquid volume, we conduct some experiments under Setup Two (i.e., Figure 5.3(b)), which involves a small bottle as the target object with 5 different water volumes ranging from empty to full. The amplitudes of the reflected CSI (i.e., $H_{t1t2,r1}$) corresponding to different water volumes are shown in Figure 5.8(a). It is easy to find that the larger the water volume, the greater the reflected CSI amplitude across all 30 subcarriers due to the increasing reflecting surface. To further quantify the relationship between the water volume and the amplitude of reflection CSI, we select 15 different water heights in three cylindrical containers of different diameters (i.e., large, medium and small). As shown in Figure 5.8(b), we observe that the amplitude of the reflected CSI is linearly proportional to the water heights for all three containers. Moreover, the larger container has faster growth rate on the CSI amplitude due to the larger reflecting surface under the same water height. Therefore, as long as the container's diameter is determined, the liquid's volume can be derived by following a linear regression model. In this work, we assume the liquid is kept in the nonmetal cylindrical containers such as plastic or glass bottle. If the liquid is in metal containers, it would be identified as metal objects based on our material identification method in Section 5.5.

Based on our preliminary study, the liquid volume estimation consists of two steps, diameter determination and liquid height estimation. To determine the diameter of the liquid container, we adopt the same method of determining the metal object's width as in Section 5.6.3. Once the liquid container diameter is obtained, we apply two different methods, the linear regression method and the neural network-based method, to estimate the liquid height by leveraging the frequency selection property across multiple subcarriers. Specifically, the linear regression method aims to build the linear regression relationship between the CSI amplitude and liquid height for each subcarrier, and integrate the prediction results from all subcarriers to derive the liquid height. The neural network-based method predict the unknown height of the liquid in containers by building a neural network model, which takes the amplitudes of all subcarriers with respect to different liquid heights as the training feature vector. At last, the liquid volume is easily obtained based on the estimated container diameter and the liquid height.

5.6.3 Shape Imaging for Metal Objects in Baggage

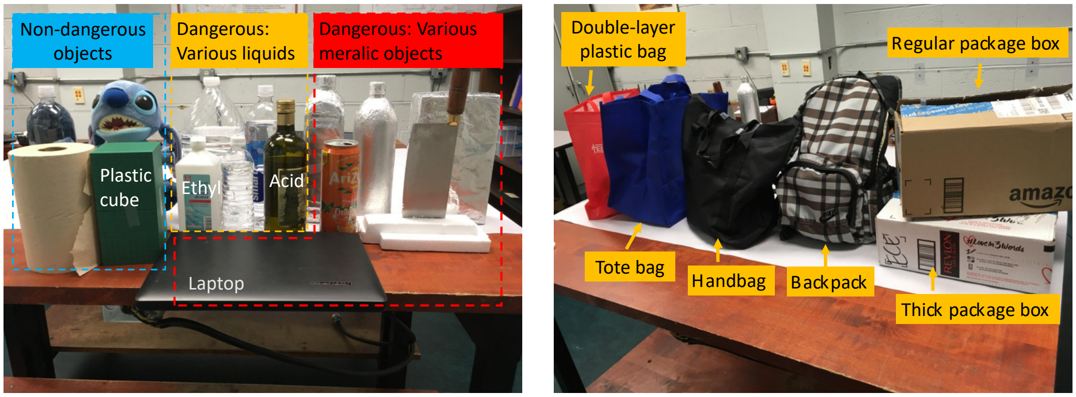
Unlike the existing studies relying on large antenna arrays to determine the shape of metal objects, we propose to image the in-baggage metal objects using commercial WiFi devices with a limited number of antennas while the baggage is moved by the conveyor belt, which is available at many entrance check points. Figure 5.9 shows the reflection channel response $H_{t1t2,r1}$ when the target object is in an opaque baggage, which moves along the track in parallel with the antenna array. The rectangular box and the water bottle are covered with tinfoil to imitate the metal objects of different shapes that are similar to homemade bombs. We find that the reflected channel response is greater when the target object is close to the central line between the transmitter and receiver, where strong reflection is usually incurred by the object. Moreover, as shown in Figure 5.9(a) and (b), both the width and position of the target object hidden in the baggage or box can be clearly identified from the reflected CSI amplitude (e.g., red color). Furthermore, when there are multiple objects in the same baggage, such as the metal object together with clothes as shown in Figure 5.9(c), the metal object dominates the reflection signals and can still be distinguished and imaged. Note that our system can detect the existence of suspicious objects even if liquid and metal objects are in the same baggage and the object imaging includes both objects. We therefore develop a threshold-based approach to capture the outline of the metal objects and separate them from other non-suspicious objects, including the baggage. We first estimate object's width, which is proportional to the object moving distance that cause reflections above a threshold by using $d = \gamma \hat{d}$, where $\hat{d}$ is the estimated width from reflection CSI amplitude and γ is the ratio, which is related to the short wavelength of WiFi signal. Once the width of the object is determined, we proceed to estimate the object's height based on the fact that the reflection CSI amplitude is proportional to the reflection area. The estimation of the metal object's height is similar to the method in Section 5.6.2. Figure 5.10 shows the final imaging results of the metal objects based on the reflection CSI amplitude of Figure 5.9. It is encouraging to find that the metal object's outlines can be well recognized, which are very close to the actual shape of the target objects even when it is hidden with other objects in the baggage.

