THE UNIVERSITY OF CHICAGO

MACHINE-AUGMENTED HUMANS AS A PRIVACY ARMOR

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY

YUXIN CHEN

CHICAGO, ILLINOIS
AUGUST 2022

This dissertation is dedicated to

my family and loved ones (Liting Wu, Quansheng Chen and Huiying Li),

my advisors Heather Zheng and Ben Zhao,

my committee member Pedro Lopes,

my friends, and SANDLab members.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Foremost, I would like to deeply thank my advisors Heather Zheng and Ben Zhao. It is their advice, inspirations, supports and patience that made all achievements in my PhD study possible. Their enthusiasm, dedication for research, the way they solve problems and their care of students have been teaching me a lot and will be my lifetime guidance. I especially thank Heather, who has always been helping me take those challenges arose during my research. After these years, I have become more confident to address future difficulties in work and life.

I would like to give special thanks to Pedro Lopes, who is my committee member and closest collaborator in various projects. He never ceased to advise, encourage, and inspire me for these years.

I am grateful to work with all my collaborators, Zhuolin Yang, Zhujun Xiao, Yanzi Zhu, Huiying Li, Shan-Yuan Teng, Zhijing Li, Zain Sarwar, Ruben Abbou, Steven Nagels, Max Liu. It would not be possible to accomplish this work without their creative thoughts and hard work.

I would also like to thank my senior colleagues Bolun Wang, Xinyi Zhang, Kevin Yao, Shiliang Tang for their wise advice. Thanks also to my lab mates, Jenna Cryan, Emily Wenger, Shawn Shan, Wenxin Ding, Christian Cianfarani, Arjun Bhagoji, Bimal Viswanath for all the great time we had together.

I would like to thank my family & friends for their unconditional support throughout my PhD study. I thank my parents Liting Wu and Quansheng Chen who have always been encouraging me. My special thanks go to my girlfriend Huiying Li. Without her support and care during my hardest time, I wouldn't have made this far.

ABSTRACT

As a variety of smart and connected sensors are being deployed everywhere, significant privacy issues arise since these devices can constantly capture our (private) behaviors in forms of image, video and sound. These sensor data can be used by adversaries to attack personal privacy. For example, leaked audio data can be processed using machine learning models to extract private conversation, track user activity, identify human speakers or even generate any speech in the voice of the speaker. These privacy attacks are fully automated and can be launched at scale. As a result, they pose a real security and privacy threat to everyone. Yet protecting users against such intrusive sensing is challenging. Privacy laws and policies can help regulate the use of sensors and machine learning models, but they are known to be difficult and slow to deploy.

In this dissertation, we explore personal privacy protection against intrusive sensing. We propose to develop low-cost wearables that users can carry and turn on/off to prevent their private information from being extracted by unauthorized parties. Along this line, we design and engineer novel wearables that protect both content and identity privacy. Our wearables also leverage the inherent properties of the human body to improve protection strength and coverage. Together, these wearables and the human body form a powerful privacy armor, providing users with full agency in privacy control.

This dissertation makes three key contributions.

First, to protect our speech privacy, we engineer a wearable microphone jammer as a bracelet, which disables surrounding microphones, including hidden ones. Our design leverages a hardware property that, when exposed to ultrasonic noise, commodity microphones will leak the noise into the audible range, which disrupts the speech recording. Our jamming bracelet also leverages natural body movements to increase protection coverage and effectiveness.

Second, we study typing content privacy, where we assess the vulnerability of wearable

keyboards to keystroke inference attacks. We show that typing using wearable keyboards can naturally defeat existing attacks because the keyboard and its layout are invisible in the physical world. We then develop a new, more sophisticated attack that can successfully infer wearable typing content using just a RGB camera. This presents a new threat against wearable typing privacy and the need for additional protection methods.

Finally, we also study identity privacy and its impact on user authentication. Since our standard biometrics data, such as face, voice, and fingerprint, can be easily captured by sensors and leaked to attackers, we develop an alternative, wearable-based authentication method based on muscle stimulation. Our proposed system authenticates a user by stimulating the user's forearm muscles with a sequence of electrical impulses (a challenge) and measuring the user's involuntary finger movements (response to the challenge). Our system produces 68 million challenges per user, using just one second of muscle stimulation. Attackers replaying used responses will be rejected, making our system highly robust against data breach and leakage.

In summary, this dissertation develops solutions to protect personal privacy against intrusive sensing, by augmenting the human body with wearables to form a ubiquitous privacy armor. We hope our work sheds light on the development of personal privacy protection in the physical world.

CHAPTER 1

INTRODUCTION

A variety of smart and connected sensors are populating into our physical world and revolutionizing the way we live, work and play. Today, our devices, homes, offices, private and public spaces are full of sensors, including cameras, microphones, biometric sensors, and many others.

However, as these sensors continue to flourish, significant privacy issues arise since they can constantly capture and save our (private) behaviors in forms of image, video and sound, either maliciously or by misconfiguration [86, 210, 184]. Using powerful machine learning (ML) models, adversaries can process these sensor data to extract our private information, and thus launch significant and unacceptable attacks against user privacy. Taking audio as an example. Leaked data can be processed by ML models to extract confidential conversation [210, 41, 40], track user activity [18], infer typed text and handwriting content [15, 232, 217], identify speakers [91] or even generate any speech in the voice of the speaker [90]. As these attacks are becoming fully automated, they pose a significant threat to all of us.

Despite significant concerns against intrusive sensing and privacy attacks, there are few tools available to protect users against them. Privacy laws and policies could help regulate the use of sensors and ML models, but they are known to be difficult and slow to deploy. A notable example is facial recognition. Despite the significant media backlash against intrusive facial recognition services like *Clearview.AI* [43], the legislative efforts to address these services remain elusive in the US [105, 8, 27].

Overview of My Work In this dissertation, we explore the idea of protecting personal privacy against intrusive sensing by low-cost wearables that users can carry and turn on/off to prevent their private information from being extracted by unauthorized parties. Our wearables also leverage the inherent properties of the human body to improve protection

strength and coverage. Together, these wearables and the human body form a powerful privacy armor, providing users with full agency in privacy control.

Along this line, we design and engineer novel wearables that protect both content and identity privacy. We start with engineering a wearable jammer to protect speech privacy by preventing unauthorized microphones from capturing and extracting our speech. We then focus on typing privacy, where we study the vulnerability of wearable keyboards to keystroke inference attacks. Finally, we propose a novel user authentication system using on-arm electrical muscle stimulation, which relies on disposable, privacy-insensitive muscle stimulation-response records rather than face/voice/fingerprint.

In the following, we briefly introduce the work included in this dissertation.

1.1 Wearable microphone jammers

Voice-based smart devices are everywhere and becoming an integral part of our life. Their implementation requires them to equip microphones that can always monitor and record our speech. Recent studies have shown that these devices can be exploited by adversaries to extract the content of our private speech. In the first part of my thesis, we seek to build a wearable that protects users' speech privacy.

To meet this goal, we engineer a wearable jammer that is worn as a bracelet. When turned on, it emits inaudible ultrasonic noise that disables microphones in the wearer's surroundings, including hidden microphones. Our design leverages a hardware property that, when exposed to ultrasonic noise, commodity microphones will leak the noise into the audible range, which disrupts the speech recordings. By building the ultrasonic jammer as a bracelet and arranging ultrasonic transducers in a ring layout, our wearable emits jamming signals in multiple directions. This eliminates the need for the wearer to manually point the jammer to a microphone. Our wearable also leverages natural body movements to increase protection coverage and effectiveness.

We confirm experimentally that our jammer is superior to existing stationary jammers by conducting a series of technical evaluations and a user study. The results demonstrate that (1) our wearable jammer largely outperforms existing static jammers in coverage; (2) it remains effective even if the microphones are hidden and covered by various materials, such as cloths or paper sheets; and, (3) our study participants feel that our wearable protects the privacy of their voice.

1.2 Typing privacy threats against wearable keyboards

Besides speech, keyboard typing is another modality we regularly use to produce content. Prior work on keystroke inference attacks shows that attackers can infer our typing content when they know the keyboard and its layout. On the other hand, when using wearable-based keyboards [138, 193, 187, 136, 139], the keyboard and its layout/size/position are known to the user using the wearable but remain hidden to any physical observers. This suggests that wearable keyboards could naturally protect our typing privacy.

The second part of my thesis seeks to understand the vulnerability of wearable keyboards to keystroke inference attacks, where the attacker has no knowledge of the user's keyboard location, size, or layout, but can only observe their finger/hand movements at a distance. Our research develops a new, more sophisticated attack that can successfully infer wearable typing content using just a RGB camera.

Our proposed attack uses a two-layer self-supervised system. In layer 1, noisy results of hand tracking on the keystroke video are used to detect keystrokes, followed by a language model to recognize keystrokes. These initial labels are filtered using multiple consistency checks to produce high confidence labels on video frames. Next, in layer 2, these labels and their corresponding video frames are used to train two 3D-CNN models that detect and recognize keystrokes from the raw video frames.

We evaluate this attack using IRB-approved user studies under a variety of conditions,

varying the target (user/typing behavior, content typed, physical environment) and attacker behaviors (hand tracking tool, attack distance). The attack is highly effective in nearly all settings, and performs well across our user study participants, despite significant different typing styles and abilities. This presents a new threat against wearable typing privacy and the need for additional protection methods.

1.3 On-arm electrical muscle stimulation for user authentication

Intrusive sensing also poses significant implications in biometric authentication. Today, when our biometric data (face/voice/fingerprint) are leaked to attackers, attackers can use them to bypass authentication systems used by banks and other critical services. There is nothing users can do to securely re-use their own data, as these biometrics are static.

To tackle this challenge, we explore a novel modality for active biometric authentication: electrical muscle stimulation (EMS). Our system, which we call ElectricAuth, stimulates the wearer's forearm muscles with an EMS challenge, i.e., a 1.2s sequence of electrical impulses and then measures the user's involuntary finger movements as a result of this challenge.

ElectricAuth authenticates users by leveraging what is typically seen as the biggest disadvantage of EMS: intersubject variability, i.e., the same electrical stimulation results in different movements in different users because everybody's physiology is different [101, 52, 58, 42, 140]. These differences arise from multiple compound factors in the field of muscle biomechanics and physiology [75, 112, 3]. All these differences add up to create individual responses to the same stimulus, which our system uses as the key feature to identify a user.

ElectricAuth also generates a very large pool of challenges by exploring an underutilized property of EMS: muscles respond differently depending on their current state of contraction, which can be altered by varying the timing between two impulses. Using four muscles, six impulses and seven time gaps, ElectricAuth encodes one of 68M possible challenges in 1.2s. As such, ElectricAuth is robust against data breaches and replay attacks because it never

reuses the same challenge twice in authentications – ElectricAuth rejects replay of recorded responses to any previously used challenges and can quickly recover from leak/breach of either authentication model or stored challenge-response pairs by asking the user to register responses to a new set of challenges (like registering new one-time passwords).

We evaluate our prototype of ElectricAuth by conducting a series of technical evaluations and user studies. The results demonstrate that: (1) ElectricAuth offers accurate user verification and resists three common biometric attacks: impersonation, replay and synthesis attacks; (2) ElectricAuth performs stably over 21 days against various muscle conditions (fatigue, humidity, etc.); (3) ElectricAuth can verify the user in 3ms on laptop's CPU and 35ms on a small embedded device after receiving a response, and can use either IMUs or a depth camera to track finger movements.

1.4 Structure of this dissertation

This dissertation is organized into five chapters. After this introduction (Chapter 1), we present a wearable microphone jammer that protects speech content privacy by disabling microphones in the wearer's surroundings (Chapter 2). We then focus on keyboard typing, where we study privacy threats against wearable keyboards. In the following chapter, we also study identity privacy and its impact on user authentication, where we propose a novel active biometric authentication system that authenticates a user by stimulating the user's forearm muscles with a sequence of electrical impulses and measuring the user's involuntary finger movements (Chapter 4). Finally, we conclude this dissertation by summarizing our contributions, and discussing the insights learned as well as potential future directions for personal privacy protection in the physical world (Chapter 5).

CHAPTER 2

WEARABLE MICROPHONE JAMMERS

Voice-based smart devices are populating into our world. Taking North America as an example, the number of these devices has more than doubled the population. Despite the significant benefits these devices have brought to us, these devices can also be exploited by attackers to monitor and record our private conversation. In this chapter, we describe how we build wearables to protect our *speech privacy*.

2.1 Introduction

Despite the initial excitement around voice-based smart devices, consumers are becoming increasingly nervous with the fact that these interactive devices are, by default, *always* listening, recording, and possibly saving sensitive personal information [125, 209, 143, 86]. Take digital voice assistants, which are featured in most smartphones, smartwatches, and smart speakers, as an example. From the outside, these interactive assistants appear to only respond to designated wake-up words (e.g., "Alexa" and "Hey Google"). However, their implementation requires them to listen continuously to detect these wake-up words. It has been shown that these devices can monitor and record sounds and conversations in real time, either maliciously [210], by misconfiguration [86], or after compromise by attackers [184]. Leaked audio data can be processed to extract confidential information [210, 41, 40], track user activity [18], count human speakers [212], or even extract handwriting content [217]. These negative implications on users' security and privacy are significant and unacceptable. To make matters worse, many other acoustic attacks (e.g., turning speakers into microphones [74], inferring the content of a printed page by recording its printing [20], inferring a 3D object's geometry by recording its printing [65], inferring typed text by listening to key presses [15, 232]) as well as many forms of espionage (e.g., industrial espionage [45, 95]) rely

on eavesdropping via hidden microphones.

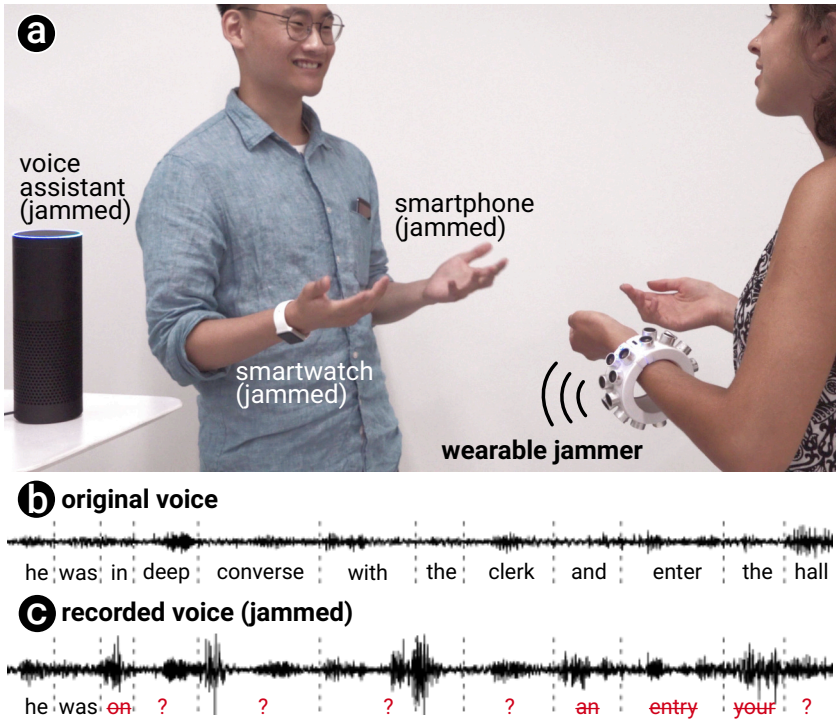

Figure 2.1: (a) We engineered a wearable ultrasound jammer that can prevent surrounding microphones from eavesdropping on a conversation. (b) This is the actual speech that the conversation partner hears, since our jammer does not disrupt human hearing. However, (c) is the transcript of what a state-of-the-art speech recognizer makes out of the jammed conversation.

Therefore, it is critical to build tools that protect users against the potential compromise or misuse of microphones in the age of voice-based smart devices. Recently, researchers have shown that ultrasonic transducers can prevent commodity microphones from recording human speech [170]. While these ultrasonic signals are imperceptible to human ears, they leak into the audible spectrum after being captured by the microphones, producing a jamming signal *inside* the microphone circuit that jams (disrupts) voice recordings. The leakage is caused by an inherent, nonlinear property of microphone's hardware. Not only

have researchers built prototypes using ultrasonic speakers [170], but also these jammers are currently commercially available to the public. However, all these devices exhibit two key limitations: (1) They are heavily directional, thus requiring users to point the jammer precisely at the location where the microphones are. This is not only impractical, as it interferes with the users' primary task, but is also often impossible when microphones are hidden. (2) They rely on multiple transducers that enlarge their jamming coverage but introduce *blind spots*—locations were the signals from two or more transducers cancel each other out. Such blind spots occur especially in close proximity to the jammer; in fact, 17% of all locations within 1.2m of a typical multi-transducer jammer are blind spots. If a microphone is placed in any of these locations it will not be jammed, rendering the whole jammer obsolete.

To tackle these shortcomings, we engineered a wearable jammer that is worn as a bracelet, which is depicted in Figure 2.1. By turning an ultrasonic jammer into a bracelet, our device leverages natural hand gestures that occur while speaking, gesturing or moving around to blur out the aforementioned blind spots. Furthermore, by arranging the transducers in a ring layout, our wearable jams in multiple directions and protects the privacy of its user's voice, anywhere and anytime, without requiring its user to manually point the jammer to the eavesdropping microphones.

We confirmed that an ultrasonic microphone jammer is superior to state-of-the-art and commercial stationary jammers by conducting a series of technical evaluations and a user study. These demonstrated that: (1) our wearable jammer outperformed static jammers in jamming coverage; (2) its jamming is effective even if the microphones are hidden and covered by various materials, such as cloths or paper sheets; and, (3) in a life-like situation our study participants felt that our wearable protected the privacy of their voice.

2.2 Background and related work

Our work builds on top of ultrasonic emitters and wearables. Also, we discuss the implications of data leaks in interactive devices, especially those with microphones and cameras. Lastly, we introduce the underlying ultrasonic jamming principle that our device is based on.

2.2.1 Privacy issues with interactive devices

As various interactive devices being deployed into daily life, privacy issues arise since these devices often rely on constant capturing of multimedia, such as photos, videos, or sound, in order to provide services that assist users' activities [26, 154].

Researchers have proposed privacy-aware methods to collect user's data by, for instance, designing improved notifications [141] or exploring configurations that are privacy-conscientious [5]. These approaches are, however, developer-centric and thus require that the user trusts the interactive system. The result is that these approaches are beneficial but not a fail-proof solution, as devices are still exposed to attackers. Lastly, these solutions do not seek to empower the users to actively protect their privacy.

For example, as the privacy implications of cameras grew in importance, webcams started to use lights that indicate their recording state [161]. However, these indicators can be disabled by attackers [30], which led to many users opting for physically covering up the webcams [124].

More recently, digital assistant devices, such as Amazon Echo, have become very popular due to their interactive (conversational) ability. These interactive devices are built with a microphone and a speaker. To interact with the user when needed, these voice assistants are designed to respond to designated wake-up words (e.g., "Alexa" and "Hey Google"). However, continuously listening is required to detect these wake-up words—this has resulted in many worldwide security breaches, where it was found that these devices leaked or saved

sensitive personal information from their users [125, 209, 143, 86]. It was shown that these devices can monitor and record all voices, sounds and conversations in real time, either maliciously [210], by misconfiguration [86], or after compromise by attackers [184]. The leaked audio can be further processed to extract confidential information [210, 41, 40], track user activity [18], count human speakers [212], and so forth. One sane option is certainly to turning these devices off one by one. Unfortunately, that still leaves eavesdropping devices that the user cannot control or that the user is simply not aware of. Instead of turning off all the devices manually, microphone jammers aim at empowering users with a tool to disrupt (jam) voice recordings whenever and wherever they want, providing a physical layer of privacy on demand.

2.2.2 Principles of ultrasonic microphone jamming

Recent work has demonstrated the feasibility of using ultrasonic transducers to disable nearby microphones [170]. The advantage of jamming by means of ultrasound is that it is "silent" to users, as ultrasound is inaudible to humans. We illustrate this type of jamming in Figure 2.2. Ultrasonic jamming is possible because these higher-frequency signals, after being captured by the microphone's non-linear diaphragm and power-amplifier, will create a lower-frequency "shadow" that happens to be in the microphone's filtering range—the audible range [170]. This technique works against billions of commodity microphones (found in phones, laptops, voice assistants, etc.), without any microphone modification. The fundamental exploit is due to the fact that acoustic amplifiers are only linear around the audible frequency range, while outside of the range (e.g., ultrasound), the amplifier's response exhibits non-linearities [170, 2]. This leakage from ultrasound to audible range adds so much audible noise on the microphone circuitry that it effectively renders voice recordings unusable.

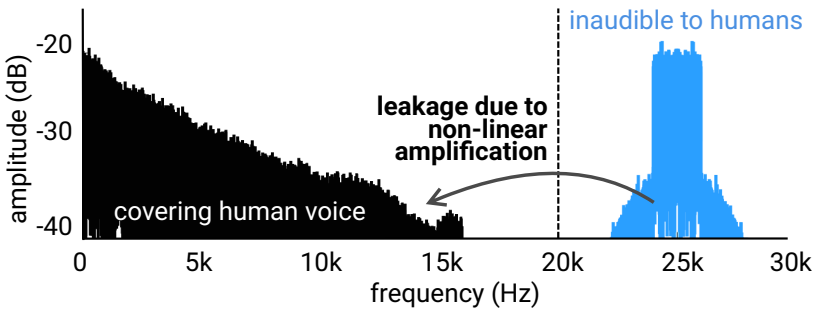

Figure 2.2: Working principle behind ultrasonic jamming (similar to [170] but here at the example of a 25kHz signal). Here, we depict how an ultrasonic jamming signal (shown in blue), which is inaudible to humans, still leaks into the recorded speech due to non-linear amplification of the microphone's circuit. The result is that the leaked signal covers up precisely the spectrum in which a user's voice is recorded (shown in black).