5.7 Performance Evaluation

5.7.1 Experimental Methodology

Experimental Setup. We implement our system on a pair of laptops, which are equipped with IWL 5300 wireless cards and three 6dBi omnidirectional dual band rubber ducky antennas. The two laptops are placed upon a wooden table in a typical indoor room, and we employ two setups as shown in Figure 5.3 to perform material identification and risk level estimation, respectively. The laptops are running Ubuntu 10.04 LTS with the kernel 2.6.36, and the WiFi card works at $5GHz$ frequency band with the transmission rate $100pkt/sec$. During data collection, two people are in the room standing by the table to imitate the practical scenarios.

Target Objects. We evaluate our system with the combination of 15 different target objects in three categories (i.e. metal, liquid and non-dangerous) and 6 representative bags/boxes in three categories (i.e., backpack/handbag, cardboard boxes, thick plastic bag) as shown in Figure 5.11. For the material identification, we put each of the 15 objects in 6 bags/boxes respectively and experiment under Setup1 in Figure 5.3. Each experiment is repeated 5 times while slightly changing the object's position and orientation. For dangerous object risk level estimation, we place the metal objects across multiple positions under Setup2 (i.e., Figure 5.3(b))



(a) Fifteen objects in 3 categories

(b) Six different bags and boxes

Figure 5.11: Various target objects and bags/boxes in the experiment.

to estimate the size (i.e. width and height). Moreover, we have the three different size containers (i.e., large, medium and small) filled with different volumes of liquid to estimate liquid volume. Overall, over 800 experimental data traces are collected during a 6-month period to evaluate our proposed system.

Evaluation Metrics. To evaluate the material identification method, we define*Identification Accuracy* as the ratio of the correctly identified objects over all the tested objects, and define *Detection Rate* as the ratio of correctly identified objects over the total objects of the same material. A high detection rate of the suspicious object reflects a low false negative rate, which guarantees that few suspicious objects could pass the security check. To evaluate the risk level estimation, we utilize *Size Estimation Error (cm)* to measure the estimation of the metal object' width and height and *Volume Estimation Error (ml)* for the estimation of the liquid volume.

5.7.2 Material Classification

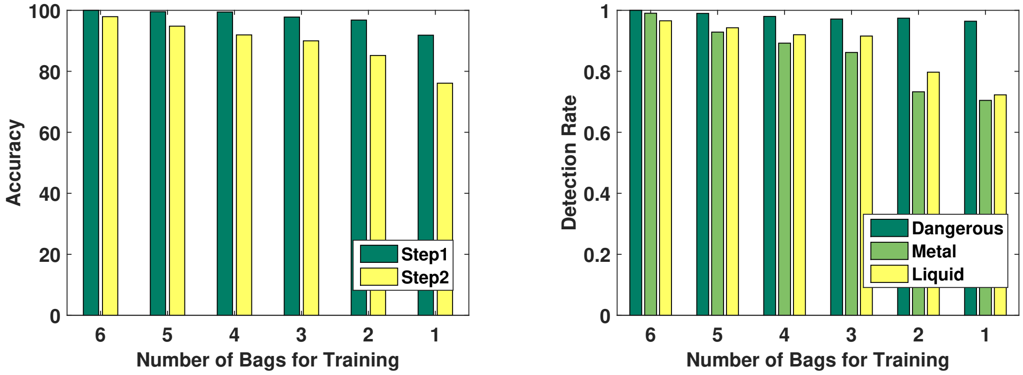
We first evaluate our material identification of the object hidden in various bags, especially when different number of bags are used for training the profile. Figure 5.12 shows that our system can achieve high accuracy in identifying the object's material when they are put in different bags. In particular, given the combination of all the 15 objects and the 6 bags in our profile, Figure 5.12(a) shows that our system can achieve 99% accuracy in classifying dangerous objects from non-dangerous (step1) and 97% accuracy to further differentiate the dangerous objects to be metal and liquid (step2). Figure 5.12(b) further shows that the overall detection rate for the dangerous material, metal and liquid are 99%, 98% and 95%. Moreover, we find that the material identification accuracy reduces a little bit as the number of bags used for profile training decreases. For example, when using half of the bags (i.e., one bag/box from each of three categories) for training, the step1 and step2 accuracy of our material classification method fall to 95% and 90% while the detection rate of dangerous objects decreases to 94%. The overall detection rate for metal and liquid objects fall to 90% and 92%. This is because the bags and boxes, though made of non-dangerous material, still induce slightly different interferences on the wireless channel, thereby resulting in the errors in material detection. But because the bags used in testing phase have the similar material with the bags/boxes used in building training profile, our system still achieves high material identification accuracy. Additionally, regardless of the number of bags used in training phase, our system can keep over 93% accuracy of detecting the dangerous material as shown in Figure 5.12(b).