2.2.3 Leveraging microphone non-linearity

This non-linearity in microphone circuitry was originally discovered by musicians and leveraged for sound synthesis [97]. Only more recently, have researchers leveraged these non-linearities as a potential tool for setting up hidden communication channels, disabling microphones, or as an adversarial avenue for injecting hidden voice commands. A series of projects leveraged this property to attack digital voice assistants [224, 184, 171]. Here, an adversary can play (arbitrary) voice commands modulated in the ultrasonic range to digital assistants and force these devices to decode them as normal voice commands. Since the original ultrasonic command is inaudible, the attacker can successfully issue commands without being detected (i.e., heard) by nearby users.

Similarly, *backdoor* [170] leverages non-linearity to build an inaudible communication channel among devices and to jam microphones. The *backdoor* jams based on either amplitude modulation (AM) or frequency modulation (FM). *backdoor* was tested in a limited set of experiments with the jammer pointing to a single microphone. Nowadays, there are commercial ultrasonic jammers, such as the *i4*. Unfortunately, although all of them are large,

bulky (0.38kg–5kg), and pricey ($799–$6900) [54, 50, 148, 88]. These jammers have also a limited angular coverage and require the users to point directly at the microphone. This is disadvantageous in that these jammers require user's attention to operate and cannot be used against hidden microphones. Inspired by these devices, we propose a novel approach that, instead, leverages the advantages of a wearable design to enhance jamming effectiveness.

2.2.4 Wearable devices based on ultrasound

Researchers have used signals in ultrasonic bands [17, 134] and near-ultrasonic bands (e.g., 18.8kHz) [39, 73] to enable interaction with/among devices. As an example, Gupta et al., utilize Doppler shifts in emitted ultrasound to enable a laptop to perform gesture tracking [73]. A variety of smartphone applications use ultrasonic signals as beacons to perform device localization and tracking [13, 197, 68], again based on the aforementioned leakage to the audible band.

2.3 A wearable jammer bracelet

We engineered a microphone jammer in a wearable form factor, which effectively jams in more directions around the user than existing approaches. To assist the reader in replicating our device, we describe the implementation details and the key design elements that enabled our wearable jammer to outperform existing jammers.

We designed our wearable jammer as a bracelet so that it can be easily activated [222, 205, 49, 153, 9] whenever the user decides to engage in a private conversation. Having the device at users' reach at all times provides them with "always available input" [175], ensuring the user is the one in control.

2.3.1 Implementation

To help readers replicate our design, we now provide the necessary technical details. Furthermore, to accelerate replication, we provide all the source code, firmware, and schematics of our implementation [1].

Our prototype, which is depicted in Figure 2.3, is a self-contained wearable device comprised of the following components: a 3D-printed 9cm ring (outer diameter) with a slit that acts as a hinge, allowing the wearer to open up the bracelet and fit it around their arm; 23 ultrasound transducers (NU25C16T-1, 25kHz), featuring 12 on the lower ring and 11 on the top ring (one transducer was removed to make space for the aforementioned hinge); a low-power signal generator (AD9833, up to 12.5MHz with 0.004Hz programmable steps); an ATMEGA32U4 microprocessor; an LED status indicator; a tactile switch (not shown); a LiPo battery (3.7V, 500mAh); a 3W audio amplifier (PAM8403), and, a 3.7V to 5V step-up regulator. Our microprocessor controls the signal generator via Serial Peripheral Interface.

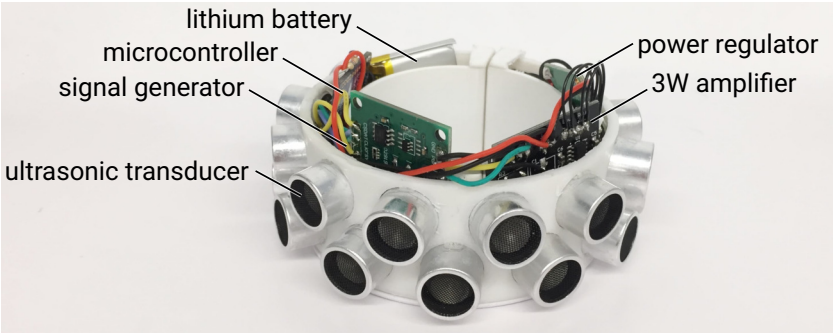

Figure 2.3: Our prototype is a self-contained wearable comprised of ultrasonic transducers, a signal generator, a microcontroller, a battery, a voltage regulator and a 3W amplifier.

1. `http://sandlab.cs.uchicago.edu/jammer`

Signal generation

We generate our ultrasound jamming signal via the AD9833 sine wave generator. This integrated circuit (IC) produces a pure sine wave at a desired frequency up to 12.5MHz. To select our sine wave's frequency, we control the AD9833 using our microcontroller via SPI[2]. In order to jam effectively, we produce not only one frequency but a range of frequencies. According to the principles of ultrasonic jamming, each of these will produce a shadow at an audible frequency; therefore, using multiple frequencies enhances jamming. We implement our signal by sweeping the frequency of the sine wave randomly between 24kHz to 26kHz (i.e., 25kHz±1kHz) in steps of 1Hz. Our sine wave frequency changes every 0.45 ms. In our earlier designs, we employed a 92kHz wave player IC that played back the white noise (25kHz±1kHz). However, we found that via empirical testing that our randomly-sweeping sine wave yielded the same jamming power with significantly less power consumption than the overly complex wave-player IC. Lastly, we amplify our signal using a 3W amplifier (PAM8403). Note that we set the amplifier to operate below maximum amplification. This reduces our power consumption and preserves signal quality (low distortion). When measured directly at any of the transducers, the loudness of our device is around 92.3dBA.

Wearable characteristics: power and weight

We measured the energy consumption of our prototype bracelet. It consumes approximately 0.47W (3.7V × 127mA) when jamming, which is ten times less energy than that used by the commercially available *i4* jammer. Thus it can continuously jam for around four hours on our 500mA battery. Furthermore, our device and battery weigh 135 grams.

2. `https://github.com/Billwilliams1952/AD9833-Library-Arduino`

2.3.2 Key design elements

Our device was designed with three key elements that allow it to outperform state-of-the-art microphone jammers.

1. Multi-directional jamming using a ring layout. Existing microphone jammers, such as *backdoor* and *i4*, embed their ultrasonic transducers in a flat (1D or 2D) layout. As a result, these jammers are effective *only* when the user points them to the target microphone. This is disadvantageous as: (1) it requires the user to steer the device, making the jamming action a primary task; and, (2) it is practically impossible to use against hidden microphones. Instead, our prototype features all its ultrasonic transducers in a ring layout, effectively enabling jamming in *multiple directions* on a plane. We will later demonstrate that our design is superior by means of both simulations and experimental evaluations.

2. Reducing blind spots by leveraging naturally-occurring movements. A significant benefit of proposing a microphone jammer as a wearable device is that we can mitigate the traditional blind spot problem, which affects all transducer arrays, by leveraging naturally-occurring movements. While a user is wearing our jammer, the device is, most of the times, being moved as the user walks, gestures, points, types, etc. It is precisely these movements that we leverage to reduce blind spots, because as the device moves in space the signal emission map moves accordingly and creates new areas of increased signal strength that blur out the blind spot areas.

3. Collocation with the user's voice. The last design element that makes a wearable design superior is its ubiquitousness. A wearable jammer is collocated with the user that it protects, whereas stationary jammers need to be installed or moved around in every space the user inhabits. Furthermore, the short distance between the jammer and the speaker's mouth prevents the use of beamforming microphone arrays to separate the signals of the human speaker and the jammer [10], making the wearable jammer a stronger defense.

2.4 Overview of experiments and study

In order to validate that wearable microphone jammers outperform existing approaches, we conducted **simulations** and three **experimental evaluations**. Lastly, to understand how participants perceive the effectiveness of our wearable jammer, we conducted a **user study**. To aid the reader in understanding the different validations we performed, we present an overview of our simulations, experiments and study:

1. Simulating jammer layouts. Prior to designing our jammer, we confirmed by means of simulation that a wearable bracelet with a ring-layout reduces blind spots when compared to stationary jammers with planar-layouts. To do so, we simulated the power of an ultrasonic signal in space after it leaves the transducer. With our simulations we found that (1) jammers with transducers in a planar layout jam mostly in one direction; (2) on the contrary, positioning the transducers in a ring layout increases jamming in multiple directions; and, (3) adding small (simulated) movement, which occurs naturally in a wearable device, results in a blind spot reduction, similar to what can be achieved using more complex control techniques with multi-frequency signals. This finding is critical because: (1) it allows us to keep the device's design and circuit simple (i.e., using a single signal source), which reduces power consumption, making it compatible with a wearable form factor; (2) our approach does not sacrifice jamming quality when compared to a more complex and hardware heavy approach (i.e., using multiple signal sources). These findings informed how we created our prototype, which we used in all subsequent experiments and user study.

2. Experiment#1: angular power distribution. We measured the angular power distribution of our wearable jammer and both existing devices (a planar jammer with 9 transducers and the commercially available *i4*). We found that our device provides a wide-spread angular coverage ($M = -3.3dBA$, $SD = 1.6dBA$), while the existing jammers are highly directional (planar jammer: $M = -19.2dBA$, $SD = 8.5dBA$; *i4*: $M = -17.0dBA$, $SD = 6.8dBA$).

3. Experiment#2: jamming speech recognizers. We measured how effectively our wearable device jams speech recognizers at different angles, when compared to a planar jammer and *i4*. We found that our wearable device jams more effectively in multiple directions with an increased word error rate (WER) when compared to the other jammers (our wearable: $M = 96.59\%$ WER, $SD = 3.97\%$; planar jammer: $M = 38.89\%$ WER, $SD = 21.72\%$; *i4*: $M = 57.55\%$ WER, $SD = 35.04\%$).

4. Experiment#3: jamming microphones covered by everyday materials. We evaluated how our wearable jams microphones that are covered with everyday materials (i.e., hidden microphones inside boxes, behind clothes, etc.); this stems from a unique feature of our device as it does not require pointing to the target microphone. We found that our device jams microphones hidden under a variety of objects, such as ordinary cloths, foam-based microphone windshields or paper sheets, with a word error rate above 97%.

5. User study. Lastly, we evaluated whether wearing our jamming bracelet impacted participants' perception of privacy. In our study, we asked groups of participants to engage in life-like conversations while they wore the bracelet one at a time. We found that participants felt our wearable protects their privacy (M=5.4 out of 7, SD=1.1).

2.5 Simulating jammer layouts

Prior to designing our jamming bracelet, we explored, via simulations, whether a wearable would be beneficial. We were interested in answering three questions: (1) how directional are jammers based on planar transducer layouts (e.g., *i4*)?; (2) how do blind spots affect a jammer with its transducers in a ring layout?; and, (3) how do the blind spots behave with respect to small movements of the jammer? All the following simulations were conducted using Matlab. For researchers interesting in replicating our simulations we provide their source code[3].

3. `http://sandlab.cs.uchicago.edu/jammer`

2.5.1 Simulation parameters

Generally speaking, our simulation computed the propagation of ultrasound from our sources to all points around the device. To model the directivity of our transducers, we utilized the piston model [142, 132] as a good approximation to the pattern supplied by the manufacturer's datasheet[4]. Our transducers are designed to operate at a central frequency of 25kHz, and our control technique sweeps the frequency of a sine wave randomly between 24kHz to 26kHz in steps of 1Hz, every 0.45ms. To simulate multiple signal sources, different random seeds are used in the generation of random frequency sweeping of each source. To simulate a planar jammer, we took the 3×3 array design by [170, 171], which features 9 transducers in a 3 × 3 planar grid. For the ring-layout, the transducers were placed in a diameter of 11cm. Our simulation runs on a 96kHz, i.e., larger than the Nyquist rate for 25kHz. Lastly, our simulation does not account for reflections.

2.5.2 Simulation algorithm

Our simulation algorithm is based of Morales et al. [142] and Marzo et al. [132]. Let S be the transducers in a jammer, with each transducer $s \in S$. Transducers are modeled as a piston source of radius $r = 8.2mm$. Let T be the time sampled in the simulation, with each time step $t \in T$. $\mathbf{P}_{ref}$ represents the transducers reference pressure; k is the wavenumber ($k = \omega/c_0$); $\mathbf{d}(p, p_s)$ is the distance between the transducer and the point; θ is the angle between the transducer's normal and the point; J_1 represents a Bessel function of the first kind, and $f_s(t)$ represents the signal transmitted by s at time t.

Given our transducer's model, the complex acoustic pressure $\mathbf{P}_{s,t}(p)$ contributed by each

4. Ultrasonic transducer (NU25C16T-1), Jinci Technology.
`http://www.jinci.cn/showgoods/736.html`

transducer s at a given position p and time t is computed as:

$$\mathbf{P}_{s,t}(p) = \frac{\mathbf{P}_{ref}}{\mathbf{d}(p, p_s)} \cdot \frac{2 \cdot J_1(k \cdot r \cdot sin\theta)}{k \cdot r \cdot sin\theta} \cdot \mathbf{f}_s(t - \frac{\mathbf{d}(p, p_s)}{c_0}) \tag{2.1}$$

The total far field generated by all the transducers at time t can be computed as the summation of the contribution of each individual transducer $\mathbf{P}_t(p) = \sum_{s \in S} \mathbf{P}_{t,s}(p)$. And the average far field generated over time can be computed as the root mean square of the contribution of each time step $\overline{\mathbf{P}(p)} = \sqrt{\frac{1}{|T|} \cdot \sum_{t \in T} \mathbf{P}_t(p)^2}$.

We simulated a total of 0.4s (roughly the average duration of a human spoken word [24]) with 13.573ms time gaps in between each sample, up to 1-meter radius around the jammer. To simulate a moving jammer, we update the position and orientation of each transducer at each sampled time step and simply repeat the aforementioned process. To simulate a small movement, we rotated all transducers by 15 degrees in 400ms – this depicts a relatively small microgesture of the wrist turning right.

2.5.3 Results

We performed four 3D simulations that suggested that a wearable jammer might outperform existing, planar or stationary, jammers. These are all depicted in Figure 2.4. For the sake of visual clarity, we plot only a 90° range of a 2D cross-section of the power distribution centered around the jammers.

Simulating planar jammers

Figure 2.4(a) shows a simulation of a planar jammer; this is the design used in all known microphone jammers. We observed a rather limited angle coverage around the jammer, suggesting that planar jammers are mostly directional. From this insight, we decided to explore non-planar layouts.

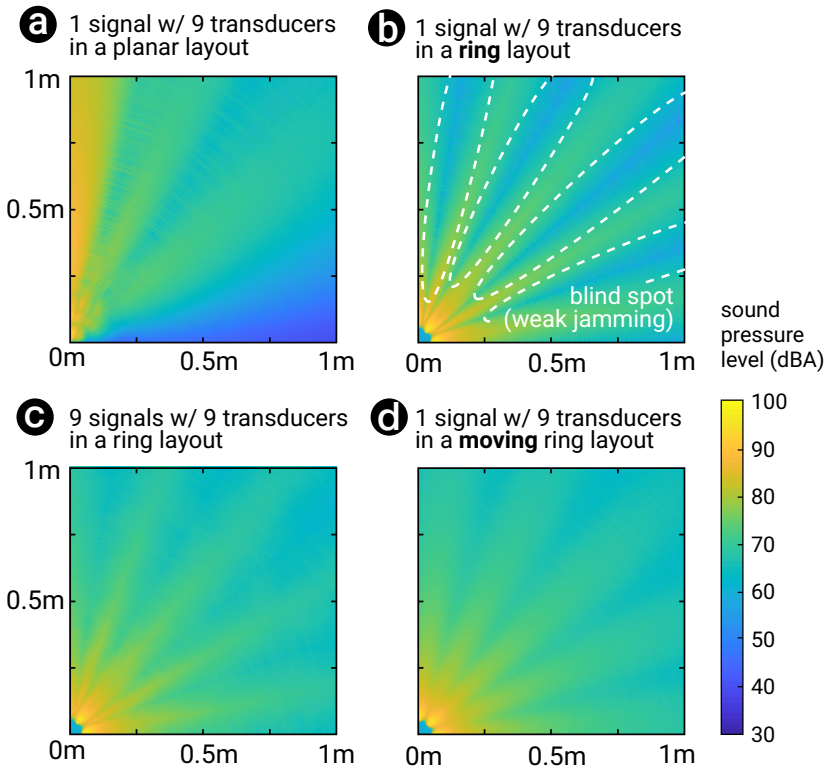

Figure 2.4: Our simulations depict how different transducer layouts radiate around the simulated device. We found that, when moving in space, a wearable jammer outperforms stationary jammers.

Simulating ring-layouts

We simulated a ring-layout with 9 transducers. The result is depicted in Figure 2.4(b). We observed that, when compared to planar layouts, it radiates in all directions, with stronger components in the horizontal plane aligned with the transducers. However, we also observed the appearance of the blind spots between transducer pairs (zones where their signals cancel each other out). These well-known blind spots are a key disadvantage of any multi-transducer jammer [126]; a microphone placed within a blind spot is unlikely to be jammed since the jamming signal intensity is weak.

One way to mitigate blind spots is to utilize a large number of out-of-phase sources. For instance, if one scales up to 9 independent signal generators it would limit phase collisions

and thus reduce blind spots. We simulated this configuration and depicted it in Figure 2.4(c). We observed a smooth radiation pattern around the center—ideal for jamming. However, this approach drastically increases the number of components required to manufacture this design, e.g., 9 signal generators and 9 amplifiers (one per transducer). This approach is thus, highly impractical for a wearable implementation, both in its hardware footprint and its power consumption.

Blurring blind spots via movement

Thus, the ideal wearable implementation would find a way to mitigate the blind spots using only one signal source and one amplifier. Figure 2.4(d) demonstrates the power of making a jammer *into a wearable*. A wearable will move in space alongside the user's body. To simulate movement, we turned the jammer by 15 degrees during the 400ms of the simulation—as would occur when the user's wrist would turn to the right slightly. The result, depicted in Figure 2.4(d), is a smooth radiation map, containing almost no blind spots. We took this as the blueprint for our wearable jammer implementation. In the following laboratory experiments, we will empirically confirm these simulation results.

2.6 Experiment#1: angular power distribution

In this experiment, we measured the angular power distribution (i.e., the power emitted at different angles) of our wearable device in comparison to our aforementioned planar 3×3 jammer and the commercially available *i4*.

2.6.1 Experimental setup

We utilized three jammers in this study: (1) The *i4* (from *Amazon.com*, $799) consists of two perpendicular rows of ultrasonic transducers, five transducers on the side and two on the top.

From our spectral analysis, the *i4* operates at the low end of ultrasonic frequency (20-24kHz), which allows its signals to travel further with slightly less power drop but unfortunately produces some disturbing audible sounds, likely due to signal leakage in its transducers resulting from the 20kHz signals. This device weighs 380 grams and consumes 4.2W of power. When measured directly at the transducers (with a sound pressure meter), its loudness is around 92.4dBA. (2) The planar jammer is an array of nine ultrasonic transducers in a 3 $\times$ 3 configuration. We built this jammer following [170, 171]; this device uses precisely the same transducers and amplifier as ours. The planar jammer used in this study operates at 25kHz$\pm$1kHz (the same signal as our device) and is completely inaudible. Similarly to [170, 171], this is not a stand-alone device and its power supply and circuitry are not integrated. When measured directly at the transducers, its loudness is around 92.6dBA. (3) Our wearable jammer was animated by a simple mechanical contraption. To move our bracelet, we used a servo motor. We programmed the servo to move 15° in 400ms, which is similar to slight wrist twist if the device was worn by a user. When measured directly at the transducers, our device's loudness is around 92.3dBA.

To measure the angular power distribution of all three devices, we placed the jammers on a table, one at a time. We measured all angles from 0° to 180° around the jammers at a distance of one meter, in steps of 5°. To obtain an accurate power measurement, we utilized the HT-80A sound level meter, which includes a well-calibrated microphone. When measuring our moving wearable, we took the average of the minimum and maximum power measured at each angle.

2.6.2 Results

The angular power distribution measured for our wearable jammer, planar jammer and *i4* are shown in Figure 2.5. We found that our device provides a wide-spread angular coverage ($M = -3.3dBA$, $SD = 1.6dBA$), while the existing jammers are highly directional (planar

jammer: $M = -19.2dBA$, $SD = 8.5dBA$; *i4*: $M = -17.0dBA$, $SD = 6.8dBA$).

Furthermore, in the case of a planar jammer or the *i4*, even within the angular sector of [0°,40°], a subtle angle change of 2° leads to a 5-10dBA drop in their jamming power. This uneven distribution is due to the aforementioned blind spot problem [126]. Instead, the power of our wearable jammer has no dramatic drops across all angles, as the movement helps to blur out the blind spots.

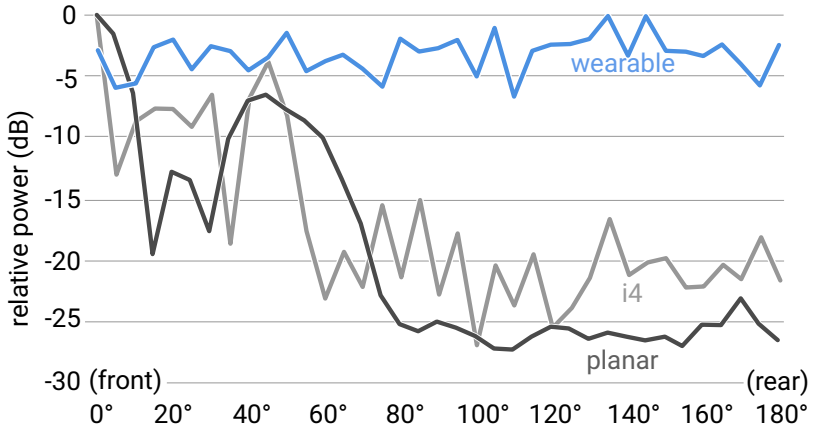

Figure 2.5: Real-world measurements of the jammer's angular coverage, in terms of the signal power as the jammer-to-microphone angle α increases from 0° to 180°, normalized by the maximum power of each jammer. The distance between the jammer and the microphone is kept at 1m. Angular coverage of the wearable jammer under movement. Jammer is 1m away from microphone.