Figure 5.13 presents a more challenging scenario, where only half of the objects in each of the three object categories are trained to build the profile. Figure 5.13(a) shows that in this scenario, if all the bags are used for training, we can achieve over 95% accuracy for step1 and 90% for step2. The overall detection rate for the dangerous materials is 96%, and the detection rate for metal and liquid objects fall to 82% and 91% as shown in Figure 5.13(b). Furthermore, we find that the material identification accuracy also reduces with decreasing number of bags used for training, due to the different bags' slight different interference. In particular, when half of the objects and half of the bags are used for training the profile, our system can achieve 91% and 85% accuracy for step1 and step2 of our material classification and the detection rates for the dangerous, metal and liquid are around 90%, 78% and 85%. The results show that our system can efficiently identify the object made of dangerous material and further classify the dangerous material types in the more complex scenarios. In an extreme case, when half of the objects and only one bag are chosen for training, the detection rate for all dangerous materials is still over 89%. The results confirm that our system can efficiently recognize the object by its material regardless of their shapes and sizes or what bags they are hidden in.

5.7.3 Risk Level Estimation based on Object Imaging

We next evaluate the performance of our system on estimating the risk level of the objects through object imaging (i.e., metal object size and liquid volume).

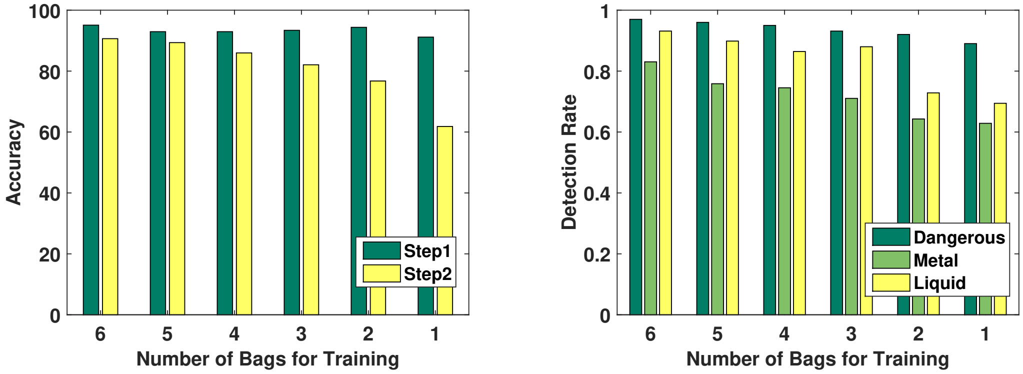
Metal Object Size Estimation. Figure 5.14(a) shows the results of our system on estimating the sizes of different metal objects. We find that our system can achieve cm-level accuracy on the size estimation of metal objects. In particular, over 80% estimation error of the metal object's widths and heights are within 0.7cm and 90% within 1cm. The average errors for



(a) Accuracy for two material identification steps

(b) Detection rate for the various objects

Figure 5.12: Material identification with different number of baggage in profile.



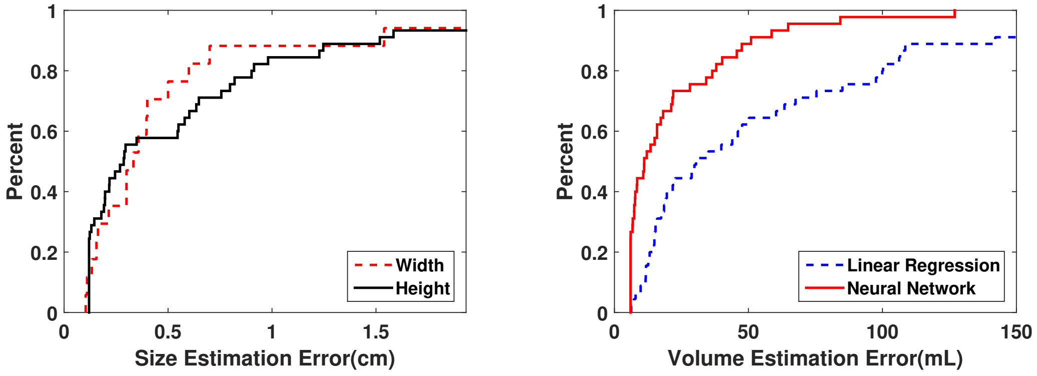
(a) Accuracy for two material identification steps

(b) Detection rate for the various objects

Figure 5.13: Material identification with half objects and different number of baggage in profile.

estimating the metal object's width and height are $0.3cm$ and $0.5cm$, respectively. The results show that our system can estimate the metal objects' size accurately, which is good to perform accurate object imaging and infer whether the metal object is suspicious to be deadly weapons or bombs.