2.7 Experiment#2: jamming speech recognizers

For an end-to-end evaluation of jamming effectiveness, we measured the ability of state-of-the-art speech recognizers to extract text from recordings of microphones jammed with our wearable or the baseline devices.

2.7.1 Experimental setup

We tested the jamming effectiveness of three jammers: our wearable device (animated by the same apparatus as in the previous experiment) and two baseline devices (the planar jammer and *i4*). We tested the jamming at multiple angles from 0° to 180°, in steps of 10° and always 1 meter away from our jammer. This experiment used the built-in microphones of a Nexus 6 and a Xiaomi Mi 6. For the sake of visual clarity, we depict only the most conservative result, i.e., the device that best evaded our jamming–the Nexus 6.

To create a comparable experiment across multiple devices and angles, we cannot rely on a human speaker. Even a trained public speaker that would not make any pronunciation mistakes, would still introduce confounding variables in our measurements as their voice would not be perfectly replicable across multiple trials, i.e., its loudness (dBAs), its direction, its timbre, and so forth. Therefore, we utilized pre-recorded speech and played it back using a speaker (JBL GO, frequency response from 180Hz-20kHz). Our speaker was calibrated so as to play the pre-recorded human speech at a standard sound level of human conversation (55-66dBA measured at 1m away according to [149]). Lastly, the recorded speeches used in our experiment were ten 1-minute long sentences taken, at random, from the LibriSpeech dataset [48], which is commonly used by speech recognition researchers.

For each trial, we played back the pre-recorded speech via the speaker and recorded it with the smartphone's microphone. Then, we fed these recordings into the IBM Speech to Text [196]—a popular speech recognizer.

To compute the effectiveness of a jammer, we take the output of the recognizer and compare it to the transcript of each sentence in the dataset (ground truth). This results in the percentage of the words that were incorrectly transcribed by the text-to-speech; this is denoted as Word Error Rate (WER) and is a common metric in speech processing.

2.7.2 Results

Our results are depicted in Figure 2.6. We found that our wearable device jams more effectively in all directions (M = 96.59% WER, SD = 3.97%) than the existing devices (planar jammer: M = 38.89% WER, SD = 21.72%; *i4*: M = 57.55% WER, SD = 35.04%). Since we did not measure much difference between the measurements obtained from the two different smartphones, our results depict an average of both. Furthermore, note that even without jamming, no text-to-speech system is perfect. In our experiment, we measured that in the absence of jamming the IBM Speech to Text had a WER around 30% for the smartphone.

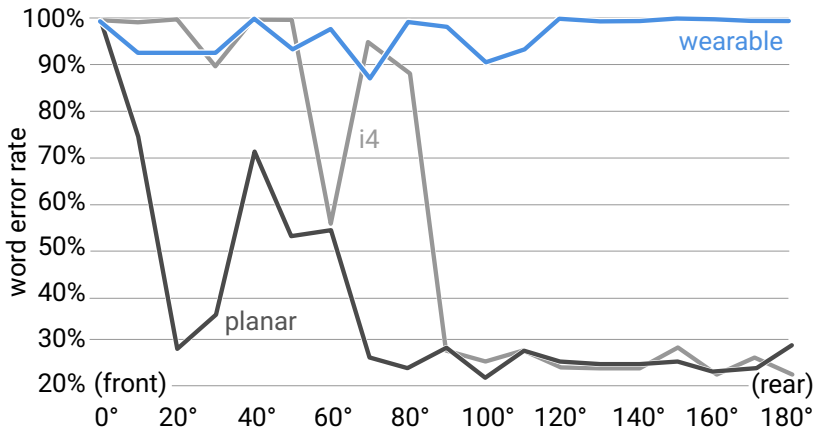

Figure 2.6: Word error rate (WER) of speech recognition for jamming with our wearable, planar jammer and *i4*. We found that the WER for planar and *i4* dropped drastically after 90°, while our wearable maintained a constant jamming effect >87%.

Moreover, we observed a similar pattern to the angular power distribution found in the previous experiment. As depicted in Figure 2.6, both the planar jammer and *i4* exhibit WER drops at 30°and 60°, around their blind spots. On the contrary, our wearable jammer maintained a high WER throughout the measured angles. Furthermore, we observed a severe drop in WER, for planar jammer and *i4*, when the microphone was placed more

than 90°away from the jammer (planar jammer: $M = 26.30\%$, $SD = 2.16\%$; *i4*: $M = 26.14\%$, $SD = 2.07\%$; our wearable: $M = 97.92\%$, $SD = 3.40\%$). First, this confirms that existing jamming approaches are highly directional. Secondly, it confirms that our approach is effective even when not pointing directly at the target device.

To exemplify the effectiveness of jamming with our wearable, we depict in Figure 2.7 three short sentences from our dataset. By contrasting the output of the text-to-speech when fed the jammed recording vs. when fed the clean recording, we observed that most words became unrecognizable. Yet, some words slipped through and were recognized, such as "space".

jammer off	**jammer on**
"now to bed boy"	"it"
"it is late and I go myself within a short space"	"space"
"most of all robin thought of his father what would he council"	

Figure 2.7: Examples of recognized sentences in clean speech case with perfect recognition and jamming case with our wearable jammer (WER 98.6%). Blank indicates nothing was recognized.

2.8 Experiment#3: jamming microphones covered by everyday materials

As we observed in our last experiment, our wearable jammer has a wide angular coverage. Thus, it affords jamming even without the user needing to point to the target microphone. This feature allows it to also jam hidden microphones that the user might not be aware of. In this experiment, we evaluated whether this type of ultrasonic jamming is effective when

the microphone is covered with a variety of materials, as it would be typical of a hidden microphone (e.g., in industrial espionage [45, 95]).

2.8.1 Experimental setup

We repeated our previous experiment (same apparatus), except we this time covered the microphones with different materials. In particular: a plastic bag (0.2mm thick Polyethylene), a plastic box (1mm thick Polypropylene), a paper sheet (from a 20lb set), a paper tissue (3-ply tissue), a cardboard box (3mm thick), a cloth (i.e., a cotton T-shirt), and two windshields (one fur and one foam) typically used in professional audio recordings. Additionally, we also recorded a baseline with no blockage applied.

2.8.2 Results

The results of the average WER are depicted in Figure 2.8. We found that the paper tissue, paper sheet, foam windshield and cloth had little impact on jamming performance, resulting in an WER of 99%; in other words, our jammer was able to jam microphones hidden by these materials and only 1% of the words were correctly transcribed by the text-to-speech recognizer. Conversely, in the absence of our jammer, the text-to-speech recognizer recovered more than 60% of the words.

On the other hand, if the microphone was covered by a plastic box or cardboard box, we observed that the jamming performance dropped considerably to WER 41.01% and 46.76%, respectively; in other words, our device did not jam through the plastic box nor the cardboard box. In these two conditions, we also observed an increase of the WER even in the absence of jamming up to 38.13% and 40.29% respectively. To sum it up, materials such as paper, cloth and foam have little impact on jamming performance, while thicker or more complex blockage materials (e.g., plastic box, fur windshield, etc) decrease the jamming performance. This result is not a limitation of our wearable design but a limitation of acoustic jamming in

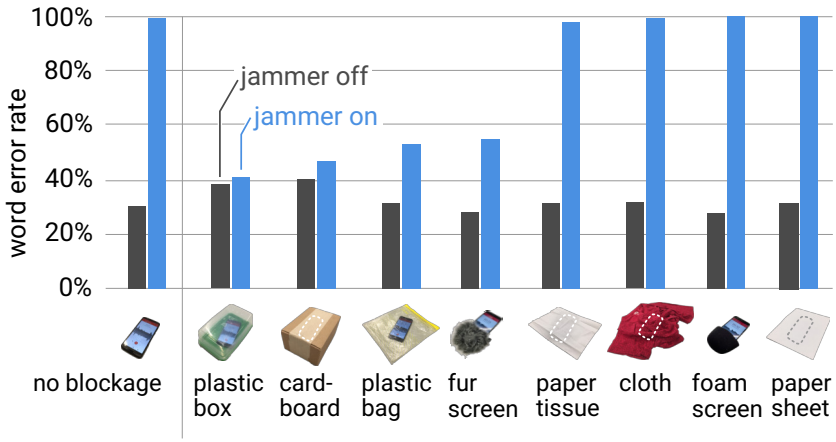

Figure 2.8: Speech recognition results when the microphone is covered up with various objects.

general, since ultrasonic waves are reflected/absorbed differently from those at the audible spectrum for a given material. Therefore, practitioners and consumers should be aware of such limitations, and a more in-depth investigation of materials accordingly to their resonant properties and acoustic impedance is required.

2.9 User study

In our earlier experiments, we focused on controlled laboratory experiments that validated the jamming effectiveness of our wearable jammer. In our final study we, instead, aim to understand whether wearing our jamming bracelet impacts one's feeling of privacy. This study was reviewed and approved by our ethics committee (IRB19-0927).

2.9.1 Study design

Participants engaged in group conversations that lasted four minutes. Neither the topic nor the volume of the conversations was controlled. We asked participants to speak one at a time (otherwise the speech recognizer cannot make sense of it) and to not disclose any personal or sensitive information. During the group conversation, participants wore our privacy bracelet

one at a time. They were asked to exchange the bracelet every minute, so that all could try it for an equal period. We recorded the conversation using four different commodity smartphones handed to each of the participants at the start of the study. We used the audio from all smartphones' recording for speech recognition. After the conversation was conducted, participants were presented with a transcript of the speech recognition (for the smartphone that they had during the study). After reading the transcript, they were asked to rate how much they felt that the bracelet had protected their privacy on a Likert scale (1-7). Lastly, note that the baseline of this study is implicit, as participants have a recollection of what they discussed in the group conversation and can judge how much the effect of the jammer influenced their perception of privacy. This study design does not allow us to repeat a non-staged conversation without the jammer nor were we interested in measuring actual word error rate (as we did that already in our previous controlled experiments).

2.9.2 Participants

To ensure that the English language level of each participant did not negatively reduce the fidelity of the speech recognizer, the candidates for this study were asked to read aloud sample sentences. Candidates who got over 70% accuracy were invited to participate in the study. As a result, we selected 12 participants (aged 18-26 years old; four self-identified as females and eight as males) from our local institution for this study. Ten of the participants had used some measure of privacy protection before, such as a laptop webcam cover, browser anti-tracking extensions, incognito mode, or VPN service. None of these participants had previously used a microphone jammer.

2.9.3 Apparatus

We used our jamming bracelet. We utilized four smartphones to record the conversation (models: Samsung S9+, Samsung S7, plus the aforementioned Nexus 6, and Xiaomi Mi 6).

Lastly, we again used IBM's speech recognizer.

2.9.4 Results

Participants rated the feeling of privacy induced by the bracelet as $M = 5.4$ ($SD = 1.1$). This result, coupled with their positive comments, which we discuss below, suggested that the bracelet provided a sense of protection for the recorded conversation. While the wearable jammer did not jam the microphones completely in all recordings, the overwhelming majority of the transcripts of the four-minute conversations had only a dozen of mostly erroneous words.

When asked about their experience with the wearable jammer, most participants stated that they felt the bracelet was "definitely blocking out most words". Participants also noted that in certain cases specific words still made it through, such as the word *facebook* (P3). Three participants commented that the bracelet is bulky but not uncomfortable (P7, P8, P12). Two also added that while at the start, the bracelet was noticeable, once they focused on conversation, they "forgot about it" (P4) or "stopped feeling odd about wearing it" (P2).

Two participants (P10 and P8) added that they felt more protected either when wearing the bracelet or by simply seeing others wear it. To this, P8 added that at the current size the device would not be discreet enough to jam without others being unaware that you are doing so. Some participants (P5, P2, P1) noted that they would have liked to better understand the range of the bracelet's efficiency.

Lastly, all twelve of the participants stated that they will use the bracelet again in the future. When asked specifically about the kinds of situations they would use it for, they noted, for instance: discussing private matters with their doctors (P1), discussing banking information (P6, P7, P10), talking to their employers (P9), or to strangers that joined a private conversation (P4).

2.10 Discussion

Our experiments and user study provided insights into the advantages of a wearable microphone jammer. We found in our experiments that a wearable jammer in a ring layout is likely to outperform stationary jammers or jammers with planar layouts. Furthermore, we found that our jammer actually provided participants from our user study with a sense of increased privacy against eavesdropping microphones. Yet, there are a range of questions and limitations that we believe are relevant to address to move the field forward.

2.10.1 Limitations of our experiments

While we designed our studies to be as insightful and exhaustive as possible, it is simply not possible to test out the jammer against an infinite amount of existing microphone-based devices. Therefore, one must take into account that while our jammer was extremely effective against the microphones we used, these word rate errors cannot be easily generalized to other devices. Furthermore, our transducers are placed around the user's arm in a circular layout, which decreases its vertical coverage. In a preliminary experiment (using the apparatus of our experiment#2) we found that our device provides a vertical jamming of over 97% (WER) up to 75°; however, the jamming drops at 90° (precisely on top of the bracelet) to 75.54% (WER).

2.10.2 Non-linearities of microphone hardware

One speculative question is whether the non-linearity of today's microphone hardware is just a transient artifact of today's devices. We believe non-linearity is likely permanent for the foreseeable future, because the MEMS microphones used for smartphones and voice-based smart devices are designed for low-cost and small form-factors [110, 178, 28].

2.10.3 Counter-attacks to our wearable jamming

It is possible that attackers might craft exploits to circumvent our wearable jammer. That being said, the most likely attack would be noise canceling techniques intended to cancel out the jamming signals. To provide some validation against this attack, we de-noised the microphone recordings of our jammed signal over our speech library (same as in our Experiment#2) using two methods: (1) the deep neural network (DNN) denoising method from Rethage et al. [169], and (2) the widely used Wiener filter [162]. We observed no improvement in the denoised speech (WER 99.64% for the DNN-based method, and 100% for the Wiener filter), when compared to the original jammed speech audio (WER 99.64%). We believe that these current de-noising techniques will be of limited effect because of two key factors of our design: (1) we use randomly changing signals, which are hard to predict and cancel out; and, (2) the motion of the user's gestures and movements is also hard to predict, making it also extremely hard to cancel out these moving signal sources. Furthermore, to make it even harder to perform noise canceling of the jamming signals, one could even design signals that exhibit cadence patterns similar to human voice.

2.10.4 Safety

Our proposed system uses ultrasonic frequencies in the 25kHz range, while the upper limit frequency that the human ear can hear is around 15k-20kHz. The U.S. Occupational Safety and Health Administration (OSHA) warns that audible subharmonics can be harmful at intense sound pressures of 105 decibels or above [173]. Therefore, we ensured our jammer did not surpass this threshold. We measured the sound pressure of our bracelet directly at the transducer and found that its maximum sound pressure is below 92dB, well within the aforementioned safety limits.

2.10.5 Unintentional and selective jamming

As with any of the current ultrasonic jamming techniques (not only wearable jamming), it is possible that a jammer could accidentally jam legitimate microphones if these happen to be well inside the jamming range, including one's own smartphone, hearing aids or emergency response devices. More work is necessary to understand the impact of ultrasonic signals on these devices and to design workarounds.

Similarly, a user cannot selectively jam devices using ultrasound jamming: e.g., a user cannot choose to avoid jamming their own smartphone while still jamming another device. On this limitation, our approach does provide more control than existing stationary jammers. Stationary jammers, once activated will jam their entire range, requiring the user to walk all the way to the jammer to disable it. In our case, users can control the jammer's behavior by simply touching the bracelet. Moreover, moving forward, one would expect that adding intensity control to the wearable jammer might allow users to tune the jamming range.

2.10.6 Future form factors

While we found that our device outperformed existing jammer approaches, it is still larger than a typical bracelet. We believe our prototype offers a great blueprint towards a low-cost and ubiquitous microphone jammer. We expect this to inspire other wearable jammer designs, such as necklaces, earrings or even clothing.

2.11 Conclusions

We proposed, engineered and validated a wearable microphone jammer that is capable of disabling any microphones in the user's surroundings, including hidden microphones. Our wearable jammer takes the shape of a bracelet worn on the user's wrist and jams ubiquitously.

Our device is based on a recent exploit that leverages the fact that when exposed to

ultrasonic noise, commodity microphones will leak the noise into the audible range. However, previous ultrasound jammers that also exploited this principle, required users to point the jammer to the target microphone. This was necessary as these devices were built based on planar transducer layouts and were, therefore, highly directional. Unfortunately, this is impractical because it requires users to constantly worry and operate the jammer by pointing it to the surrounding microphones. Furthermore, this is sometimes impossible as users might desire to protect themselves from hidden eavesdropping microphones.

Instead, we found that our device outperforms these state-of-the-art jammers: (1) our wearable jams in multiple directions since its transducers are arranged in a ring layout; and, (2) our wearable jammer leverages natural hand gestures that occur while speaking to blur out blind spots, which are the main disadvantage of any jammer based on multiple transducers. We validated these advantages by means of simulation and three laboratory experiments.

Lastly, we conducted a user study with 12 participants that revealed that in a life-like situation participants felt that our wearable protected their voice privacy. We believe our wearable provides privacy in a world in which more and more devices are constantly eavesdropping on our conversations.

CHAPTER 3

TYPING PRIVACY THREATS AGAINST WEARABLE KEYBOARDS

Like our speech, our typing (on a keyboard) is also a key modality for generating private content such as emails, documents, patents, etc. Existing works have demonstrated the impact of keystroke inference attacks, where attackers use sensor observations (e.g., video, sound) to infer our typed content. In this chapter, we describe our effort on understanding privacy implications of wearable keyboards.

3.1 Introduction

Jane walks into an airport lounge. With her flight boarding in an hour, she has just enough time to get online to write a few emails and pay some bills. Jane has heard stories about privacy risks of working in public places, e.g. people reading over shoulders and even stealing passwords with recording devices. So she finds an empty table in a corner, and before sitting down, checks the nearby area (e.g. ceiling, under the table) for sensing devices. Satisfied, she puts on her VR headsets, and starts working in Horizon Workrooms. As Jane notices some passengers walk by and a few others sitting with their own mobile devices, she wonders: "Is it safe for me to write sensitive content/emails or type passwords? Have I taken enough precautions to protect myself?"

In the age of machine learning and remote work, Jane's concerns are actually quite realistic (and common). Machine learning tools have grown increasingly proficient at extracting keyboard keystrokes from a variety of side channels and sensory data. Meanwhile, accommodations for remote work have untethered employees from their offices, and work is often done on the road or in public settings, e.g. airports, cafes, trains and airplanes, with different typing interfaces, e.g. phones, tablets, laptops and lately wearable keyboards [138, 187]. For

millions of affected workers, a simple question remains: "Are reasonable precautions enough to protect them and their data from invasive keystroke inference attacks in real-world settings?"

Despite extensive prior work on keystroke inference attacks, the answer to this question remains unclear. This is due in part to the reliance of existing attacks on novel yet restrictive scenarios where the attacker has access to specific types of sensor data or side-channel information. But under new wearable typing interfaces, those attack assumptions can be easily broken.

We consider existing keyboard inference attacks in two broad categories: vision-based attacks, and non-vision attacks (e.g. everything else). The latter group does not rely on vision techniques, but instead distinguishes keystrokes using data (e.g. audio, vibration) gained by placing sensors close to Jane. For example, audio-based attacks place a microphone next to Jane to capture key-specific sounds generated from typing on a mechanical keyboard [231]. RF-based attacks [111, 221] place WiFi devices close to Jane (e.g., 30cm) to capture subtle signal variations caused by her one-finger key entries. Other work explores the use of electromagnetic (EM) or LTE measurements to infer one-finger key entries. To succeed, they require either placing an EM sniffer right under Jane's table [93], or zero movement anywhere within 20 meters of Jane [115].

The other category of attacks are vision-based, and generally rely on strong assumptions on specific viewing angles, or other extra information such as precise keyboard layouts and reflective keyboard surfaces. Some attacks rely on direct (or birds-eye) views of Jane's keyboard and fingers, by either placing a camera above Jane [21] or by capturing a view of the screen reflected by her eyeballs [164, 213]. To help train inference models, a recent work produces synthetic data by overlaying a thumb image on mobile keyboards [113]. In contrast, other attacks operate on "normal" frontal views, but require extra visual cues to locate Jane's pressing fingertip. Not only must attackers know the exact keyboard location/size/layout,

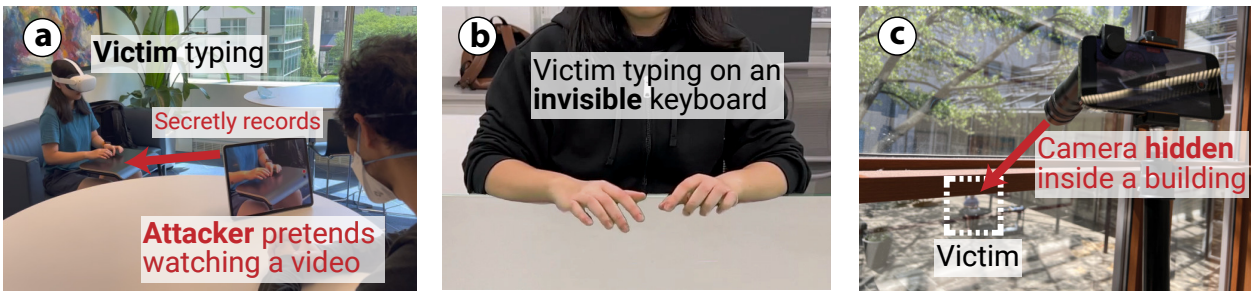

Figure 3.1: Sample attack scenarios: (a) an indoor lounge scenario where the attacker watches a video while recording the victim typing; (b) the victim can type on an "invisible" keyboard and types directly on the table; (c) a long-range outdoor scenario where the attacker hides a smartphone with a budget macro lens (<$60) inside a building (behind a window) to record the victim in the courtyard (≈12m away) typing.

they also need added info such as reflections around the pressing fingertip, created by a reflective typing surface [219], or the ground truth location of a key in both the video and the typed content, i.e., the "Enter" key on a PIN pad [179].

We note that wearable keyboards can disable most of these vision-based attacks, as when using wearable keyboards such as TapType [187] and Meta Quest 2 [137], the keyboard and its layout/size/position are known only to the user but remain hidden to any physical observers. By taking additional precautions, a privacy-aware user can also effectively disable most, if not all of these attacks. For example, Jane can avoid RF/EM attacks by looking around her work area for sensing devices, while the complex motion of her touch typing (using 10 fingers) is much harder to distinguish via RF/EM signals compared to 1-finger typing targeted by prior attacks. She can protect herself against existing vision-based attacks by checking for overhead cameras, and her eyes are naturally protected just by looking down at her device. Finally, audio based attacks are ineffective on today's touchscreen keyboards.

A vision-based wearable keyboard typing inference attack. In this work, we want to understand if today's wearable keyboard users can already resist keystroke inference attacks, after taking simple precautions such as those mentioned above. We ask if it is possible to recover text typing, by simply pointing a RGB camera at a wearable key-

board user's hands from a distance. Without external information from side-channels, can attackers invade Jane's privacy in realistic settings such as airports, coffee shops, or outdoor courtyards?

In this threat model, the attacker has no information about Jane, except that she types in English. The attacker has no knowledge of Jane's keyboard location, size, or layout, no videos of Jane typing known text, no knowledge or use of any visual cues, no access to or control of any sensor, device, and side-channel beyond a single RGB camera observing Jane at a distance. This threat model is designed to realistically capture what an attacker can do on a first encounter with a wearable keyboard user. Figure 3.1 illustrates several sample scenarios covered by our threat model.

Keyboard inference in this threat model is extremely challenging for several reasons. First, human typing is a complex process that is user-specific, highly variable, and heavily dependent on content and keyboard structure/layout [22, 53, 35]. Thus it is impractical to train a keystroke classifier that generalizes to different targets, devices and environments. Second, observations in a realistic attack are short, e.g., 15-minutes captures ∼3000 keystrokes, 2 orders of magnitude less than required for general self-supervised vision tools [57, 14]. Finally, hand tracking tools try to identify key-press events in videos, but suffer from large tracking errors due to artifacts in RGB video like depth ambiguity, finger occlusion (e.g., in frontal views, some fingers can be blocked by the fingers in front of them just enough that hand tracking cannot properly track them) and motion jitter.

A self-supervised approach to keystroke inference. In this work, we propose a new approach to keystroke inference with no additional input other than video captured from a distance via commodity phone cameras. The key insight is to use a two-layer self-supervised system, where noisy results of hand tracking on the target video are used to run keystroke detection/clustering, followed by a language-based Hidden Markov Model (HMM) to recognize keystrokes. These initial labels are filtered using multiple consistency checks

to produce high confidence labels on video frames, which are then used to train two 3D-CNN models that detect and recognize keystrokes from the video. This two-layer process is illustrated by Figure 3.2.

We evaluate this attack using IRB-approved user studies. Given the ongoing development of wearable keyboard typing at this moment (e.g., noticeable lag, low accuracy, and requiring specific typing style [195]; tapping-based keyboards have slow typing speed, low accuracy and/or significant training requirements), it is impractical to use wearable keyboards in user studies. Instead, we conduct our user studies using customized iPad keyboards to emulate the ultimate wearable keyboard typing experience. These customized iPad keyboards also allows us to emulate advanced wearable keyboards where users can use different keyboard layouts and even type on a physically invisible keyboard. We evaluate the proposed attack under a variety of conditions, varying the target (user/typing behavior, content typed, physical environment) and attacker behaviors (hand tracking tool, attack distance). The attack is highly effective in nearly all settings, and performs well across our user study participants, despite significantly different typing styles and abilities.

Ethics. Our goal is to bring attention to privacy risks from video-based keystroke inference attack in public settings. Beyond careful user study design to minimize participant harm, we believe our study can increase awareness to protect users, and lead to further adoption of simple and effective mitigation awareness, e.g. portable barriers to prevent line-of-sight.

3.2 Background and related work

We begin by discussing human typing, wearable keyboards, existing keystroke inference attacks and vision based hand tracking.

First, typing is a cognitively complex process that relies on many human factors: language/memory faculties, attention states, and typing muscle memory [1]. Human typing behaviors are complex, user-specific, time-varying, and heavily content-dependent, the key-

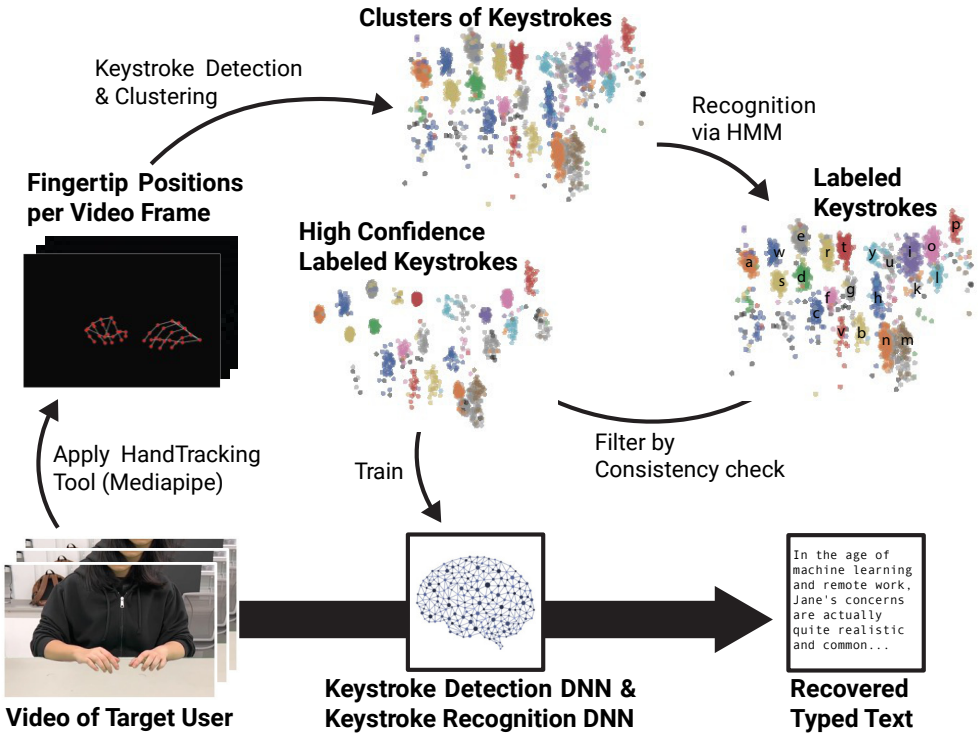

Figure 3.2: Our self-supervised approach to keystroke inference. We first run unsupervised inference on fingertip data extracted from each video frame, from which we identify keystrokes with high confidence labels (this process is marked by thin arrows). We use these as training data and build DNN models that detect and recognize keystrokes directly from the video (thick arrow).

board/input device, and other environmental factors [22, 32, 53, 35, 67]. Different typing styles and motions are a main reason why there are no known models that reliably extract multi-finger keystrokes from human hand/finger behaviors.

Second, wearable keyboards have been by widely explored by researchers, and more recently, they are emerging as commercial products. These wearable keyboards have several advantages/purposes: (1) moving the input away from size-constrained touchscreens, providing users with full-size keyboard typing-like experience; (2) saving touchscreen space; (3) providing passive haptic feedback by allowing users to type on everyday objects (e.g., a table); (4) supporting input over a prolonged period without fatigue; (5) provide a productive way to type in VR/AR/MR. These wearables record keystrokes as sensor data, followed by

a decoding phase to translate these data into typed keys. Different forms of sensor data have been used: acceleration/inertial-based [187, 193, 99, 71]; vision-based [139, 137, 59, 66]; inductive telemetry-based [191]; and infrared-based [69]. Most of these wearable keyboards leverage users' existing typing muscle memory, and/or display keyboards only to the user wearing the headset. As a result, a wearable keyboard's layout/size/position is only known to the user and is hidden to physical observers.

3.2.1 Existing keystroke inference attacks

There are numerous keystroke inference attacks in existing literature. Given our focus on general keystroke inference, we *do not* consider special subcases such as "invasive" attacks where the attacker has internal control over the user's device, and actively controls its sensors.

We broadly categorize existing attacks into two categories: non-vision attacks and vision-based attacks.

Non-vision attacks. The attacker collects data about the target user (i.e., Jane)'s typing by placing sensors near them. Sensors might include audio microphones or side-channels such as vibration, electromagnetic (EM) and RF.

Audio. Acoustic-based attacks place a microphone next to the target's keyboard to capture key-specific sounds generated by a mechanical keyboard [231], and its attack range can be extended to 15m using a bulky parabolic microphone [16]. These attacks work on multi-finger typing, but depend heavily on keyboard sound quality. They are generally ineffective when there is loud environmental noise or when the keyboard produces little sound.

Vibration. Key press events generate subtle vibrations. An accelerometer within 5cm of a keyboard can pick up vibrations induced by typing [130]. The physical proximity required by this attack makes it easy to detect in practice.

EM. Recent work identified that typing on a touchscreen generates EM signals (via

coupling), which vary with keys [93]. This attack only supports 1-finger inputs on a PIN pad, and requires an EM sniffer placed right under Jane's table. The attacker must know the PIN pad layout and position.

RF. By placing WiFi devices close to the target (e.g., 20cm [221]), an attacker could capture the subtle WiFi signal variation caused by 1-finger key entries. One attack [111] achieves 1.5m attack distance but requires the target to connect to an AP that the attacker controls. Furthermore, these attacks all require the exact PIN pad layout/position and user-specific training data [221, 38, 111]. Another study [6] targets multi-finger typing, but again requires placing WiFi devices close to Jane (30cm) and user-specific supervised training.

Cellular LTE signals can also be leveraged to infer 1-finger typing. A software defined radio within 15m of the target PIN pad can capture the LTE signal (sent by a LTE base station within 150m) reflected by Jane [115]. This attack fails if there is any moderate movement anywhere within 20 meters of Jane. Again, this attack requires knowledge of the keyboard layout and user-specific training.

Vision-based attacks. Vision-based attacks also often target 1-finger typing. We divide them into two groups based on the angle or "view" of the attacker.

Birds-eye view. Many attacks require a direct (bird's eye) view of the target's keyboard and fingers, as if the attacker is viewing through the target's eyes. This is done by either placing a camera above (or just behind) the target, depending on how the target places or holds the keyboard [21, 113], or by capturing the screen reflected by their eyeballs[1] [164, 213].

Frontal view. Other attacks can use indirect ("frontal") views, but require extra visual cues to locate fingertip keypresses. Aside from knowing the exact keyboard location/size/layout, the attacker must know the lighting/reflection patterns around the fingertip (by relying on a reflective typing surface [219, 220]), or the precise location of a specific (frequently used) key in both record video and the typed content, i.e., the "Entry" key on

1. The target holds a phone vertically while thumb typing. Thus their eyeballs or sunglasses reflect the phone's screen and the typing finger.

a PIN pad [179]. Finally, it is possible to record the target's upper body movement when typing, e.g. during a video chat, and use them to infer keystrokes [172]. Again, the attacker must know the exact keyboard layout.

Summary. Existing work has demonstrated the feasibility of keystroke inference attacks under novel but often restrictive scenarios where the attacker has access to specific types of sensor data and/or keyboard information. In this work, we consider a general (and more realistic) attack scenario, where the attacker uses only a frontal view of Jane's typing hands and nothing else (see the threat model in §3.3).

3.2.2 Vision-based hand tracking

Given our goal of general keystroke inference without side-channel data, we have to incorporate current vision tools for hand tracking. 3D hand tracking (or handpose estimation) is a long-standing problem in computer vision, and today's tools provide good but still noisy results, We describe current tools here, and later discuss how our system design overcomes errors generated by tools such as MediaPipe.

There are two types of hand tracking tools available today. Depth-based hand tracking [37] requires a high-precision depth sensor, which are either bulky (e.g., Microsoft Kinetic) or limited to short distance (e.g., iPhone's depth sensor works within a range of 50cm). In contrast, RGB-based hand tracking supports longer range, but is much more challenging due to occlusion, depth ambiguity, significant variation in camera viewpoint and appearance condition [144]. Today's SOTA models provide cm-level accuracy on known poses, e.g., 1cm mean error for a small set of gestures [107, 215] and 5cm mean error on others [144]. To our knowledge, there is no specialized hand tracking tool for keystroke detection. Prior work on keystrokes [67] tracks fingers by placing 52 reflective markers on hands and using 8 infrared cameras recording at 240fps, far from our realistic attack scenarios.

Mediapipe. Several general-purpose hand tracking tools can extract arbitrary handposes

from RGB videos at 30-60fps, and the most well-known is Mediapipe [223] (Google 2020). It extracts handposes as 2.5D coordinates of 21 joints per hand (horizontal, vertical, depth relative to the wrist). The public release includes only binary code, without the DNN model or its training data. Our work uses Mediapipe to extract handposes from recorded typing videos.

3.3 Threat model

In pursuit of a realistic attack in common everyday settings, we consider users who might be vulnerable while working in public places like cafes, airport lounges, or outdoors in courtyards or on park benches. We consider an attacker who records them typing (with a frontal-view video of their hands) from a distance using a single RGB camera (a commodity camera phone), then processes the video to reconstruct the typed content. Thus we make two simple assumptions:

- The attacker knows the language used by the target (English in this work)
- The attacker has a frontal view of the target's hands

Our work differs significantly from prior work in that we do not rely on side-channel data or other assumptions. These are assumptions that we **do not make**:

- The attacker has no knowledge of the target's keyboard layout (keyboard could be customized via third party apps like Gboard [11]), and may not have a clear view of the keyboard (e.g., typing on an iPad with a screen protector or a projection-based virtual keyboard)
- The attacker has no labeled data or prior observations of the target. This follows our assumption that the attack is opportunistic and has no prior planning.
- The attacker cannot install, access and manipulate any sensor, device and channel beyond the single RGB camera they are holding at a distance from the target.

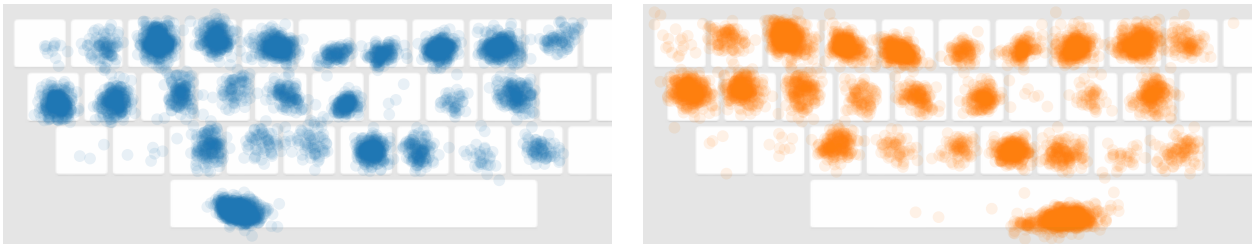

Figure 3.3: Touchpoints of keystrokes recorded by a touchscreen keyboard, where each circle is a touch point. The separation between neighboring keys's touchpoints is barely $1cm$ in average.

3.4 Design alternatives and system overview

In this section, we consider the challenges of a general keystroke inference attack, weigh two potential solutions, and describe their shortcomings. We then present a new self-supervised approach which when given a video, extracts specific frames and labeled keystrokes, and uses them to train customized DNN models that accurately detect and recognize keystrokes for that video.

3.4.1 Potential solutions and their limitations

In exploring the feasibility of a general keystroke attack, we considered a number of possible approaches, all of which produced less than effecitve results. We found two challenges to be the most difficult to overcome. First, users consistently hit edges of keys as they typed, meaning that for most users, the separation between positions of their fingers hitting two neighboring keys, is quite small. Figure 3.3 confirms this, using actual recorded positions of how two different users typed on their iPad keyboards. Second, we found that a reasonable length video (e.g. 10mins), provided roughly 3000 keystrokes for moderate speed typists, which is insufficient data for existing self-supervised tools to train a classifier for 27 keys (26 letters and space bar).

Here we consider in detail two potential attack designs: (1) training a supervised DNN

keystroke inference model on one set of users, and relying on model transferability to successfully apply that model to videos of other target users; (2) using unsupervised inference based on handpose data extracted from the video (using hand tracking tools), which is easier to model and interpret compared to a DNN. As we discuss below, we find that both faced significant limitations under our attack scenario.

Transferability-based attacks. An intuitive approach to general keystroke inference is to perform supervised training of a DNN keystroke inference model using labeled data collected from a set of users, then apply it to other target users. This leverages the concept of model transferability, the idea that models trained for one instance of a task can perform reasonably well on other instances of that or similar tasks.

In practice, we find that supervised inference models trained on one set of users fail to generalize when applied to video of other users. The implication is that the mapping from movements to keystrokes is user-specific enough to prevent transferability across users.

To confirm this, we recorded videos of 16 users typing on the exact same keyboard with the same camera angles. We applied transfer learning (using a well-known gesture recognition DNN model) to train keystroke recognition models, and performed leave-one-out cross validation. In each experiment we used labeled data from 15 users to train and tested the model on the 1 left-out user. While the trained models can correctly decode 99+% of keystrokes from any trained user, the transferability to the new user is very low – the mean character error rate is 48% and the word error rate is 98% across all the experiments. Note that this is already under near-optimal conditions where everyone uses the **exact same** keyboard.

Unsupervised inference using fingertip data. Without labeled training data of the target, one practical alternative is to run unsupervised inference directly on a processed version of the visual data. Since keystrokes directly result from fingertips touching the keyboard, the attacker can apply a hand tracking tool on the video to extract, for each frame,

the fingertip locations of the ten fingers, and use this data to detect and also distinguish between different keystrokes. Such analysis can then be combined with language-based models like HMM to infer the key of each detected keystroke. This is similar to the methodology used by prior audio-based keystroke inference attacks [231].

While attractive, this solution relies heavily on the accuracy of the fingertip data extracted from the RGB video. Since neighboring key presses are separated on average by less than $1cm$ (see Figure 3.3), the finger tracking precision needs to be at the level of millimeters. This is unfortunately infeasible with today's hand tracking tools. The resulting errors in the fingertip data propagate into the inference pipeline, and significantly degrade inference accuracy. Later in §3.5.5 we present a detailed study to illustrate its impact using different hand tracking tools.

3.4.2 Key insights

Curating labeled training data for a target video from its own fingertip inference result. When we observed that unsupervised inference on fingertip data is highly susceptible to hand tracking errors, we noted that its inference result is quite skewed: some keystrokes are accurately predicted while others are erroneous and cannot be corrected using post-analysis tools. If we can identify these accurately predicted keystrokes, we can use them as labeled training data to train DNN models that accurately detect and recognize keystrokes on the **same** video. Since the DNN models operate on raw video frames, they are no longer affected by errors introduced by hand tracking tools.

Identifying high confidence labels using consistency checks. We propose to identify correctly predicted keystrokes by checking the consistency between a keystroke's inference result (produced by a language model) and its spatial position on the keyboard estimated from its fingertip data. For all keystrokes assigned for the same key, the points where they touch the keyboard should form a tight cluster.

3.4.3 Attack design: overview

Following the above insights, we propose a new attack methodology, which applies two layers of data analysis on an attack video to curate labeled training data and then train DNN models that detect and recognize the keystrokes from the same video (see Figure 3.2). These high-performance DNN models take as inputs a sequence of raw video frames (rather than their hand tracking results) and output a sequence of characters as the recovered content.

Overall, the attack includes the following two steps, one per layer. We discuss each in detail in the subsequent sections.

Step 1: Unsupervised inference on handpose data (§3.5). Given a video on the target, we first apply a hand track tool (Mediapipe in our current implementation) to extract handpose data from each video frame. We then apply a unsupervised inference pipeline on this sequence of (noisy) handpose data to detect and cluster keystrokes, followed by a HMM-based language model to infer the character of each detected keystroke. This creates an initial label for each keystroke frame of the video.

Step 2: Self-supervised DNN inference on video data (§3.6). Given the initial noisy label of the detected keystrokes, we apply consistency checks to identify keystrokes with high confidence labels, and use them to curate labeled training data to train a DNN-based detector of keystrokes and a DNN-based classifer to recognize the detected keystrokes (i.e., mapping each to a key). We apply multiple noise-aware training methods to address any residue errors in the curated training data.

3.5 Unsupervised inference on handpose data

We start from the initial step of unsupervised inference on handpose data. This is done using a sequential pipeline: first detecting keystrokes (i.e, when a key is pressed), clustering keystrokes by their touchpoints, and applying a language-based analysis to estimate the

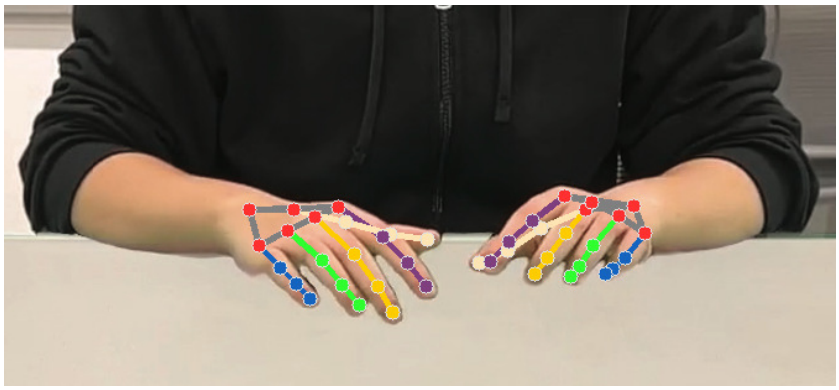

Figure 3.4: An example of Mediapipe hand tracking output.

typed content. While the methodology is similar to that of audio-based attacks [231], our contribution is realizing it in the context of general vision-based attacks. In the following, we describe the handpose data used by our pipeline, the three inference components, followed by a study on the impact of hand tracking noise.

3.5.1 Handpose data

Our pipeline operates on the fingertip coordinates per video frame, identified by the hand tracking tool. This configuration is carefully chosen to address finger occlusion and depth ambiguity of the keystroke video.