Liquid Volume Estimation. The performance of liquid volume estimation is presented in Figure 5.14(b), where we apply two different methods, linear regression and neural network for the volume estimation respectively. We find that both methods can achieve high accuracy on liquid volume estimation. The neural network-based method achieves even higher accuracy with the median error as small as $16ml$. Moreover, over 80% estimation errors are within 35ml. The results validate that our system can accurately estimate the liquid volume, and provide



(a) Metal objects' width and height

(b) Liquid volume estimation

Figure 5.14: Accuracy of estimating the object's size and volume.

significant information to derive the risk level of liquid objects.

5.8 Summary

This work explores the feasibility of using off-the-shelf WiFi signals to detect suspicious objects (i.e., metal and liquid objects) hidden in baggage without penetrating into the user's privacy. Our solution is timely as it demonstrates the possibility to reuse the prevalent WiFi technology to perform suspicious objects detection at every public area vulnerable to adversarial activities without requiring the installation of high-cost security-checking infrastructures. The designed system can also estimate the risk level of the target object through object imaging to estimate the shape/volume of the metal/liquid objects. Specifically, we deploy two different system setups for separating the refraction signals and the reflection signals from the CSI complex and recognizing the object's material and shape, respectively. Our system removes the raw phase noise in CSI and reconstructs the CSI complex, which can robustly capture the dominant interference caused by the suspicious material even when the object is hidden in the baggage. We also derive the reflection channel from CSI complex that can enable us to estimate the object's shape and volume at a fine level using the long-wavelength WiFi signals. Extensive experiments are conducted with 15 objects and 6 bags over a 6-month period. The results show that our system can detect over 95% dangerous objects in different types of bags and successfully identify 90% dangerous material types. In addition, our system can achieve the average errors of 16ml and 0.5cm when estimating the shape/volume of the metal/liquid object, respectively.

Chapter 6
Conclusion

In this dissertation, we examine the security threats and opportunities of the mobile devices (e.g., smartphones and wearable devices) regarding to personal privacy and public security. We demonstrate that the mobile devices could leak the user's private information such as social relationships, demographics and ATM PIN numbers, which causes serious security breaches. We also show that the mobile technologies could be leveraged well to protect not only the personal privacy but also the public security. In particular, we present a scalable inference system that has the potential to derive people's activities at daily visited places leveraging surrounding access points and utilize such information to infer the fine-grained social relationships and demographics. This implemented system only uses the simple signal features of surrounding access points such as MAC addresses and Received Signal Strengths without sniffing the Wi-Fi traffic data. Moreover, we develop a PIN-sequence inference framework to recover the user's secret key entries when the user accesses the key-based security systems such as ATM keypads and regular keyboards. The system does not require any training or contextual information, which makes it applicable in real world adversarial contexts. Furthermore, we propose a wearable-assisted verification system for Voice Assistant (VA) systems (e.g., Amazon Echo and Google Home), which verifies whether the voice command received by the VA system comes from the legitimate user based on examining the voice commands recorded in two domains (i.e., audio and vibration). Finally, we explore the feasibility of using off-the-shelf WiFi signals to detect suspicious objects (i.e., metal and liquid objects) hidden in baggage without penetrating into the user's privacy. The designed system can further estimate the risk level of the target object through object imaging to estimate the shape/volume of the metal/liquid objects. This solution is timely as it demonstrates the possibility to reuse the prevalent WiFi technology to perform suspicious objects detection at every public area, which does not require the installation of high-cost security-checking infrastructures.

Bibliography

[1] All about skimmers. http://krebsonsecurity.com/all-about-skimmers/.

[2] Android sensor event.
http://developer.android.com/reference/android/hardware/SensorEvent.html.

[3] How strong is your password?
https://www.msecure.com/blog/how-strong-is-your-password/.

[4] Invensense motionfit sdk quick start guide. http://store.invensense.com/datasheets/invensense/AN-MPU-9150IMF.pdf.

[5] Is it acceptable to wear a watch on the right wrist?
http://www.askandyaboutclothes.com/forum/showthread.php?116570-Is-it-acceptable-to-wear-a-watch-on-the-right-wrist.