2D fingertip data. We focus on fingertips rather than all 21 joints provided by Mediapipe [201], because a keystroke is produced by a fingertip pressing down on the surface. Figure 3.4 shows an example of Mediapipe's hand tracking on video frame. Inference using fingertip data incurs less complexity but also less tracking errors. Furthermore, while Mediapipe provides a 2.5D[2] coordinate per fingertip (i.e., the pixel coordinate x, y and a relative depth to the wrist), we find that the relative depth carries little information but much unwanted noise as the wrist moves naturally during typing. As such, we only use the

2. To the best of our knowledge, there is no tool providing 3D tracking of keystroking fingers in a frontal-view RGB video.

2D fingertip coordinates per video frame.

Non-thumb data only. In frontal views, it is difficult to capture all 10 fingers due to finger-on-finger occlusion, especially when typing with multiple fingers per hand. When using Mediapipe in real-world attacks, we find that the target's thumbs are often blocked by other fingers just enough to prevent Mediapipe from tracking them properly (e.g., the detected thumbs flutter frequently). Thus we choose to operate on the 8 non-thumb fingertip data. Our design can still detect and recognize thumb-based keystrokes using non-thumb data, by leveraging natural correlation in finger motions.

Preprocessing. After extracting fingertip data from each video frame, we perform smoothing to remove potential noise. Here each fingertip has a sequence of pixel coordinate (x, y), one per video frame. We apply a standard low-pass butterworth filter with a cut-off frequency to smooth each individual fingertip's sequence. Since the common typing speed is around 200-300 keystrokes per minute (3-5Hz), we set the cut-off frequency to 6Hz.

3.5.2 Detecting keystroke events

Since the attacker has no knowledge (or even visual) of the keyboard, we propose to detect a keypress by detecting negative peaks on the fingertip acceleration – when a finger actively presses down and hits a key, its motion reduces/stops abruptly. Not yet knowing which finger touched the keyboard, we use the maximum value of the fingertip acceleration of all four non-thumb fingers (per hand) to run the detection. Here the acceleration is computed by taking the double derivative on each fingertip's y-coordinate across frames.

Handling spurious peaks. Interestingly, *not every negative acceleration peak maps to a keypress.* The spurious peaks come from two main sources: (1) the noise in fingertip data, and (2) the noise in human typing behavior since we often make unconscious hand movements similar to those of subtle keystrokes, e.g., when hesitating or thinking about what to type. As most spurious peaks have small prominence values, we apply statistical thresholding to

filter them out. Rather than pre-defining a "magic" threshold, we compute a threshold for the current attack video by modeling the peak prominence value p as a Gaussian mixture of keypress and no keypress ones. In this case, the dip between the two hills in the probability distribution of p would approximate the threshold required to produce equal mis-detection and false alarm rates.

Can thumb-based keystrokes get detected? Using only non-thumb fingertip data, our design can still detect keystrokes made by thumbs. This is because the muscles used to move our fingers are inter-connected, and thus the finger movements naturally correlated. When we press a key using a thumb, the other 4 fingers on the same hand also move down with it. Our acceleration based detection can detect thumb-based keystrokes, often at an accuracy comparable to those of non-thumb keystrokes.

One would think that since the acceleration of thumb-based keystrokes is computed from non-thumb fingertips, it should be weaker than those of non-thumb keystrokes. It is not true according to our measurements – the two show similar average peak prominence, and some non-thumb keystrokes are weaker than thumb ones (see the peak prominence distribution in Figure A.1 in Appendix). Thus in§3.5.3, we apply a different method to separate thumb and non-thrumb keystrokes.

3.5.3 Clustering detected keystrokes

Next, we organize the detected keystrokes (a mix of thumb and non-thumb ones) into clusters. This clustering result is later used in conjunction with a language model to infer the typed content (§3.5.4). We cluster keystrokes by estimating their touchpoints on the target's keyboard, which directly relate to the typed key. The exception is thumb-based keystrokes, where we can only estimate the "fake" touchpoint made by a non-thumb finger. Thus we propose to process them separately from the non-thumb keystrokes. With this in mind, our clustering process includes four steps: (i) identify the pressing fingertip, (ii) apply perspective

transformation to convert a 2D fingertip coordinate (obtained via the frontal view) into a touchpoint on the keyboard (i.e., the birds' eye view), (iii) separate the detected keystrokes into 2 groups: non-thumb and thumb based keystrokes and finally, (iv) cluster keystrokes in each group based on their estimated touchpoint locations.

Identifying the pressing fingertip. Since finger movements are correlated [216], negative acceleration used in §3.5.2 can effectively identify the keystroking hand, but not the pressing finger. Instead, for each frame, we estimate the vertical displacement of each non-thumb fingertip from its average vertical location across the video, and locate the finger with the largest displacement (to reach the keyboard). We note that due to depth ambiguity, this identification method is more effective for keys in the front row since their displacement estimation is more accurate.

Estimating a keystroke's touchpoint on the keyboard via perspective transformation. The pressing fingertip's 2D coordinate (x, y) is from the frontal-view video frame, and thus a skewed/compressed representation of its touchpoint on the target's keyboard. To reduce the effect of skew/compression, we apply perspective transformation to map each (x, y) to a touchpoint on the target's keyboard (in a birds' eye view). We first mark the 4 points on the video to indicate the keyboard's planar surface[3]. We then compute a homography matrix H between this planar surface and the video frame's perpective, using an OpenCV function (`perspectiveTransform`) [61]. By multiplying (x, y) with H, we estimate its corresponding touchpoint on the keyboard.

Separating non-thumb and thumb keystrokes. We separate them by analyzing the standard deviation of the typing hand's 4 non-thumb fingers' displacements. This is because a thumb keypress would trigger similar motions at the 4 non-thumb fingers (moving down together). In short, their displacements would be similar, thus the stds are generally smaller

3. It could be 4 corners of keyboard if the device is visible or 4 points on the table to indicate the planar surface.

than those of non-thumb keystrokes. We compute the threshold by treating the thumb keystrokes as a single key input, whose frequency is bounded by 20% [131]. This detection will introduce errors that depend on the target's typing behavior and keyboard layout.

Clustering non-thumb keystrokes. Given the estimated touchpoints of all non-thumb keystrokes, we run K-Means with 33 clusters to cover keys that can be inferred by a language model and to allow frequently used keys to form multiple clusters. This is because studies have shown that high frequency English keys can have more than 5 times amount of samples compared to low-frequency keys [200].

Clustering thumb keystrokes. We choose to cluster thumb keystrokes (instead of just separating them by the typing hand) to help mitigate errors made when separating non-thumb and thumb keystrokes. We first treat each detected thumb keystroke as a non-thumb keystroke, and estimate its touchpoint. We compute the distance of each "fake" touchpoint to the closet centroid of the non-thumb clusters (produced in the above step), and use this distance to cluster the thumb-based keystrokes. This produces roughly 10-15 clusters (depending on the attack video).

Identifying 'delete' cluster. We declare a cluster as 'delete' (or 'backspace') if satisfying two conditions: (a) the cluster is at the very edge of the touchpoint map, and (b) the cluster instances were pressed multiple times consequtively. Upon detecting the 'delete' keystrokes, we follow its actual operation to remove their previous keystrokes.

3.5.4 Inferring typed content via HMM

Given a sequence of detected keystrokes and the clusters of those keystrokes, the attacker can apply a language-based Hidden Markov Model (HMM) [163] to estimate the typed content [231]. This is done by exploring the causal link between the keystrokes (and their hidden states representing the typed key) and the clusters. This inference requires computing a transitional matrix $\mathbf{T}$ and an emission matrix $\mathbf{E}$. $\mathbf{T}$ is a $N \times N$ matrix that defines the

transition probabilities between the N hidden states, where N is the number of keys in the alphabet. Assuming the target types English, the attacker can pre-compute $\mathbf{T}$ using a large English corpus. For our attack implementation, we randomly select 40,000 sentences (52,000 unique words) from the CNN/DailyMail dataset [78], and set $N = 29$ to cover 26 letters, comma, period and space key. $\mathbf{E}$ is a $N \times M$ matrix, where $M = \#$ of clusters, $M < 50$ in our implementation. It measures the probability distribution of the N hidden states that produce the M clusters. HMM estimates $\mathbf{E}$ using a special Expectation-Maximization (EM) algorithm [25] to analyze the cluster data.

Given $\mathbf{T}$ and $\mathbf{E}$, the attacker applies the Viterbi algorithm [202] to infer the most likely hidden state sequence (or typed keys) for the keystroke sequence. Note that unlike [231], we do not pre-identify the thumb cluster as the 'space' key because we do not make assumption of the target's typing behavior. Furthermore, since EM runs local optimization [89], it can often converge to a local minima. Thus we run several randomly initialized iterations of HMMs and select the one that produces the most high confidence keystroke labels (discuss in §3.6.1). We empirically confirm that this also leads to the lowest character error. Finally, since $\mathbf{E}$ is estimated from the keystroke data, we find that 150-200 words are generally sufficient under perfect clustering and keystroke detection.

3.5.5 Impact of hand tracking noise

To examine the impact of hand tracking noise, we invited 2 volunteers (PA and PB) to type a randomly selected set of corporate emails (roughly 500 words, 28 sentences) on an iPad. This experiment is IRB-approved. We consider three hand tracking methods to extract fingertip data from the videos.

- **Perfect 3D tracking**: By tracking all 10 fingertips precisely in 3D, one should accurately detect when/where a fingertip touches the keyboard. We emulate this by assuming perfect keystroke detection and using the actual screen touchpoints recorded

Figure 3.5: The estimated touchpoints of the detected non-thumb keystrokes, using (left) perfect 3D hand tracking, (middle) 2D hand tracking using marker, and (right) Mediapipe. We mark each point by a color defined by its ground truth key entry. Black points indicate extra detections.

by the iPad as the input to the clustering algorithm.

- **Marker-assisted 2D tracking**: To emulate a high-performance 2D hand tracking, we place color markers near each participant's fingertips and locate each fingertip by its assigned marker on the video frame. The 2D tracking error is around 1cm. We also observe that the thumb tips are occluded in more than 32% of the video frames. Thus a practical attack should focus on non-thumb fingertips.
- **Mediapipe**: We use non-thumb fingertip data provided by Mediapipe. To avoid bias introduced by color markers, we ask our participants to type two sessions, one with markers and one not. We run Mediapipe on the video without.

As discussed earlier, the hand tracking error can affect keystroke detection, clustering, and HMM-based inference. Figure 3.5 plots, for user PB and the three tracking methods, the estimated touchpoints of the detected non-thumb keystrokes, which are the input into the clustering algorithm. Here we color each point by its ground truth key input. For both marker-tracking and Mediapipe, the tracking error creates overlaps among different keys, and misdetects some thumb keystrokes as non-thumb ones (i.e., the points in the very bottom).

Next, Table 3.1 lists the detailed results, from keystroke detection accuracy, clustering accuracy to content accuracy. For a fair evaluation, we also list the content accuracy after

		Detection		Cluster.	w/o GSpell		w/ GSpell	
		Miss (%)	Extra (%)	Acc. (%)	CER (%)	WER (%)	CER (%)	WER (%)
Perfect	PA	0.0	0.0	99.8	3.1	16.2	1.6	7.0
3D	PB	0.0	0.0	99.6	4.2	22.5	1.8	8.9
Marker	PA	6.0	5.0	88.9	31.5	78.5	29.0	54.8
Assisted	PB	6.0	3.0	93.2	26.0	60.4	23.5	39.6
Mediapipe	PA	7.0	5.0	85.4	40.3	84.3	39.4	67.1
	PB	10.0	6.0	85.5	55.2	82.6	54.7	71.7

Table 3.1: Performance of unsupervised inference on handpose data, using three different hand tracking tools.

applying a public spell check tool by Google Docs [83] (hereby referred to as GSpell for brevity). With perfect hand tracking, clustering is effective but not perfect because pressings near the key edge (compared to near the center) are harder to separate. The content recovery, evaluated as character error rate (CER) and word error rate (WER) (defined in §3.7.1), is also not perfect, mostly due to the error made by the HMM inference. But a standard spell check like GSpell can correct most of them. For the other two tracking methods, the tracking noise leads to 9-16% detection errors and 5-15% clustering errors, which propagate to the HMM inference component. Even after applying GSpell, the content accuracy is still low. The marker-assisted tracking is more accurate than Mediapipe, thus achieves better inference results. Together, these results demonstrate the severe impact of hand tracking noise and the significant difficulty facing a practical attack, which cannot assume perfect 3D tracking or marked-assisted tracking.

3.6 Self-supervised inference on video data

After applying unsupervised inference on handpose data, the attacker obtains a noisy label on individual frames of the attack video. That is, for each detected keystroke, its corresponding video frame is labeled by its inferred key (‘a’-‘z’, space, comma, period, as well as

'backspace'). The rest of the frames are not labeled (representing no keypress). While many video frames are wrongly labeled, our design seeks to identify the ones with high confidence labels and use them to train DNN inference models that operate on the entire set of video frames without applying hand tracking.

The key challenges facing this step include (1) how to identify high confidence labels, (2) how to train DNN models with limited training data to detect keystrokes and recognize their typed keys, and (3) how to suppress the impact of noisy labels during model training. We discuss them next in details.

3.6.1 Finding high confidence labels

Not knowing the keyboard layout, we filter keystrokes with high confidence labels using consistency check within each cluster and across clusters.

HMM label consistency within each cluster. HMM makes inference on the detected keystrokes by exploring how the keystroke clusters interact with the language model. Thus ideally, HMM should map keystroke instances in a cluster to a single key. But when the keystroke detection, touchpoint estimation and clustering are noisy, HMM often assigns different labels to keystroke instances in the same cluster to best match the language statistics. It can either incorrectly predict a legit and accurately clustered keystroke instance, or correctly predict a wrongly positioned and clustered keystroke instance. Considering this uncertainty, we propose to identify keystrokes with high confidence labels as those whose label matches the "majority label" of its cluster (i.e., the most popular label among the keystroke instances in the cluster).

Cross-cluster consistency check. Some detected keystrokes are false (i.e, no key is pressed) and some are identified with a far-off pressing finger. Together, these keystrokes can create multiple "spurious" clusters. HMM would wrongly predict most (if not all) instances in a spurious cluster, which the above intra-cluster check fails to address. Instead, we detect

them using a *cross-cluster* consistency check, leveraging the fact that any valid clusters whose majority labels are the same should be right next to each other on the touchpoint map. Specifically, we first sort the clusters by size, and starting from the largest cluster, find its majority label, mark this label as "claimed" and move to the next cluster. If the cluster's majority label has already been claimed and its touchpoint area is not close to the cluster who has claimed the label, this cluster is marked as 'spurious' and no instance is selected.

When doing cross-cluster check, we make an exception for clusters whose majority label is the 'space' key. This is because most users use one or both thumbs to type the space key. Recall that in §3.5.3 we apply a separate clustering on the thumb keystrokes based on their non-thumb touchpoint (i.e., computing the touchpoint by treating it as a non-thumb keystroke), which creates multiple clusters. If any of these clusters are labeled by HMM as 'space', it is most likely valid.

3.6.2 Training DNNs using limited data

After identifying keystrokes with high fidelity labels, we use them to train two DNN models, one to detect keystroke events and one to classify the key of the detected keystrokes.

Learning finger motion from a block of video frames. While our unsupervised inference operates on per-frame handpose data (to make the analysis tractable), both DNN models take as inputs a set of consecutive video frames (16 frames in our implementation). By operating on this short video segment, both models seek to discover and use rich finger motion features to make decisions, leveraging the labeled training data. Next, we discuss the detailed training process.

DNN-based keystroke detector. We implement the detection as a binary classifier. Since this detector will run against the entire attack video (a 10 min video has 360,000 frames), we choose a light-weight 3D-CNN model (ResNet-10 [77]) and apply transfer learning using a public teacher model, which is pre-trained using the EgoGesture dataset [227].

To train this binary classifier, we curate both positive and negative training data, leveraging the keystrokes detected by the unsupervised inference step without filtering. This is because the consistency check in §3.6.1 targets recognition consistency rather than keystroke detection consistency. For each detected keystroke, we find its corresponding video frame i and form a video segment of 16 frames, using 8 frames before i, i, and 7 frames after i. As such, the detected keystroke's frame is centered in the video segment. This video segment is labeled as 'positive.' To build 'negative' video segments, we apply a length-16 window on the video sequence between two consecutive keystroke frames. Finally, since the curated positive and negative labeled data will contain noise, we apply multiple techniques during the model training to identify/suppress noisy labels (discussed in §3.6.3).

At inference time, the trained binary classifier will scan through the entire video sequence, each time taking 16 consecutive frames as the input, and output a probability score. Thus near a keystroke frame, multiple video segments will have high probability values for 'positive.' We use a peak detection method to identify the video segment where the keystroke frame is in the center.

DNN-based keystroke classifier. For this multi-class classifier, we use ResNeXt-101 [211], a well-known 3D-CNN architecture for video-based classification tasks. The training pipeline is similar to that of gesture recogition [103] – we apply transfer learning from a public teacher model (ResNeXt-101 pre-trained on the Jester dataset [133]). Our classifier takes as input a video sequence of 16 frames, and outputs a label out of the 29 classes ('a'-'z', space, comma, period).

We build its training dataset using the high confidence labeled data (identified by §3.6.1). For each keystroke with a high confidence label l, we build a video sequence of 16 frames (8 before and 7 frames after), and label this segment with l. To reduce training complexity, we crop each video frame uniformly to only include the area around the hands/keyboard/table (56×56 pixels). Like the above, the keystroke frame is in the center of the video segment.

3.6.3 Noise-aware model training

Since the training data for both DNN models can be noisy, we apply three kinds of noise-aware training techniques [12, 183] to suppress their impact on the trained models.

Preventing overfitting. We apply Mixup [225, 226] to mitigate overfitting in our 3D-CNN models, which applies data augmentation like interpolation to smooth the decision boundary between classes. Under our input configuration, we achieve this by linearly interpolating two random training inputs (i.e., blending each pair of video frames in two video segments) and their labels (in terms of their one hot vector representations).

Identifying trusted samples. DNN models are known to have memorization effect [183, 218, 92, 12], where noisy labels take longer to learn than clean labels and thus have higher loss during early training stages. We leverage this effect to emphasize learning on small-loss training samples (which are likely to have clean labels), by giving these samples with larger weights in the overall training loss computation.

Self-correcting noisy samples. The above technique can identify trusted samples and some untrusted (noisy) samples. Instead of discarding those noisy samples, we apply the concept of label refurbishing [168] to correct them using the knowledge that the current model has learnt. Specifically, noisy labels are refurbished as a linear combination of their actual model inference result and their noisy label. Here we adopt dynamic bootstrapping [12] to adapt the weight of the combination based on the training loss. As such, noisy labels receive more supervision from the model while the model itself learns more from cleaner samples.

In our experiments, we find that all three techniques are beneficial when training the keystroke detector, while the first techique is already sufficient to train a high-performance DNN classifier, possibly because we only use high conficence labels as its training data.

3.7 Experimental evaluation

We evaluate our video based attacks using real-world user studies under a diverse set of conditions. All studies were approved by our Institutional Review Board (IRB info ommited for anonymity). In this section, we organize our experiments and their results by four groups:

- **Attack performance under different scenarios**, including environments (indoor/outdoor, varying attack distances and blockages), keyboard device (visible/invisible keyboards, varying size and layout), and content typed (§3.7.2);
- **Attack performace across 16 different users**, who have different typing behaviors and abilities (§3.7.3);
- **Contributions of individual components** (§3.7.4);
- **Attack complexity** in terms of computing time (§3.7.5).

3.7.1 Experiment setup

Target (or victim) configuration. In each experiment, we ask our study participants to sit naturally in front of a table and place a keyboard device at their comfortable position. By default, we ask them to type email sentences (about 500 words) randomly selected from the Enron corporate email dataset [44]. For a fair evaluation, we ask each user to correct any typing errors using the backspace key, so that the final content matches the chosen data. As such, the actual keystroke data covers 26 letters, space, comma, period, backspace, and any other characters that they wrongly pressed and later corrected using backspace.

We ask our participants to type on customized iPad keyboards instead of wearable keyboards (except one study using a portable physical keyboard to demonstrate the attack effectiveness against various typing devices). This is because: (1) the less satisfied user experience of wearable typing at this moment. For example, VR typing has noticeable lag and low display resolution of keyboard while requiring users to be able to touchtype [195];

tapping-based keyboards have slow typing speed, low accuracy and/or significant training requirements. By using a iPad keyboard, we can emulate the ultimate wearable keyboard typing experience; (2) currently wearable keyboards are lack of customization support. Our customized iPad keyboards allow us to emulate advanced wearable keyboards where users can use different keyboard layouts and even type with a physically invisible keyboard.

We do not apply any restriction to our participants except that the table and the keyboard device stay stationary during each study. Our participants are free to move and leave the seat during the study. In fact, to encourage natural movements, in our indoor experiments we placed a cart of snacks nearby, which they need to leave the chair to reach, and many did so.

Attacker configuration. We consider an attacker who uses their smartphone camera to record the keystroke video. We experiment with three iPhone models (iPhone X, 13Pro, 13Pro max) where the recorded video is consistently set to 1280 $\times$ 720p at 60fps. This is also the common camera setting for today's phones. Since our attack implementation uses Mediapipe to extract handpose data, the attacker needs to position the phone such that both hands are visible and that Mediapipe is working. This is done using the real-time hand tracking API provided by Mediapipe [135]. In general, we find that the camera needs to be 10° above the keyboard. As such, the camera height will increase gracefully with the attack distance.

The attacker runs spell correction to further polish the recovered content. We build a fully automated spell correction using Google Doc's built-in function. While Google does not provide an API for this, we develop a browser automation script using the Selenium library [177] to access the function. We query it at different levels of granularity (paragraph, sentence, phrase and word) to maximize correction effectiveness.

Evaluation metrics. We evaluate the effectiveness of the attack by comparing the typed and recovered content at the character, word and semantic levels, and the accuracy of

recognizing individual keys (precision/recall).