[6] Malicious cloned games attack google android market. naked security:.
http://nakedsecurity.sophos.com/2011/12/12/malicious-cloned-games-attack-google-android-market/.

[7] Update: Theft devices on Salina ATMs.
http://www.ksal.com/theft-devices-found-on-salina-bank-atms/.

[8] Wearable device shipments predicted to surge 173% this year.
http://www.cnet.com/news/shipments-of-wearable-device-to-surge-173-this-year/.

[9] Why wear a watch on the wrist where you're hand dominant.
https://www.reddit.com/r/Watches/comments/1wzub5/question_why_wear_a_watch_on_the_wrist_where/.

[10] Wifi scanning every 5 seconds.
http://androidforums.com/threads/wifi-scanning-every-5-seconds.631388/, 2012.

[11] Understanding wireless scanning. http://www.juniper.net/document ation/en_US/network-director1.5/topics/concept/wireless-scanning.html, 2013.

[12] American Time Use Survey: Students. http://www.bls.gov/ TUS/CHARTS/STUDENTS.HTM, 2014.

[13] American Time Use Survey: Work and employment. http://www.bls.gov/TUS/CHARTS/WORK.HTM, 2014.

[14] Knowing your digital audio recorder, 2014. http://www.audioforensicexpert.com/knowing-your-digital-audio-recorder/.

[15] The global public wi-fi network grows to 50 million worldwide wi-fi hotspots. https://www.ipass.com/press-releases/the-global-public-wi-fi-network-grows-to-50-million-worldwide-wi-fi-hotspots/,
2015.

[16] Wearable ID: Is it a fit for your campus?, 2015. https://www.cr80news.com/news-item/wearable-id-is-it-a-fit-for-your-campus/.

[17] Charts: How american men and women spend their time. http://www.usnews.com/news/articles/2013/06/24/charts-how-american-men-and-women-spend-their-time,
2016.

[18] Google maps apis. https://developers.google.com/maps/, 2016.

[19] The google maps geolocation api. https:// developers.google.com/maps/documentation/geolocation/intro, 2016.

[20] Google places api. https://developers.google.com/places, 2016.

[21] Hidden voice commands example. http://www.hiddenvoicecommands.com/white-box, 2016.

[22] Municipal wireless network. https://en.wikipedia.org/wiki/ Municipal_wireless_network, 2016.

[23] United states department of agriculture economic research service. http://www.ers.usda.gov/data-products/.aspx, 2016.

[24] Unwired labs location api. https://unwiredlabs.com/, 2016.

[25] Wifimanager. https://developer.android.com/reference/android/net/wifi/ WifiManager.html, 2016.

[26] Set up voice match on google home.
`https://support.google.com/googlehome/answer/7323910?hl=en`, 2018.

[27] Number of smartphone users worldwide from 2014 to 2020 (in billions), 2019. `https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/`.

[28] Wearable technology - statistics and facts, 2019.
`https://www.statista.com/topics/1556/wearable-technology/`.

[29] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand. Capturing the human figure through a wall. *ACM Transactions on Graph.*, 34(6):219:1–219:13, 2015.

[30] F. Adib and D. Katabi. See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013.

[31] W. Albazrqaoe, J. Huang, and G. Xing. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pages 333–345, 2016.

[32] T. B. Amin, J. S. German, and P. Marziliano. Detecting voice disguise from speech variability: Analysis of three glottal and vocal tract measures. In *Proceedings of Meetings on Acoustics 166ASA*, volume 20, page 060005. ASA, 2013.

[33] S. A. Anand and N. Saxena. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. 2011.

[34] L. Atlas and S. A. Shamma. Joint acoustic and modulation frequency. *EURASIP Journal on Applied Signal Processing*, 2003:668–675, 2003.

[35] D. Balzarotti, M. Cova, and G. Vigna. Clearshot: Eavesdropping on keyboard input from video. In *IEEE S&P*, pages 170–183, 2008.

[36] 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 (ACM CCS)*, pages 245–254, 2006.

[37] D. Bharadia, K. R. Joshi, and S. Katti. Full duplex backscatter. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, 2013.

[38] L. Blue, H. Abdullah, L. Vargas, and P. Traynor. 2ma: Verifying voice commands via two microphone authentication. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 89–100. ACM, 2018.

[39] R. B. Braga, A. Tahir, M. Bertolotto, and H. Martin. Clustering user trajectories to find patterns for social interaction applications. In *W2GIS*, pages 82–97, 2012.

[40] B. Caddy. 5 sensor technologies that are set to break out in wearables, 2019. `https://www.wareable.com/wearable-tech/5-wearable-sensor-technologies-incoming-7026`.

[41] L. Cai and H. Chen. Touchlogger: Inferring keystrokes on touch screen from smartphone motion. In *USENIX HotSec*, 2011.