- **Character error rate (CER)**: We compute CER as the total Levenshtein Distance between the typed and recovered content divided by the number of characters in the typed content. The Levenshtein Distance between two strings [108] measures the minimum number of character-level operations (insertions, deletions and substitutions) to convert one string into another.
- **Word error rate (WER)**: This is similar to CER except calculated at the word level. We use a public NLP tool [63] to compute WER, which applies dynamic string alignment to match words. We note that WER is a highly strict metric since one incorrect character is counted as a word error even when the word is comprehensible given the context.
- **Semantic content similarity (Similarity)**: We evaluate the semantic similarity between the typed and recovered content using CopyLeaks [46], a commercial tool for detecting plagiarised and paraphrased content. It reports a similarity score between 0-100% that accounts identical, minor changes and related meaning between any two documents. The similarity score is generally higher than 1-WER by capturing semantic correlation in words/sentences. However, we find that when WER is high (>50%), CopyLeaks produces a similarity score (much) smaller than 1-WER.
- **Per-key precision and recall**: Finally, we also compute the precision and recall of each character typed by the target to analyze the recovery rate for each indiviudal key.

In §3.7.3 we also study the effectiveness of recovering websites typed during the study, in terms of **top-k accuracy**.

3.7.2 Performance under different scenarios

We begin by a set of experiments to understand the feasibility of launching the proposed attack under different scenarios, exploring the impact of physical environment, attack distance,

Distance (m)	Height (m)	CER (%)	WER (%)	Similarity (%)
Indoor, visible keyboard (iPad)				
0.8	0.3	1.1	6.0	98.8
1.8	0.6	1.1	5.2	98.0
2.4	0.6	0.3	1.8	99.4
3.0	1.1	0.4	2.0	99.4
Indoor, invisible keyboard, no visual cue				
1.8	0.6	0.5	2.6	99.6
2.4	0.6	1.1	3.8	98.0
3.0	1.1	1.0	4.0	99.2

Table 3.2: Attack performance in an indoor environment at different attack distances. The camera height (2nd column) refers to the relative distance above the keyboard.

keyboard device/layout, typed content, and attack observation window. For consistency, we invited a single participant to run all the experiments.

Scenario #1: indoor lounge, varying attack distance. We consider typical indoor public spaces like a lounge or cafe, where the target sits by a table and uses an 12inch iPad's on-screen keyboard to type corporate emails (from the Enron email dataset). We set four iPhone cameras at 0.8, 1.8, 2.4 and 3 meters away from the target. The camera heights are 0.3, 0.64, 0.64 and 1.09 meters above the target's keyboard. As mentioned earlier, the camera needs to be 10° higher than the keyboard for Mediapipe to function, thus the height increases with the attack distance. We use the camera's built-in optical zoom-in (1x for 0.8m, 2x for 1.8m and 3x for 2.4 and 3m) to capture the keystroke videos (60fps, 720p).

Table 3.2 summarizes the attack performance in terms of CER, WER and Similarity at the four attack distances, after the target has typed 28 email sentences (501 words, 10.9 minutes). Across the content recovered from the four attack videos, the CER is consistently low (0.3-1.1%), while the WER varies between 1.8% and 6.0%. This variance is mostly caused by the difference among the four videos. But more importantly, the semantic similarity between the typed and recovered content is consistently high (¿98%).

Scenario #2: indoor lounge, invisible keyboard. Next, we consider a more "ex-

treme" case where the keyboard device itself is invisible to the attacker, e.g., the target uses a VR/AR system to view the keyboard in their own VR world and type directly on the table surface. We emulate this scenario using the well-known green screen method – covering the table with green cloth, changing the iPad's screen display to green hue, recording the attack video, and then keying out green colors. This allows us to remove the keyboard completely from the video while preserving the participant's hands. An example frame is shown in Figure 3.1 (d). Table 3.2 summarizes the attack performance. The mean CER, WER and similarity are 0.8 ± 0.3%, 3.4 ± 0.6% and 98.9 ± 0.7% across the three distances. This result confirms that our attack *does not rely on any knowledge of the keyboard or any visual cue of the keystrokes on the keyboard.*

Scenario #3: indoor, blockage by passing pedestrians. In public spaces, passing pedestrians can block the attacker's view of the target from time to time. Using local measurements, we find that each blockage lasts roughly 0.2s or 12 consecutive video frames. Thus we emulate on a given video the effect of P=5 and 10 passing pedestrians per minute, each blocking 12 consecutive video frames. The blockage instances are randomly distributed over time but do not overlap with each other. The chosen P values represent one passing pedstrian every 12 and 6 seconds, respectively, which correspond to very busy environments. Table 3.3 summarizes the attack performance, in terms of mean and std for CER, WER, and Similarity, since we run 5 experiments per P value. We see that while blockage by pedstrians increases CER and WER, our attack can still recover most of the content at a high seminatic similarity.

Scenario #4: outdoor, long-distance, through-glass attack. We also consider scenarios where the target is working in an outdoor courtyard, while the attacker records a video at a distance longer than the indoor scenarios. Specifically, we position a smartphone inside a nearby building's second floor, behind the glass, to record the target's typing. Here the target cannot observe the attacker (see Figure 3.6). The smartphone's camera is roughly

# Human Passing per minute	CER (%)	WER (%)	Similarity (%)
0	1.1	6.0	98.8
5	5.8 ± 0.3	15.7 ± 1.6	92.9 ± 2.1
10	9.8 ± 0.5	22.1 ± 1.0	84.9 ± 1.5

Table 3.3: Attack performance when passing pedestrians block the attacker's view of the target from time to time.

Figure 3.6: The exprimental setup of our long-range, through-glass attack. The attacker video-tapes the victim's hands, by placing a smartphone camera with a budget macro lens inside a nearby building's 2nd floor, behind the glass.

12 meters away from the target. We attach a budget macro lens (less than 60 USD) to the camera to help zoom-in onto the target's hands. Despite the complex lighting condition (sunlight, glass reflections and shadows), our long-range attack is still effective – recovering 82.4% of typed words and achieving a high semantic similarity of 87% (see Table 3.4). In parallel, we also set up another smartphone camera in the courtyard (4.5 meters away from the target), which is able to recover 96.8% of typed words accurately. Comparing the two videos (of the same typing session), we find that the 12m/through glass video appears more bland or dull, which likely affected the overall inference quality.

Scenario #5: Varying keyboard type, size and layout. We are interested in understanding how our attack performs when the target uses different typing devices or keyboard layouts. We ask our participant to type on three different portable keyboards: 12.9-inch iPad, 11-inch iPad, and a bluetooth folding keyboard purchased from Amazon.

Attack condition	Distance (m)	CER (%)	WER (%)	Similarity (%)
Outdoor, open space	4.5	0.9	3.2	96.0
Outdoor, through-glass	12.0	5.2	17.6	87.2

Table 3.4: Attack performance in long-range outdoor scenario.

Typing Device	Dimension (mm)	CER (%)	WER (%)	Sim. (%)
iPad Pro 12.9 inch	281 x 215	1.1	6.0	98.8
iPad Pro 11 inch	248 x 179	1.8	6.0	95.9
Foldable keyboard	210 x 85	0.9	4.2	98.8
iPad, secret layout	281 x 215	2.4	6.6	96.4

Table 3.5: Attack performance on different typing devices.

The first two are touchscreen keyboards and the third one is a more compact, rubberish keyboard. Results in Table 3.5 show that the attack is highly effective for all three keyboards.

We also explore the impact of keyboard layout on the attack. We emulate the case where the target uses a secret layout that is significantly different from the default QWERTY layout: we change the 'a' key in the QWERTY layout to 'z', 'b' to 'a', 'c' to 'b', ..., 'z' to 'y'. Since it takes a lot of practice to type well on a new layout, we let the participant type on the original layout, but modify the content to be typed to emulate the layout change (e.g., the target inputs 'bme' instead of 'and' in the QWERTY layout to emulate typing 'and' using the new layout). The attack result is shown in the last row of Table 3.5, which is similar to the rest of the table. This shows that our attack is effective against customized layout.

Scenario #5: Varying content type and length. Finally, we examine attack performance when the target types different types of content. Beside corporate emails, we also consider machine learning paper abstracts (numerous technical terms), a Shakespearean play (Coriolanus) with numerous medieval period phrases, and medical patents (numerous medical terms). Table 3.6 summarizes, for each experiment, the content type and length (# of

	Content Length			Recovered Content		
	#Words	#Sen.	Dur. (min)	CER (%)	WER (%)	Sim. (%)
Paper Abstract	696	37	16.7	2.3	11.4	90.5
Shakespeare's Play	732	26	14.5	1.8	9.8	92.0
Medical Patent	708	36	18.2	0.7	6.5	97.3
Corporate Emails	654	40	13.9	1.1	5.5	99.1
	501	28	10.9	1.1	6.0	98.8
	199	10	4.3	2.8	13.1	94.5

Table 3.6: Attack performance when the target types content of different kinds and lengths. For corporate emails, the participant typed 40 sentences. We then shortened the video to match 28 and 10 typed sentences, respectively.

words, # of sentences and video duration). Our attack remains highly effective across the four very different types of content. Furthermore, even by observing just 10 email sentences (199 words, 4.3 minutes of observation), our attack can successfully recover 87% of the typed words and achieve a high similarity score (94.5%).

3.7.3 Performance across different users

Next, we examine our attack performance on different individuals, exploring the impact of typing styles and behaviors. We recruited 16 participants (P0-P15) locally (mean age=24.4 years, std=6.4 years; 6 females, 10 males). For consistency, we use the same 12.9in iPad as the typing device. The camera is placed roughly 0.8 meters away from the keyboard, 0.25-0.4 meters above the keyboard. The actual camera placement is chosen to obtain a stable result in Mediapipe (using its real-time API), which varies slightly across the 16 participants since they have different typing gestures. The typed content is a set of emails chosen from the Enron email dataset [44] ($\approx$ 500 words). Across the 16 participants, the (active) typing time is 10.7 ± 2.5 minutes. In addition to the emails, the participants typed 25 websites randomly extracted from the top-1000 websites in [127].

		Email			Website	
	CPM	CER (%)	WER (%)	Sim. (%)	Top-1 Acc.(%)	Top-3 Acc.(%)
P0	169	0.7	3.4	99.6	100.0	100.0
P1	292	1.1	6.0	98.8	96.0	100.0
P2	284	2.3	8.4	97.2	96.0	100.0
P3	319	3.6	11.2	94.8	50.0	62.5
P4	291	5.0	12.1	93.5	100.0	100.0
P5	329	5.8	16.5	90.4	88.0	96.0
P6	313	5.2	14.9	87.6	92.0	100.0
P7	276	5.1	20.0	84.0	92.0	96.0
P8	342	8.0	25.5	83.4	96.0	100.0
P9	344	6.9	17.8	79.9	96.0	96.0
P10	331	11.6	32.9	71.0	92.0	100.0
P11	379	12.3	44.8	62.8	100.0	100.0
P12	308	13.6	35.5	59.0	68.0	76.0
P13	415	22.8	62.7	14.8	76.0	88.0
P14	198	1.0	3.6	97.4	96.0	100.0
P15	322	3.9	15.2	91.0	96.0	100.0

Table 3.7: Attack performance for all 16 participants (P0-15). The CPM column refers to their typing speed.

Observed typing behaviors. Our 16 typists display different typing behaviors. First, the number of fingers used varies – 13 participants use multiple fingers per hand while 3 use only two index fingers. The detailed finger usage is in Table A.1 in Appendix. Second, the typing speed varies largely between 169 character-per-minute (CPM) and 415 CPM. Finally, 6 participants exhibit multi-touch behaviors, where they press the next key without releasing the current one, e.g., while the index finger is still pressing 't', the pinky presses 'a'.

Failed Mediapipe cases. We find that Mediapipe functions reasonably with some occassional flickers, except for 2 participants (P14 and P15). For both, Mediapipe failed to produce consistent results, where the detected fingers shift largely across frames. This is likely because these two users have very long, thin fingers. Instead of removing them from our evaluation, we apply the marker-assisted 2D tracking (see §3.5.5) by placing color tapes on their nails (except for the thumbs) and run our attack.

Overall attack performance. Table 3.7 summarized the attack results for all 16 participants. Our attack can effectively recover the typed corporate emails. The mean CER, WER and similarity are 6.8% ± 5.8%, 20.7% ± 16.2% and 81.6% ± 21.7% between the inferred sentences and the ground truth. The attack performance does vary largely across the participants. For 10 out of 16 participants (P0-9), our attack achieves a low CER (0.7%-8%) and a high semantic similarity ($\geq$ 80% for 10 participants, $\geq$ 90% for 6 participants). For the two non-Mediapipe users (P14, P15), the accuracy is also high. We provide some samples of recovered and original text in Figure A.2 in Appendix, at difference CER values (3.8% – 11.8%).

Our attack can also effectively recover websites typed during the attack window. Given the recovered text, we compute the edit distance to each website of the top-1000 websites to compute top-k accuracy. Across all 16 participants and websites tested, the top-1 accuracy is 89.6% ± 13.7%, and the top-3 accuracy is 94.7% ± 10.7%.

Three less effective cases: P11-13. Our attack is less effective on P11, P12 and P13. After a deeper study, we identified their unique behaviors that affect the attack performance.

- P13: *multi-touch, high speed typing* – Typing at a blasting speed of 415CPM, P13 used multi-touches constantly and the finger motion was much weaker than others. It is hard to detect and recognize these very subtle keystrokes.
- P12: *fake presses and 2-hand presses* – P12 exhibited frequent hesitation-retraction, i.e. press down towards a key, hesitate and then retract the finger(s) before hitting the key. The motion of these "fake" keypresses matches that of real keypresses. Also P12 often presses keys using both hands simultaneously (i.e., multi-touch by 2 hands). Our current attack design does not consider this case.
- P11: *subtle thumb presses* – For P11, another high speed typist at 379CPM, our attack missed 'space' keystrokes more often than other users. This is because P11 typed 'space' with a thumb so subtel that there is very little motion at the non-thumb fingers. This

Characters	Avg. # appear.	Avg. Freq. (%)	Precision (%)	Recall (%)
Space	500	16.6	94 ± 5	93 ± 8
e, t	258	8.5	97 ± 3	95 ± 4
a, o, i, n, r, s	173	5.7	96 ± 4	95 ± 4
l, h, d, c, u, m, y, g	76	2.5	93 ± 5	91 ± 10
w, f, p, b, v	38	1.3	91 ± 12	90 ± 9
k, q, x, j, z	5	0.2	92 ± 14	61 ± 26

Table 3.8: Character-level precision and recall for all participants, bucketized by # of appearances of the character in the content.

contributed to the fast typing speed but also misled our attack.

Impact of character frequencies. We examine the inference accuracy on the character level. We found that characters that are frequently used in the typed content are more accurately inferred. This is because they appear more often in the high confidence training data. Table 3.8 lists the character-level precision and recall, where we group characters in 6 buckets based on the number of appearances in the typed content ($\geq$500, $\geq$200, $\geq$100, $\geq$50, $\geq$25, <25). We report the mean $\pm$ std for both precision and recall across the characters in each bucket. While the frequency does affect the inference result, we only see visible degradation at the last bucket whose average frequency is 0.2% and only appeared 5 times in the content in average.

3.7.4 Contributions of different components

To understand how each component contributes to the attack, we conduct an ablation study on P3 and P9, who display different performance levels. Table 3.9 lists P3's result. P9's result is in Table A.2 in Appendix and follows the same trend. For a fair comparison, all the reported results were obtained after running the same automatic spell correction function.

These results show that the unsupervised inference on handpose data is highly sensitive to hand tracking noise. By selecting high confidence labels to train DNN detector and classifier,

Unsup. Infer	*DNN Detector*	*Label Filter*	*DNN Classifier*	*Noise Train*	**CER** (%)	**WER** (%)	**Sim.** (%)
✔					22.5	59.4	9.1
✔	✔				16.2	46.0	47.4
✔	✔	✔	✔		5.0	16.1	88.7
✔		✔	✔	✔	8.3	23.5	78.5
✔	✔	✔	✔	✔	3.6	11.2	94.8

Table 3.9: Contribution of each design component, tested on P3.

our attack reduces the CER from 22.5% to 3.6%, and boosts the semantic similarity from 9.1% to 94.8%. We can also clearly see the contribution of individual components.

3.7.5 Attack complexity

We test our attack pipeline on a server with a Intel Xeon Silver 4214 CPU and a NVIDIA TITAN RTX GPU. For a 12-min attack video (500 words), it takes 40 minutes to produce the final spell-corrected content. Specifically, the unsupervised inference takes 9.8 min (dominated by HMM's EM optimization), the DNN detector takes 10.3 min (8.3 min spent on model training), the DNN classifier takes 10.2 min (9.8 min spent on model training), and finally the automatic spell check (GSpell) takes 8.8 min. Here we exclude the Mediapipe extraction time since it can be done in real-time while recording the video.

3.8 Conclusion

This work describes our experiences developing a vision-based keystroke inference attack against wearable keyboard users. Our results show such attacks can succeed in realistic scenarios, and users working in public settings should take precautions to protect their privacy from potential attackers. Beyond checking nearby areas for suspicious sensors and microphones, users should consider physical screens that block external line of sight to their hands while typing. This is likely the easiest and most effective defense against these attacks. Other

potential defenses include modifying the wearable keyboards to emit specialized lights onto the user's hands to obfuscate the attacker's observation, or dynamically varying the wearable keyboard layout, or including "false" typing content to break the attacker's inference pipeline.

We also note limitations and caveats to the proposed attack. First, our study found that today's hand-tracking tools were unable to accurately track fingers at high speed. Future improvements in hand-tracking might overcome this challenge and lead to simpler systems for vision-based keystroke inference. Second, our self-supervised approach still requires data of valid keystrokes for a particular key to be recognized by the eventual DNN models. Keys that appear very infrequently in the video will have noisier curated training data, and less accurate recognition. Note that their infrequency means these errors will have lower impact on overall inference of typed text.

Finally, our experiments assumed traditional typing sessions in stationary settings. Our results may not hold for hybrid input methods that augment keystroke typing with either predictive text or voice-based input, or in active settings, e.g. a moving train or airplane experiencing turbulence.

CHAPTER 4

ON-ARM ELECTRICAL MUSCLE STIMULATION FOR USER AUTHENTICATION

While Chapter 2 and 3 focus on protecting content privacy (related to speech and typing), in this chapter we turn our attention to identity privacy and its impact on user authentication. Today, biometric authentication is widely used — our face, voice and fingerprints are being used to identify us. However, these (static) biometrics can also be captured by surrounding sensors, and once leaked to attackers, they can no longer be trusted. In this chapter, we propose an alternative form of user authentication using electrical muscle stimulation which is robust against data leakage.

4.1 Introduction

Biometric authentication is a technique that identifies an individual by their unique biological characteristics, such as their iris [206], fingerprints [128], or even one's voice [33]. To identify their users, these interactive systems compare a previously stored biometric key to incoming, typically real-time, biometric data of the user wishing to authenticate. Compared to traditional password or PIN based systems, biometric authentication offers significantly better usability as it does not require users to memorize passwords or PINs. As such, biometric authentication is getting widely adopted, replacing passwords in many contexts [185].

However, the key feature of biometric authentication is typically also its key flaw: once the biometric data is compromised (e.g., stolen in database breaches or recorded by an external attacker), there is nothing the user can do to securely re-use their own data. For example, if someone steals a user's fingerprints, this user can never trust a fingerprint-based interactive system. Unfortunately, these threats are not theoretical and many biometric systems have been breached. For instance, the biggest known biometric data breach involved a database

of 27.8M records, including fingerprints and faces [85].

To tackle this shortcoming, researchers turned to interactive systems that feature a *challenge-response* as a form of active biometric authentication. One example is Velody [109], which challenges a user by vibrating her palm and measuring the user's unique vibration-response. The advantage of these systems is that, if the stored challenge-response pairs are breached, the system can quickly recover by simply asking the user to submit responses to a new set of challenges. As such, researchers seek to find more modalities that afford challenge-response biometric authentication.

In this work, we propose and explore a novel modality for active biometrics: electrical muscle stimulation (EMS). To understand and evaluate the potential of EMS as a biometrics system for interactive applications, we engineered a prototype that performs user authentication via EMS. Our system, which we call ElectricAuth, stimulates the wearer's forearm muscles with an EMS-based challenge, i.e., a 1.2s long sequence of electrical impulses on four of the user's muscles. Then, it measures the user's involuntary movements that result from this EMS challenge. In Figure 4.1, we illustrate our system with the example of authenticating a user in VR. Here, ElectricAuth uses the VR headset's hand tracking to observe the response of the user's muscles to the EMS-challenge as their individual finger muscles are actuated.

ElectricAuth makes three key contributions in the design of EMS-based biometric authentication.

First, ElectricAuth authenticates users by leveraging what is typically seen as the biggest disadvantage of EMS: *intersubject variability*, i.e., the same electrical stimulation results in different movements in different users because everybody's physiology is different [101, 52, 58, 42, 140]. This unique response to EMS across users is well-known and well-documented in the early HCI works that pioneered the use of EMS in interactive devices, for instance: "(..) stimulation level differed between users and was clearly dependent on the muscle and fat level

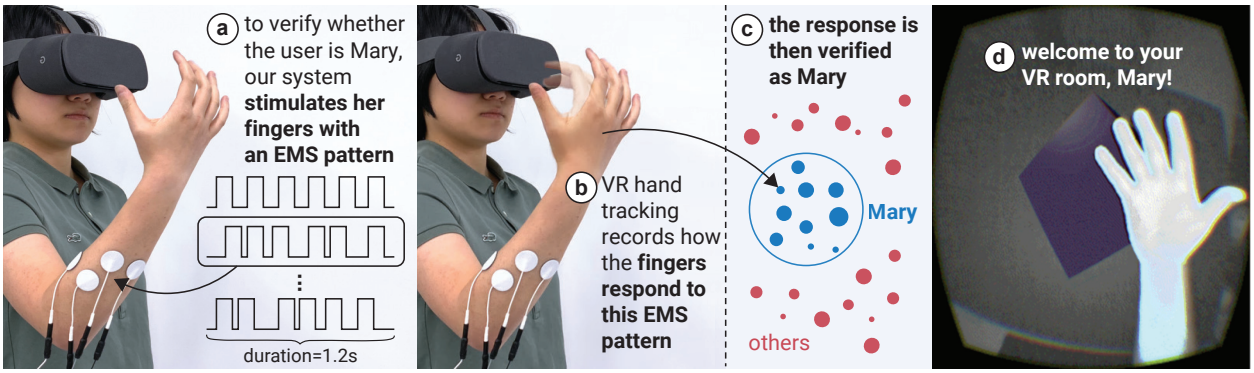

Figure 4.1: We propose a novel modality for authentication: electrical muscle stimulation (EMS). To explore it, we created an interactive system that (a) stimulates the user's forearm muscles with electrical impulses (i.e., using one of 68M possible EMS challenges); (b) measures the user's involuntary finger movements, which are unique because everybody's physiology is different; (c) verifies this response using an authentication model, and immediately eliminates this challenge, making our system secure against data breaches and replay attacks as it never reuses the same challenge. We demonstrate it here using the example of (d) authenticating a VR user without passwords or PINs.

and thickness of the arm" (from Kruijff et al. [106]) and, similarly, "(...) levels according to individual variations" (from PossessedHand [192]). In fact, researchers in the field of muscle-biomechanics and physiology demonstrated how this uniqueness arises from multiple factors, such as differences in muscle contractility [75], muscle elasticity [199], muscle viscosity [51], the limb's mass and shape [146], skin conductance [112], bioimpedance [47, 176] and even nerve conduction [3]. All these differences add up to create individual responses to the same stimulus, which our system uses as the key feature to authenticate a user.

Second, ElectricAuth generates a very large pool of challenges by exploring an under-utilized property of EMS: muscles respond differently depending on their current state of contraction, which can be altered by varying the timing between two impulses. Using four muscles, six impulses and seven time gaps, ElectricAuth encodes one of 68M possible challenges in 1.2s. As such, ElectricAuth is robust against data breaches and replay attacks because it never reuses the same challenge twice in authentications – ElectricAuth rejects replay of recorded responses to any previously used challenges, and can quickly recover from

leak/breach of either authentication model or stored challenge-response pairs by asking the user to register responses to a new set of challenges (like registering new one-time passwords).

Finally, we evaluated our prototype of ElectricAuth by means of four different evaluations, each shining light on a different facet of our research question: (1) in our user studies, we found that ElectricAuth offers accurate user verification and resists three common biometric attacks: impersonation, replay and synthesis attacks; (2) in our exploratory longitudinal study, we found that ElectricAuth's pre-trained authentication model performed stably over 21 days against various muscle conditions (fatigue, humidity, etc.) that were absent from the training data; (3) in our technical evaluation we showed that ElectricAuth, after receiving a response, can verify the user in 3ms on laptop's CPU and 35ms on a small embedded device; we also confirmed the use of depth camera as an alternative motion tracking modality (since our prototype uses IMUs); and, (4) we generated synthetic impersonator responses to test ElectricAuth's robustness against impersonation attacks at scales larger than our user studies.

4.2 Related work

This work builds on the fields of wearables, electrical muscle stimulation, and biometrics.

4.2.1 Electrical muscle stimulation

Electrical muscle stimulation (EMS) is a technique from medical rehabilitation [188] that induces involuntary movements by delivering electrical impulses to the user's muscles. This is typically achieved by non-invasive methods such as attaching pairs of electrodes to the user's skin (e.g., on top of the muscles that control finger movement, located in the forearm). Electrode pairs are typically driven using safe and medical compliant muscle stimulators [102].

The range of motion of an induced muscle contraction depends on several key factors. Even in the very first interactive use of EMS in HCI, by Kruijff et al. [106] in 2006, the po-

tential causes of EMS' intersubject-variability were discussed: "(..) stimulation level differed between users and was clearly dependent on the muscle and fat level and thickness of the arm (...)". Similarly, in PossessedHand [192], Tamaki et al. also found "(...) stimulation levels according to individual variations". In fact, researchers in the fields of muscle-biomechanics and physiology have been investigating precisely which factors drive a muscle's unique response to electrical impulses, including: the location of the electrodes [192, 167]; the electrical waveform characteristics, such as frequency and amplitude of the impulses [192, 167, 106]; the target muscle's contractility [75], i.e., the ability of muscle fibers to shorten; muscle elasticity [80, 199], i.e., the ability of the elastic tissue present in the muscle fibers to return to its original length when a tensile force is removed; muscle viscosity [51], i.e., the internal bio-lubrication of the muscle inhibits the muscle from reacting too quickly to protect against stretch injuries; the limb's mass and shape [52, 58, 146, 42, 36]; skin conductance affects non-constant current EMS devices [112, 106], bioimpedance [47, 176]; and, even nerve conduction [3, 186], i.e., the speed of nerve signal transmission. However, it is not possible to precisely determine how much each factor weighs in the final variability, as these are tied together in complex non-linear ways, and this is still an open research question in muscle physiology. More importantly, all the aforementioned factors are relevant to our proposed technique since these vary-across users. Typically, a combination of these explains the *inter-subject variability* seen in EMS-based interactive systems, which is why researchers report long periods of calibration [192, 189, 117, 120] and even specifically mention differences across users [106, 192].

Recently, researchers started to engineer interactive devices based on EMS. These tend to fall into two broad categories: (1) haptic devices that increase immersion/realism of virtual environments, and (2) interactive devices that facilitate information access via proprioception. As far as interactive devices that increase immersion, EMS has been used to render forces in mobile devices [116], virtual reality [117, 120] or augmented reality [121, 64]. As a

means of general information access, EMS has been especially used for haptic training (e.g., learning a musical instrument [192], operating a tool the user is not familiar with [119]) or eyes-free communication (e.g., communicating walking directions via leg stimulation [189], communicating a state of a variable via wrist movements [118]).

Unlike these interactive systems that use EMS as a form of force feedback or as an information channel, we explore EMS in a new direction: leveraging user's unique muscular responses to EMS as a form of active biometric authentication.

4.2.2 Biometric authentication

Biometric authentication verifies an individual by their unique biological characteristics. To verify a user's identity, a biometric authentication system compares a previously stored biometric key from a particular user to incoming, typically real-time, biometric data of the user wishing to authenticate. Compared to traditional password or PIN based methods, biometric authentication offers significantly better usability by not requiring the user to memorize passwords or PINs.

Existing biometric systems can be categorized into two types: passive and active biometrics.

Passive biometrics. Passive biometrics rely on physiological characteristics that naturally occur in users, which can be either static or dynamic. Static data, e.g., fingerprints [81], handprints [72], facial and eye features [128, 206, 152, 4], is often used for authentication. Biometrics based on dynamic data recognize patterns that vary over time, e.g., heartbeats [87], gait [190], mouse movements [94], keystrokes [194], speech features [19], body movements [157, 145], pulse-response [166] and bioimpedance [82, 176]. Compared to static data, these display greater complexity and are harder to model.

Passive biometrics are vulnerable to data thefts and replay attacks as reported by numerous incidents and studies [150, 208, 214, 207, 29, 100, 60]. This is because the identity

(also known as "key") associated with each user is physically "hard-coded" and then used *repeatedly* for all authentications. Thus after a key has been compromised (e.g., stolen from a database), an adversary can bypass authentication until the key is replaced. Finally, there is a small number of available biological traits per user that act as suitable keys, e.g., once all ten fingerprints are compromised, this user can never again rely on fingerprint authentication.

Active biometrics via challenge-response. Active biometrics leverage a user's physiological response to a given stimulus (also known as "challenge") injected by the interactive device. The assumption is that each user's response to a given challenge is unique. Thus, each *challenge-response* is effectively a biometric password. Examples of challenge-response biometrics include leveraging: the palm's response to vibrations [109], reflexive eye behaviors in response to visual stimuli [182], or even EEG responses [114]. These systems authenticate implicitly so the user does not need to consciously follow the challenge, e.g., the palm vibrates and the user is authenticated [109].

Compared to passive biometrics, active systems are more robust against data thefts and replay attacks. This is because each user can potentially generate many challenges, each triggering a different response. The system uses a new challenge in each authentication session, preventing attackers from using previously observed responses to breach it.

Lastly, while many challenge-response authentication systems leverage the user's movement (e.g., gaze [165] or wrist shakes [155]), these require explicit action from the user. Unlike these, our novel exploration of EMS-based authentication provides the advantages of movement-based challenge-response while automatically delivering the challenge and eliciting the user's *involuntary* response.

4.3 Implementation

To help readers replicate our design, we provide the necessary technical details. Furthermore, to accelerate replication, we provide source code and training scripts[1]. Here, we describe in detail the prototype we implemented for our user studies, which is based on sensing the user's movements using inertial measurement units (IMUs). However, this is just one possible configuration for our concept. As depicted in Figure 4.1, other tracking systems, such as optical tracking [204, 147], are likely feasible alternatives.

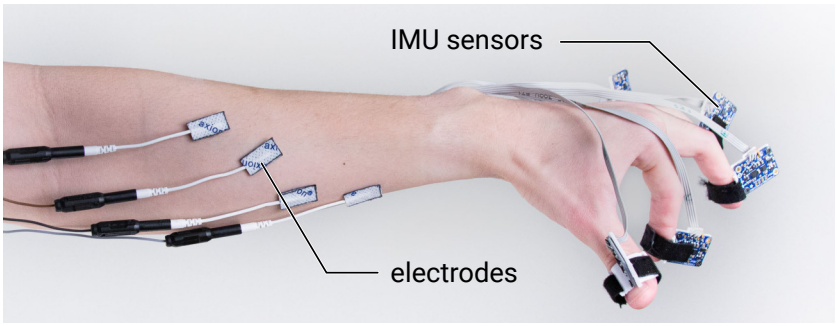

Figure 4.2: IMU-based version of our EMS authentication system, which we used for our user studies.

4.3.1 System overview

ElectricAuth consists of three components: (1) a medically-compliant **EMS device** that delivers EMS challenges to the user, (2) a **motion sensor** that captures the actuated limb's movements, such as IMUs or depth cameras, and (3) a **trained machine learning model** that classifies the user's movements and performs authentication. Figure 4.2 depicts one concrete implementation of our system using EMS and IMUs attached to a user's forearm, which we used for our user studies.

1. `http://sandlab.cs.uchicago.edu/electricauth`

1. EMS hardware.

EMS stimulator: For delivering EMS impulses we use the Hasomed Rehastim, a medical compliant device with eight individually controllable channels. This device has often been used in interactive systems based on EMS [122, 120, 121]. To control the EMS stimulation, our software sends serial commands via USB using the Hasomed's Science Protocol [76]. These impulses have a latency of <1ms.

Customized EMS sleeve: As with any device based on EMS, we start by calibrating the electrode placement for each user at her registration session. Our calibration aims at targeting four muscles on the user's forearm that actuate finger and wrist rotation. At the anterior forearm we stimulate two muscle groups: (1) primarily the *flexor carpi radialis* and partially the *flexor digitorum profundus*; and, (2) the *flexor pollicus longus*. At the posterior forearm we stimulate two muscle groups: (1) primarily the *extensor digitorum* and partially the *extensor digiti minimi, extensor pollicis brevis & longus*; and, (2) the *extensor indicis*. As is typical with EMS-based systems, these electrode positions are adjusted for each user during the registration session to ensure comfort. Because each user has a different muscular anatomy and body shape, the resulting electrode locations are different across different users.

After calibration, the resulting electrode layout for a particular user is fixed by making an EMS-electrode sleeve (fabric with electrodes stitched to it) that this user wears any time they use ElectricAuth. Moreover, the sleeve becomes part of each user's own challenge definition, i.e., an attacker trying to impersonate a particular user will require obtaining or copying the user's sleeve, which we later validate in our studies by actually providing the impersonators with the EMS sleeves of the legitimate users.

EMS parameters: Our EMS stimuli on all electrode locations are the same: single-shot square-impulses with an intensity of $10mA$ and a pulse-width of $200\mu s$. We chose this configuration for two reasons. First, we configured EMS impulses to generate small and subtle finger movements rather than large conspicuous movements typical of most existing

EMS research, because this enables more practical authentication scenarios. While these smaller movements are harder to recognize, our results suggest that our authentication model can accurately track these (see Section 4.7). Second, we opted to make all impulses uniform to shine light in the fact that intersubject variability in EMS arises from factors external to EMS waveform characteristics.

Our EMS challenges are constructed by sequencing these standardized pulses to one of the four channels the user's forearm is connected to. For instance, one can construct a challenge with a sequence of six impulses, each followed by a resting period. We detail the engineering of our pulse sequences in Section 4.3.2.

2. Motion sensing.

We utilized a set of five 9-DOF inertial measurement units, attached to the fingers via a 3D printed ring (NXP Precision 9DoF, comprised of the FXOS8700 3-Axis accelerometer and magnetometer and the FXAS21002 3-axis gyroscope). These sample the fingers' acceleration and rotation at 50Hz (post-sample interpolated to 100Hz) with a precision of $\pm$4g at 14-bit for acceleration and $\pm 250^{\circ}$/s at 16-bit for rotations; note that we do not use the magnetometer. These IMUs are sampled by a ATSAMD21G18 ARM Cortex M0 48 MHz processor, via a TCA9548A I2C Multiplexer. Finally, our sensing board relays the IMU data via serial over USB to our software.

While attaching IMUs to each finger has been shown to be a reliable way to capture hand pose [84, 55], we believe many alternative tracking systems are possible to realize EMS-based authentication, such as depth cameras [181, 204], RGB cameras [34, 180], and others [98]. We provide a short evaluation that confirmed the use of depth cameras as an alternative tracking system in Section 4.9.

3. Authentication software and pipeline.

The software component of ElectricAuth, written in Python, handles all the interactions between EMS device, motion sensing, model training and real-time authentication. The

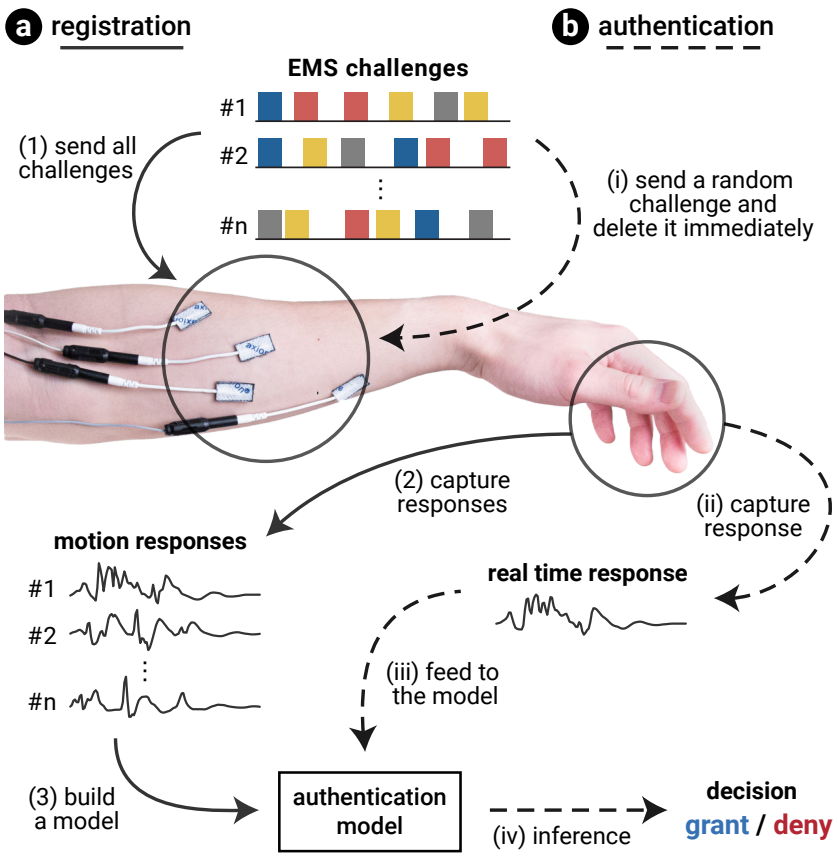

Figure 4.3: Interactive pipeline for the registration (registering a new user) and authentication phase (interactive use in runtime). User response can be caputured using a motion capturing device, e.g., IMUs and cameras (not shown). In this system, the EMS device and electrodes are wearable; the motion capturing device is either wearable or placed near the user; while the authentication model can be remote.

pipeline of ElectricAuth, which is depicted in Figure 4.3, is comprised of two phases: (a) **registration** and (b) **authentication**.

In the **registration** phase, marked by solid lines in Figure 4.3, registering a new user (after calibration) is as follows: (1) a set of n EMS challenges are sent one at a time; (2) the user's movements in response to each challenge are recorded; (3) these responses are used to train a machine learning-based authentication model for this user. The number of challenge-response recorded per user is the primary factor that dictates the total time the system needs for registering a single user (we detail this in Section 4.4).

In the **authentication** phase, marked by dashed lines in Figure 4.3, verifying a user's identity in run-time is as follows: (i) one random EMS challenge belong to the claimed identity is chosen, deleted immediately from the database, and sent to the user via EMS; (ii) the user's movements in response to the challenge are recorded; (iii) the motion responses are fed into the trained authentication model of the claimed identity; (iv) the system determines whether this user is legitimate (i.e., being the claimed identity) or not.

4.3.2 Engineering EMS-based challenges

As our system is the first that explores EMS for authentication, we dedicated a significant part of our exploration in understanding how to increase the challenge pool using EMS; a large challenge pool is what makes a challenge-response based authentication system robust against data breaches and replay attacks. Naively, one can stimulate the user's muscles with individually configurable pulses; however, this (1) requires more calibration time and (2) does not reveal the mechanisms that explain these individual responses. Therefore, we kept purposely all EMS impulses uniform for all users of our system; this grants us more confidence in interpreting the unique responses as originating from the physiological differences between users. Yet, this introduces a challenge when it comes to diversifying the challenge pool.

One straightforward solution (adopted by many existing works on challenge-response biometrics [109, 182, 114]) is to sequence stimuli but separate them by a fixed time gap. If we were to adopt this as well, the maximum number of EMS challenges would be S^L, where S is the number of unique stimuli in the system and L is the number of stimuli in each challenge. For example, a sequence of six EMS impulses over four possible EMS channels, with a fixed rest period between each impulse, results in $4^6 = 4,096$ challenges. We were interested in whether we could dramatically surpass this approach.

To significantly increase our challenge pool, we explored a rather unused property of human muscles that causes them to respond differently to EMS depending on their current

state of contraction. We call this *temporal dependency*.

Temporal dependency. We empirically found, in our preliminary pilots, that a subject's response to an EMS stimulus is affected by the previous stimulus in the same challenge, and the impact depends on the time gap between them (represented as τ).

Figure 4.4 shows two example traces of a finger's acceleration when we stimulate the user's muscles with a sequence of two stimuli (A and B) but vary the time gap between A and B (i.e., $\tau = 0.1s$ and $\tau = 0.17s$). The measured acceleration displays different characteristics when we vary τ. The strongest candidate for a physiological explanation is that muscle contractility and elasticity vary with muscle length [198, 70], and the response to a stimulus depends on the muscle lengths at the time of stimulation. Thus, depending on the gap between A and B, the subject's unique contractility [75] and elasticity [80, 199] will lead to different responses.

The use of temporal dependency affords a large EMS challenge pool by varying the time gaps between consecutive stimuli. Assuming they all produce distinguishable responses, the number of unique challenges (of length L) is upper bounded by $S^L \cdot T^{L-1}$, where T is the number of distinct time gaps. For our ElectricAuth prototype, we utilize $S = 4$ EMS channels and $T = 7$ different time gaps ($\tau = \frac{1}{30}$s, $\frac{2}{30}$s, ..., $\frac{7}{30}$s), which in early pilots we found to lead to sufficiently different movement outcomes. The maximum number of unique challenges is 112 (L=2), 87,808 (L=4) or 68,841,472 (68M) (L=6), compared to 16 (L=2), 256 (L=4) or 4,096 (L=6) when we do not vary the time gap.

Further increasing the challenge pool. Encoding longer challenges is another way to expand the challenge pool. With $S = 4$ stimulus locations and $T = 7$ time gaps, sending $L = 8$ pulses (<2s) increases the pool size to $53,971,714,048$ ($4^8 \times 7^7$). Also it is possible to add more electrodes or customize EMS impulses to further diversify the pool.

Checking for uniqueness. Ideally, every challenge-response authentication in the pool is unique. However, in practice this might not be the case given the granularity and sensitivity

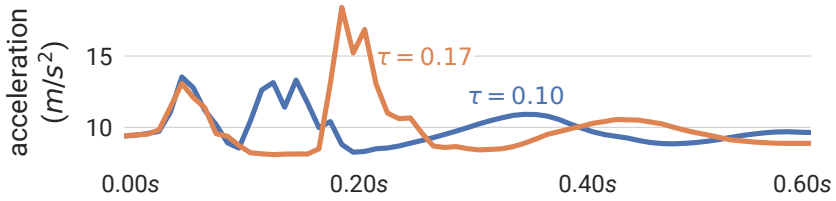

Figure 4.4: An example of how a response changes when the time gap between two EMS stimuli varies: we vary the time gap from 0.1s (blue curve) to 0.17s (orange curve).

of motion sensors. To enforce uniqueness, ElectricAuth can apply a verification step during user registration. Specifically, after generating new challenges for a user at the registration phase, it collects the corresponding responses and checks the similarity across these responses and previously registered responses (e.g., computing the mean square error (MSE) between raw responses). If a new challenge is identified as a previously registered challenge, this new challenge is removed.

4.4 User authentication model

We now present the design of ElectricAuth's user authentication model. ElectricAuth requires a trained authentication model per legitimate user, which is used to verify whether a test subject is indeed that user. To do so, the model takes as input the response to a given challenge designed for the legitimate user, and outputs whether the test subject is legitimate. Our authentication model was designed with two objectives in mind: (1) minimize the amount of samples collected from the user (i.e., reducing registration overhead) and (2) resist common attacks (e.g., impersonation and replay attacks) and data breaches.

4.4.1 Overview

Initially, we explored implementing our model using specific features of the user's IMU data in response to particular EMS challenges (so called feature-engineering). However, we quickly

realized a major downside of this approach: as the response data we capture in real-time from the IMUs is complex (thirty concurrent data streams: 5×3 axes of acceleration and 5×3 axes of rotation), simple feature extraction might not capture the full expressivity of the data. Therefore, after experimenting with this approach, we turned to neural network based models.