[42] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[43] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530, 2016.

[44] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Y. L. Huang. Body area network security: Robust key establishment using human body channel. In *Proceedings of the 3rd USENIX Conference on Health Security and Privacy (HealthSec)*, pages 5–5, 2012.

[45] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 183–195. IEEE, 2017.

[46] Z. Chen, S. Wang, Y. Chen, Z. Zhao, and M. Lin. Inferloc: calibration free based location inference for temporal and spatial fine-granularity magnitude. In *IEEE CSE*, pages 453–460, 2012.

[47] N. Cheng, P. Mohapatra, M. Cunche, M. A. Kaafar, R. Boreli, and S. Krishnamurthy. Inferring user relationship from hidden information in wlans. In *IEEE MILCOM*, pages 1–6, 2012.

[48] N. Cheng, X. Wang, W. Cheng, P. Mohapatra, and A. Seneviratne. Characterizing privacy leakage of public wifi networks for users on travel. In *IEEE INFOCOM*, pages 2769–2777, 2013.

[49] S. Chi and C. H. Caldas. Automated object identification using optical video cameras on construction sites. *Computer-Aided Civil and Infrastructure Engineering*, 26(5):368–380, 2011.

[50] K. Crager, A. Maiti, M. Jadliwala, and J. He. Information leakage through mobile motion sensors: User awareness and concerns. In *Proceedings of the European Workshop on Usable Security (EuroUSEC)*, 2017.

[51] M. Cunche, M.-A. Kaafar, and R. Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11:56–69, 2014.

[52] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra. Contextual localization through network traffic analysis. In *IEEE INFOCOM*, pages 925–933, 2014.

[53] C. Davies. Apple watch to swap physical for virtual buttons says report, 2018. `https://www.slashgear.com/apple-watch-to-swap-physical-for-virtual/-buttons-says-report-09533696/`.

[54] P. L. De Leon, M. Pucher, and J. Yamagishi. Evaluation of the vulnerability of speaker verification to synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 20:2280 – 2290, 2012.

[55] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga. Evaluation of speaker verification security and detection of hmm-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2280–2290, 2012.

[56] P. Electronics. Pyramid car audio, 300 watt aluminum bullet horn in enclosure with swivel housing. `http://www.pyramidcaraudio.com/sku/TW28/300-Watt-Aluminum-Bullet-Horn-in-Enclosure-wSwivel-Housing`, 2018.

[57] H. Feng, K. Fawaz, and K. G. Shin. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 343–355. ACM, 2017.

[58] G. T. Flitton, T. P. Breckon, and N. M. Bouallagu. Object recognition using 3d sift in complex ct volumes. In *Proceedings of the British Machine Vision Conference*, pages 1–12, 2010.

[59] D. L. Fuller. Microelectromechanical systems (mems) applications?microphones. *Rochester Institute of Technology Microelectronic Engineering*, pages 1–43, 2005.

[60] B. Han, J. Li, and A. Srinivasan. Your friends have more friends than you do: Identifying influential mobile users through random-walk sampling. *IEEE/ACM Transactions on Networking*, 22:1389–1400, 2014.

[61] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu. Cryptographic key agreement for mobile radio. *Digital Signal Processing*, 6(4):207–212, 1996.

[62] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A.-M. Laukkanen. I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry. In *Interspeech*, pages 930–934. Citeseer, 2013.

[63] M. Hébert. Text-dependent speaker recognition. In *Springer handbook of speech processing*, pages 743–762. Springer, 2008.

[64] D. Huang, R. Nandakumar, and S. Gollakota. Feasibility and limits of wi-fi imaging. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 266–279, 2014.

[65] A. IOS. Siri, 2017. https://www.apple.com/ios/siri/.

[66] S. Jain and et al. Lookup: Enabling pedestrian safety services via shoe sensing. In *ACM Mobisys*, pages 257–271, 2015.

[67] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *ACM WMASH*, pages 110–118, 2004.

[68] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *ACM Sensys*, pages 43–56, 2010.

[69] T. Kinnunen, B. Zhang, J. Zhu, and Y. Wang. Speaker verification with adaptive spectral subband centroids. In *International Conference on Biometrics*, pages 58–66. Springer, 2007.

[70] B. Kr?mer. Classification of generic places: Explorations with implications for evaluation. *Journal of Environmental Psychology*, 15(1):3 – 22, 1995.

[71] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking (ACM Mobicom)*, pages 483–494, 2014.

[72] C. Lashkari. Types of sensors in wearable fitness trackers, 2018. `https://www.news-medical.net/health/Types-of-sensors-in-wearable-fitness-trackers.aspx`.

[73] H. Li, Z. Xu, H. Zhu, D. Ma, S. Li, and K. Xing. Demographics inference through wi-fi network traffic analysis. In *IEEE INFOCOM*, 2016.

[74] K. C.-J. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous mimo networks. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 146–157, 2011.

[75] J. Lindberg and M. Blomberg. Vulnerability in speaker verification-a study of technical impostor techniques. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[76] J. Liu, Y. Wang, k. Kar, Y. Chen, J. Yang, and M. Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. In *ACM Mobicom*, 2015.

[77] L. Liu and et al. Toward detection of unsafe driving with wearables. In *ACM WearSys*, pages 27–32, 2015.

[78] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang. When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (ACM CCS)*, pages 1273–1285, 2015.

[79] Logitech. Logitech s120 speaker. `https://www.logitech.com/en-us/product/s120-stereo-speakers`, 2018.

[80] X. Lu and E. I. Pas. Socio-demographics, activity participation and travel behavior. *Transportation Research Part A: Policy and Practice*, 33(1):1 – 18, 1999.

[81] F. Maggi and et al. A fast eavesdropping attack against touchscreens. In *IEEE IAS*, pages 320–325, 2011.

[82] 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 (ACM CCS)*, pages 551–562, 2011.

[83] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song. On the feasibility of side-channel attacks with brain-computer interfaces. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 143–158, 2012.

[84] P. MAZZEI. Florida school, on edge since shooting, requires students to carry clear backpacks. https://www.nytimes.com/ 2018/03/21/us/florida-school-shooting-clear-backpacks.html, 2018.

[85] Y. Michalevsky, D. Boneh, and G. Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *USENIX Security Symposium*, pages 1053–1067, 2014.

[86] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: your finger taps have fingerprints. In *ACM MobiSys*, pages 323–336, 2012.

[87] K. S. R. Murty and B. Yegnanarayana. Combining evidence from residual phase and mfcc features for speaker recognition. *IEEE signal processing letters*, 13(1):52–55, 2006.

[88] M. Newlands. The top wearable payment technology, 2017. https://due.com/blog/wearable-payment-technology/.

[89] X. Pan, Z. Ling, A. Pingley, W. Yu, N. Zhang, and X. Fu. How privacy leaks from bluetooth mouse? In *ACM CCS*, pages 1013–1015, 2012.

[90] A. Parate and et al. RisQ: recognizing smoking gestures with inertial sensors on a wristband. In *ACM MobiSys*, pages 149–161, 2014.

[91] L. R. Rabiner and B. Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1975.

[92] T. S. Ralston, G. L. Charvat, and J. E. Peabody. Real-time through-wall imaging using an ultrawideband multiple-input multiple-output (mimo) phased array radar system. In *IEEE International Symposium on Phased Array Systems and Technology*, pages 551–558, 2010.

[93] J. Raphael. Android wear on wi-fi: Using a smartwatch without a phone nearby. https://www.computerworld.com/article/2919013/android/android-wear-on-wi-fi-using-a-smartwatch-without-a-phone-nearby.html, 2018.

[94] 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*, 2014.

[95] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, 3(1):72–83, 1995.

[96] M. Ryan. Bluetooth: With low energy comes low security. In *USENIX WOOT*, pages 4–4, 2013.

[97] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann. Tracking human mobility using wifi signals. *PLOS ONE*, 10:1–11, 2015.

[98] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[99] D. Seamon. A way of seeing people and place: Phenomenology in environment-behavior research. *Theoretical Perspectives in Environment-Behavior Research*, pages 157–78, 2000.

[100] V. Sekara and S. Lehmann. The strength of friendship ties in proximity sensor data. *PLoS ONE*, 9:1–8, 2014.

[101] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. Spot localization using phy layer information. In *Proceedings of the 10th international conference on Mobile systems, applications, and services (ACM MobiSys)*, 2012.

[102] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti. Predicting user traits from a snapshot of apps installed on a smartphone. *SIGMOBILE Mob. Comput. Commun. Rev.*, 18:1–8, 2014.

[103] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *ACM MobiCom*, pages 39–50, 2013.

[104] M. Sherman and et al. User-generated free-form gestures for authentication: Security and memorability. In *ACM Mobisys*, pages 176–189, 2014.

[105] J. Shi, L. Meng, A. Striegel, C. Qiao, D. Koutsonikolas, and G. Challen. A walk on the client side: Monitoring enterprise wifi networks using smartphone channel scans. In *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM)*, 2016.

[106] N. Shokhirev. Hidden Markov Models. http://www.shokhirev.com/nikolai/abc/alg/hmm/hmm.html.

[107] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha. Beware, your hands reveal your secrets! In *Proceedings of the 2014 ACM Conference on Computer and Communications Security (ACM CCS)*, pages 904–917, 2014.

[108] D. Spill and A. Bittau. Bluesniff: Eve meets alice and bluetooth. In *USENIX WOOT*, pages 5:1–5:10, 2007.

[109] Statista. Number of connected wearable devices worldwide from 2016 to 2021, 2018. https://www.statista.com/statistics/487291/global-connected-wearable-devices/.

[110] I. Std. 802.11n-2009: Enhancements for higher throughput. Available at http://www.ieee802.org, 2009.

[111] K. Technologies. Keysight technologies 33509b. `https://www.alliedelec.com/keysight-technologies-33509b`, 2018.

[112] R. Togneri and D. Pullella. An overview of speaker identification: Accuracy and robustness issues. *IEEE circuits and systems magazine*, 11(2):23–61, 2011.

[113] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, pages 3–18. IEEE, 2017.

[114] D. Turcsany, A. Mouton, and T. P. Breckon. Improving feature-based object recognition for x-ray baggage security screening using primed visualwords. In *IEEE International Conference on Industrial Technology*, pages 1140–1145, 2013.

[115] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4052–4056. IEEE, 2014.

[116] S. Vhaduri and C. Poellabauer. Wearable device user authentication using physiological and behavioral metrics. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2017.

[117] C. Wang, X. Zheng, Y. Chen, and J. Yang. Locating rogue access point using fine-grained channel information. *IEEE Transactions on Mobile Computing*, 2016.

[118] H. Wang, T. T.-T. Lai, and R. Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *ACM MobiCom*, pages 155–166, 2015.

[119] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie. Human respiration detection with commodity wifi devices: do user location and body

orientation matter? In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 25–36. ACM, 2016.

[120] J. Wang, J. Xiong, X. Chen, H. Jiang, R. K. Balan, and D. Fang. Tagscan: Simultaneous target imaging and material identification with commodity rfid devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, pages 288–300, 2017.

[121] J. Wang, K. Zhao, X. Zhang, and C. Peng. Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization. In *ACM Mobysis*, pages 14–27, 2014.

[122] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1):3–19, 2013.

[123] X. Wang, Y. Wu, and W. Xu. Windcompass: Determine wind direction using smartphones. In *Sensing, Communication, and Networking (SECON), 2016 13th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[124] Y. Wang, Y. Chen, F. Ye, J. Yang, and H. Liu. Towards understanding the advertiser's perspective of smartphone user privacy. In *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 288–297, 2015.

[125] 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 *Proceedings of the 20th annual international conference on Mobile computing and networking (ACM MobiCom)*, pages 617–628, 2014.

[126] WeChat. Voiceprint, 2017. https://thenextweb.com/apps/2015/03/25/wechat-on-ios-now-lets-you-log-in-using-just-your-voice/.

[127] J. Wiese, J. I. Hong, and J. Zimmerman. Challenges and opportunities in data mining contact lists for inferring relationships. In *ACM UbiComp*, pages 643–647, 2014.

[128] D. Wu, D. Zhang, C. Xu, Y. Wang, and H. Wang. Widir: walking direction estimation using wireless signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 351–362. ACM, 2016.

[129] L. Wu, B. Brandt, X. Du, and B. Ji. Analysis of clickjacking attacks and an effective defense scheme for android devices. In *IEEE Conference on Communications and Network Security (IEEE CNS)*, pages 55–63, 2016.

[130] L. Wu, X. Du, and X. Fu. Security threats to mobile multimedia applications: Camera-based attacks on mobile phones. *IEEE Communications Magazine*, 52(3):80–87, 2014.

[131] Z. Wu, X. Xiao, E. S. Chng, and H. Li. Synthetic speech detection using temporal modulation feature. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7234–7238. IEEE, 2013.

[132] Z. Xu, K. Bai, and S. Zhu. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *ACM WISEC*, pages 113–124, 2012.

[133] H.-S. Yeo, G. Flamich, P. Schrempf, D. Harris-Birtill, and A. Quigley. Radarcat: Radar categorization for input & interaction. In *Proceedings of the 29th ACM Annual Symposium on User Interface Software and Technology*, pages 833–841, 2016.

[134] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117. ACM, 2017.

[135] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 301–315. ACM, 2015.

[136] L. Zhang, S. Tan, and J. Yang. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 57–71. ACM, 2017.

[137] L. Zhang, S. Tan, J. Yang, and Y. Chen. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1080–1091. ACM, 2016.

[138] T. Zhu, Q. Ma, S. Zhang, and Y. Liu. Context-free attacks using keyboard acoustic emanations. In *ACM CCS*, pages 453–464, 2014.

[139] Y. Zhu, Y. Zhu, Z. Zhang, B. Y. Zhao, and H. Zheng. 60ghz mobile imaging radar. In *Proceedings of the 16th ACM International Workshop on Mobile Computing Systems and Applications*, pages 75–80, 2015.