We implemented a robust authentication model that, for each registered user, integrates two deep neural network (DNN) models to resist both impersonation and replay attacks. Specifically, authentication starts with **(1) an unsupervised anomaly detector**, which verifies whether a response was produced by the user the model belongs to (i.e., the legitimate user); this step prevents *impersonation attacks*, in which a different user attempts to gain identity of the legitimate user. If a response passes the anomaly detector, it then enters **(2) a challenge classifier**, which detects and rejects *replay attacks* by verifying whether the response is the reaction to the challenge used in the current authentication session.

Both models are trained using only the challenge-response pairs of this legitimate user collected during registration. When the user (re)registers a new set of challenges, we retrain both models from scratch using the new data. This also enables ElectricAuth to recover from data and model breaches.

4.4.2 Detailed model design

1. Verifying user via unsupervised anomaly detection.

We implement user verification as unsupervised anomaly detection [31], where the detection model is trained on *only* the legitimate user's responses collected during registration. At run time, the model verifies whether an input response was likely originated by the legitimate user. This anomaly-based detector leverages the fact that responses from other users will display characteristics different from those of the legitimate user. Thus the model is designed to produce normal output when the input response comes from its legitimate user, but

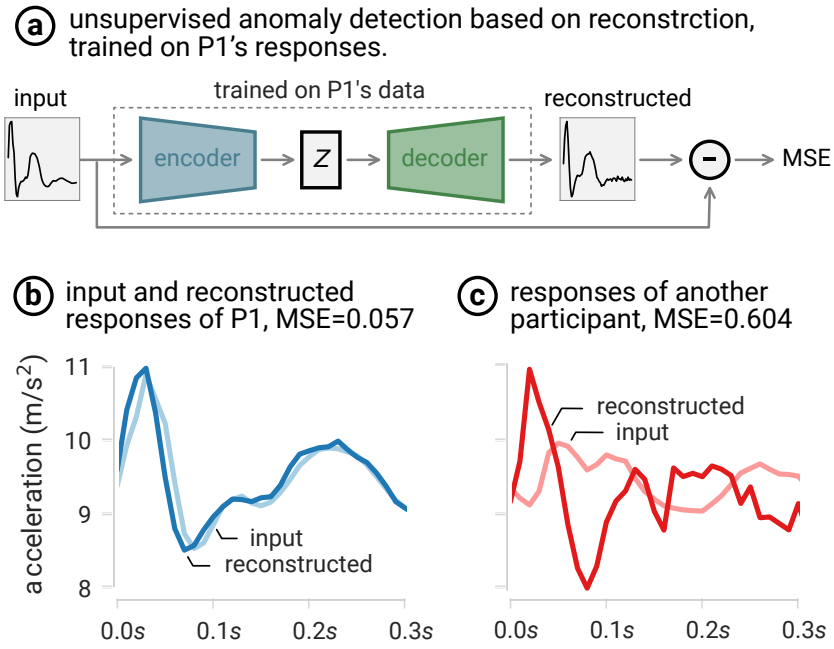

Figure 4.5: Authentication starts with an anomaly detection, which verifies if a response came from the legitimate user that the model belongs to (P1 in this example). (a) The anomaly score is the MSE of the input and model-reconstructed responses. We illustrate how our anomaly detector correctly: (b) identifies P1 (legitimate user) with a low MSE and (c) rejects P2 (impersonator) with a high MSE.

abnormal output when the input comes from any other user. This design prioritizes generality as the model is trained *without* requiring knowledge on other users.

For our prototype, we apply a reconstruction error based anomaly detection system [158]. Specifically, we use variational autoencoder (VAE) [56], a DNN architecture well-known for *automatically* capturing complex patterns in target data. As shown in Figure 4.5, each VAE starts from an encoder to extract latent features from each input response, followed by a decoder to reconstruct the response from these features. It then computes the mean squared error (MSE) between the input and reconstructed responses, and outputs it as the anomaly score of the input. Ideally, the anomaly score will be low when the input response comes from the legitimate user and high when the input comes from a different user. Thus, the system can configure a threshold on the anomaly score, where a value larger than the threshold indicates the test subject is not the legitimate user (i.e., the user verification

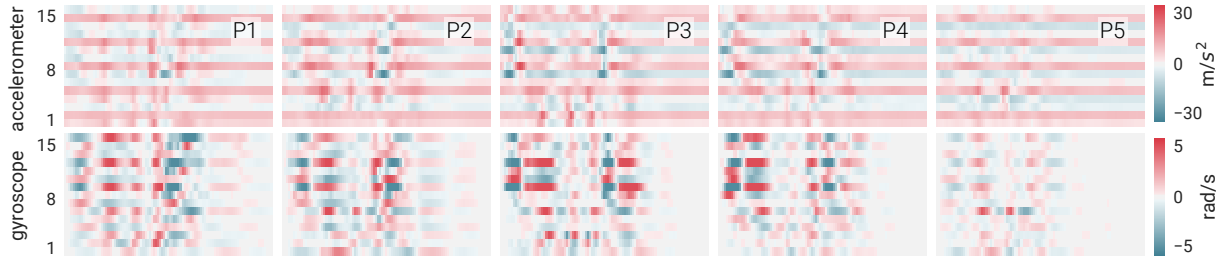

Figure 4.6: Sample responses of a P1's challenge (with L=6 impulses) and impersonators' responses (P2, P3, P4 and P5) to the same challenge. Each row is a sensor channel and each column is one data sample. Here we show one second of responses. When tested on P1's anomaly detection model, the corresponding anomaly scores for P1-5 are 0.70, 5.03, 9.44, 8.81 and 7.50, respectively. In this case, the model can easily detect impersonators.

fails). In ElectricAuth, we choose the threshold during model training to reach a desired false rejection rate (i.e., the probability that the model rejects the legitimate user's input responses).

For our implementation, we train our VAE using each legitimate user's responses to all the chosen challenges collected during registration. The data aggregation (across challenges) creates a reasonable amount of data to train the VAE successfully. We consider a common VAE architecture [62], where the encoder contains two dense layers of 400 and 200 neurons, respectively, and the decoder contains two dense layers of 400 and 3600 neurons, respectively, to match the input size.

To illustrate the effectiveness of our model, we plot in Figure 4.5b-c the input and reconstructed responses of a legitimate user (here, $P1$ of our user study) and a different participant $P2$ (also from our user study), respectively, using the model trained for $P1$. For the sake of visual clarity, we only plot the responses from only one accelerometer axis. Both responses are not used for model training. We see that $P1$'s response is well-approximated by the model-reconstructed response; in fact, with a very low MSE of 0.057. On the other hand, $P2$'s response (when tested on $P1$'s model), produces a large MSE of 0.604, around 10-fold higher than the MSE of the legitimate user ($P1$).

Figure 4.6 shows the responses (collected by the IMUs) of five subjects (P1-5) to a challenge designed for P1 (the legitimate user in this case). When tested on P1's anomaly detection model, the anomaly scores for these responses are 0.70, 5.03, 9.44, 8.81 and 7.50, respectively. Thus P1's model easily rejects P2-5 as impersonators.

2. Verifying challenge via challenge classifier.

Next in the authentication pipeline, ElectricAuth verifies whether the input response matches the challenge used in the current session. As mentioned earlier, this is designed to resist replay attacks, where an attacker, after obtaining a copy of the legitimate user's responses to previously used challenges, replays one of these responses to bypass authentication.

ElectricAuth implements challenge verification by training a classifier: given an input response, it determines the corresponding challenge. If the identified challenge matches the challenge used in the current authentication session, authentication is granted; otherwise, rejected. Moreover, the classifier also detects when the input response comes from any challenge *not* used to train the classifier, because the classifier will output a low confidence score.

Our implementation uses a Convolutional Neural Network (CNN) for this classification task [151]. It contains four convolutional and two dense layers. Each convolutional layer employs 64 filters sized 5 to extract information from the input. The information is then fed into the two dense layers containing 128 and 112 neurons, respectively. At the end, a softmax function is applied to the output to produce a probability distribution over potential challenges. We train our CNN using the *same* training data used in training the above anomaly detector, except that we now label each response by its corresponding challenge.

4.5 Contributions, benefits and limitations

Our main contribution is that we explore EMS in a new direction, i.e., leveraging EMS's *intersubject variability* as a novel modality for active biometric authentication.

ElectricAuth inherits the advantages of both biometric and password authentication: (1) As with any biometric authentication device, ElectricAuth does not require memorization or cognitive effort – this makes our system suited for a wide range of users, including those with cognitive impairments; (2) Unlike passive biometrics (such as fingerprints), ElectricAuth's challenge-response structure makes it secure against data breaches and replay attacks; Lastly, (3) ElectricAuth leverages temporal dependency to create a very large set of challenges – in this way, ElectricAuth can dispose a challenge anytime like a one-time password.

On the flipside, ElectricAuth is subject to several limitations: (1) Like any solution based on electrical muscle stimulation, ElectricAuth requires some initial adjustments of the electrodes (during registration) that ensure pain-free operation, and also periodic re-gelling of adhesive electrodes to prevent electrodes from fatigue and eventually affecting the authentication accuracy; (2) ElectricAuth requires user's hands to be free while authenticating, making it more suitable for hands-free applications; (3) As with existing biometric devices, ElectricAuth requires initial registration. Specifically, each challenge needs to be registered in advance; Lastly, (4) while a single EMS impulse can be very short (e.g., 200μs) to achieve very high accuracy, we expanded our sequence to 1.2 seconds of muscle stimulation, as such ElectricAuth takes ∼1300ms to authenticate a user in runtime. While this is certainly fast enough for most applications, it is longer than some passive approaches, such as fingerprint recognition.

4.6 Overview of evaluations

We evaluated our concept of using EMS for authentication by means of four different evaluations, each shining light on a different facet of our research question. All studies were approved by our Institutional Review Board (IRB no. omitted for anonymity). To aid the reader in understanding the different validations we performed, we present an overview of our evaluations with a preview of their respective results:

I. User studies. We evaluated the feasibility of EMS as an active biometric with three experiments and 13 participants. We found that that ElectricAuth resists three common attacks: (1) impersonations attacks, in which participants played impersonators against each legitimate user (attack success rate or false acceptance rate: 0.17%); (2) replay attacks, in which participants mimic the movements of the legitimate user from videos (success rate: 0.00%), or replay a perfect record of response to any used challenges directly into the IMUs (success rate: 0.00%); and, (3) synthesis attacks, in which we synthesized data from the participants' data to attack their authentication models (success rate: 0.2-2.5%).

II. Exploratory longitudinal study. We conducted a longitudinal study over 24 days and for two participants, to examine ElectricAuth's authentication model over time and against various muscle conditions (fatigue, humidity, etc.). We found that an authentication model, trained using the first three days and tested over the next 21 days, performed very stable over time and on muscle conditions unseen during training (false rejection rate ≈2%, with a SD around 3%).

III. Technical evaluation. A technical in which we measured ElectricAuth's latency, model training time, and the feasibility of using depth cameras as an alternative motion tracking modality.

IV. Testing model robustness at scale, using synthetic data. We applied a data-driven approach to better understand how our system might scale to larger numbers of users that is simply impractical to test in the laboratory. To realize this, we employed the user study data to train deep generative models that produce synthetic impersonator responses, and used these data to further evaluate ElectricAuth. We found that, across all the data-driven experiments and for all legitimate users, no generated response was accepted by ElectricAuth (attack success rate: 0).

4.7 User studies

To evaluate the feasibility of EMS as an active biometric we conducted a user study, with three sub-experiments, which allowed us to understand: **(1) authentication accuracy**, in which we evaluated the accuracy of our system; **(2) impersonation attack**, in which we evaluated its robustness against attackers trying to impersonate legitimate users; and, **(3) replay attack and synthesis attack**, in which we evaluated its robustness against three replay attacks (human mimicry, record-replay, breach-replay) and one online synthesis attack.

In total, we collected 70,000+ wrist and finger movements as responses to EMS challenges (stimulation patterns). We analyzed the performance of ElectricAuth using four standard metrics, typically employed to assess a system's authentication performance: **(1) False rejection rate (FRR)**, which measures how often a legitimate user is mistakenly denied, at a specific threshold; **(2) False acceptance rate (FAR)**, which measures how often an illegitimate user is mistakenly authorized, at a specific threshold; **(3) Equal error rate (EER)**, the rate at which the measured FRR equals the measured FAR for a certain threshold; and, **(4) Receiver operator characteristic curve (ROC curve)**, which describes the relationship between FRR and FAR as a curve, by varying its threshold.

4.7.1 Experiment#1: authentication accuracy

The goal of our first study was to understand the authentication accuracy of our system. Furthermore, as we were interested in the impact of the length of the EMS challenges on its performance, we recorded participants' movements to three sets of challenges, based on their number of impulses $L = 1, 2, 6$ (referred to as length-1, -2, and -6 challenges, respectively). For each challenge set we stimulated participants' forearms and recorded finger movements using IMUs.

Participants. We recruited 13 participants from our institution (mean age= 24 years, SD= 3 years; mean weight= 66.3 kg, SD= 13.3 kg; mean height= 171.2 cm, SD= 8.2 cm; 7 females, 6 males). Participants were compensated with 50 USD for their time.

Apparatus. Participants wore our system on their left forearm. This included the EMS and IMU components, which were fitted by an experimenter. To ensure participants' comfort with EMS, we calibrated it so that all electrode channels operated pain-free. To ensure that all target muscles were correctly stimulated (see Implementation for details), we gradually increased the intensity during calibration, following calibration process similar to [23]. If a participant felt any discomfort before reaching the target intensity, we moved to another electrode position. To minimize fatigue, participants rested their elbow on a resting base.

After calibration, we recorded each participant's exact electrode locations by making a custom sleeve with marked positions. These 13 sleeves were later used in Experiment #2, where we examined impersonation attacks (i.e., each impersonator wore the sleeve of a legitimate user to attack our authentication system).

During the study, participants did not receive any specific instruction, since we wanted them to react naturally to the EMS impulses.

EMS challenges. The EMS challenges in our study were configured as previously described, i.e., a challenge was comprised of a sequence of single-shot square-impulses with an intensity of $10mA$ and a pulse-width of $200\mu s$; these sequences were of length-1, -2 or -6. In between each pair of impulses we included a time gap. Each gap was one of seven possible durations ($\frac{1}{30}$s, $\frac{2}{30}$s, ..., $\frac{6}{30}$s, $\frac{7}{30}$s); thus, the recording duration of a length-1, -2, and -6 challenges were 0.6s, 0.8s and 1.2s, respectively. While length-1 challenges were collected in this experiment, these were only used for an analysis in Experiment#2 (anomaly detector performance).

Procedure. To test whether ElectricAuth correctly authenticates our 13 participants, we first registered each participant. Our system did this automatically: (1) a participant feels

an EMS challenge, (2) their forearm muscles react involuntarily, and (3) our system records the response. We repeated this process 10 times per challenge. These ten responses were shuffled to remove potential sequence effects. These responses were then randomly divided into a training set (eight responses) and a testing set (two responses). Then, our system took these eight responses (for all challenges) and trained the anomaly detector and challenge classifier for each participant. As cross-validation, we repeated this process to produce 10 authentication models per participant and reported the average test results of these models in all our subsequent experiments.

For length-1 and -2 challenges, we tested the full set of challenges (a total of four for length-1 and 112 for length-2). For length-6 challenges, we were forced to test only a subset, since the full set includes $68,841,472$ challenges, which would be fatiguing for participants. Therefore, we randomly chose 115 challenges from the full set.

In total, each participant performed 2310 trials: 40 trials of the four length-1 challenges (10 repetitions); 1120 trials of the 112 length-2 challenges (10 repetitions); and, 1150 trials of the 115 length-6 challenges subset (10 repetitions).

Results: overall authentication accuracy. We first examine the accuracy of the end-to-end authentication model, which depends on the accuracy of both the anomaly detection model and the challenge classification model. We defined overall accuracy as the probability that a legitimate response successfully passed the two-step authentication. Note that the accuracy is dependent on the anomaly threshold used by ElectricAuth's authentication model. During model training, we configured the threshold to reach a planned false rejection rate (FRR). Note that the threshold is determined using just the training data (without the knowledge of any run-time testing data). Ideally, the run-time measured FRR (i.e., 1−accuracy) should equal to the planned FRR.

For each of the 13 registered participants, Table 4.1 summarizes the measured FRR (i.e., = 1−accuracy) aggregating the results across all 115 challenges (of length 6). Here we

participant	planned FRR	
	2%	*5%*
P1	2.3	6.1
P2	2.1	5.5
P3	2.1	4.9
P4	1.7	5.4
P5	2.6	4.8
P6	2.7	6.2
P7	2.5	5.0
P8	2.3	5.5
P9	3.2	5.8
P10	2.2	5.0
P11	2.3	5.0
P12	2.8	5.5
P13	2.6	5.4
AVG(SD)	2.4(0.4)	5.4(0.4)

Table 4.1: The measured false rejection rate (FRR, %) for all registered participants (P1-P13) closely matched the planned FRR. The measured FRR was calculated for each participant using their test responses to 115 length-6 challenges.

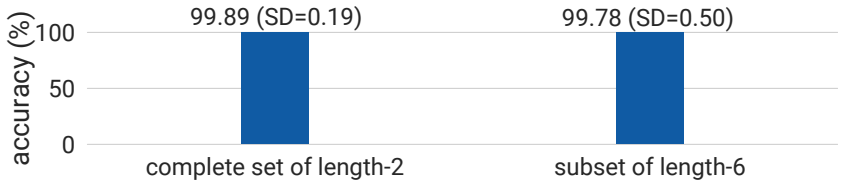

Figure 4.7: ElectricAuth's challenge classification accuracy for length-2 and length-6 challenges.

reported the results for planned FRR of 2% and 5%. We see that the measured FRR closely matched the planned FRR. Across all the participants, the mean measured FRR is 2.4% (SD of 0.4%) and 5.4% (SD of 0.4%), respectively, matching the two planned FRR values (2% and 5%).

Results: challenge classification accuracy. Digging deeper into the accuracy of our system, we turn to evaluate the accuracy of challenge classification model (as it is the main component protecting against replay attacks). Our accuracy findings are depicted in Figure 4.7. For length-2 challenges (complete set, i.e., 112 of them) the average accuracy is 99.89% (SD=0.19% across users). And for length-6 subset of challenges we found an accuracy of 99.78% (SD=0.50%). These results also show that the challenges (full set of length-2, subset of length-6) are unique across each other.

4.7.2 Experiment#2: impersonation attacks

In this user study, we measured our system's ability to resist impersonation attacks.

Participants. For this study, we invited all 13 participants from Study#1. Participants were briefed that they would play an attacker trying to impersonate other participants. Participants were compensated with 50 USD for their time.

Procedure & apparatus. For each target participant, we applied their customized challenges (used in Experiment #1) to the other 12 participants (as impersonators) and collected their responses. Impersonators were asked to wear the sleeve fabricated for each target participant in Experiment #1. These sleeves grant the impersonator with the exact electrode positions of the legitimate user. We also tested cases where impersonators wear their own sleeves and other participants' sleeves and found that wearing the target participant's sleeve leads to the most effective attack; thus we focused on it.

In total, each participant performed 3240 trials: 480 trials of the length-1 challenges (10 repetitions per challenge, impersonated 12 other participants); 2760 trials of the length-6 challenges subset (2 repetitions per challenge, impersonated 12 other participants).

Impersonating someone else by using their electrode placement does not guarantee comfortable use, i.e., we did not adjust electrodes to preserve the legitimate participant's placement. While no participant felt uncomfortable with length-1 challenges, there was some discomfort on a few length-6 trials (3.8% of the total); anytime a participant voiced discomfort, we stopped the stimulation and discarded this trial.

Results: performance of anomaly detector. To deepen our understanding of intersubject variability and the anomaly detection model performance, we first compared the responses to a single stimulus (or length-1 challenge), submitted by each target participant in Experiment#1 and the 12 impersonators in this experiment. We fed these responses to the target participant's anomaly detection model and recorded their anomaly scores. For the sake of visual clarity, we normalized these anomaly score values by the target participant's

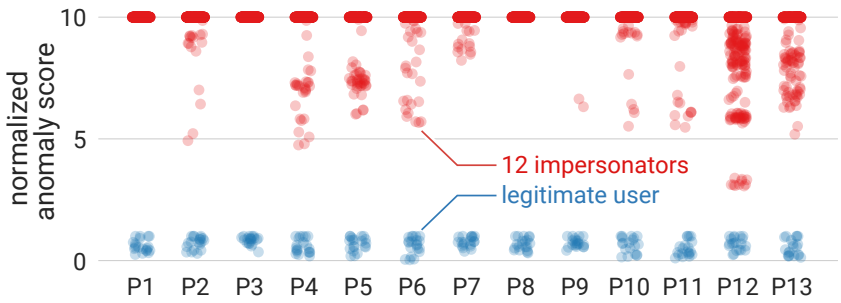

Figure 4.8: Normalized reconstruction error for the responses to each participant's length-1 challenges, submitted by both the legitimate user and the 12 impersonators. For visual clarity, we capped the value at 10.

average anomaly score value (see Figure 4.8).

We found that our anomaly detector for each participant is well-trained and can distinguish impersonators from the legitimate participant. This is clear as Figure 4.8 depicts a large separation between the legitimate participant and the impersonators. It also confirms EMS intersubject variability.

Results: robustness against impersonation attack. We examined the end-to-end success rate of impersonation attacks against each participant, using the attack data collected on length-1 challenges (complete set) and length-6 challenges (the 115 subset).

Figure 4.9(a) depicts the false acceptance rate (aggregated across 13 participants' models since they are consistent) against length-1 and length-6 challenges, for the planned FRR of 2% and 5%, respectively. With length-1 challenges (4 challenges), the impersonation attack failed. With length-6 challenges, the attack exhibited a very low success rate, only 0.83% (SD=1.14%) at planned FRR=2% and 0.17% (SD=0.32%) at planned FRR=5%. Again this suggests that our system is robust against impersonation attacks. Figure 4.9(b) shows the ROC curves under impersonation attacks with length-6 challenges, where ElectricAuth achieves an EER of 1.31%.

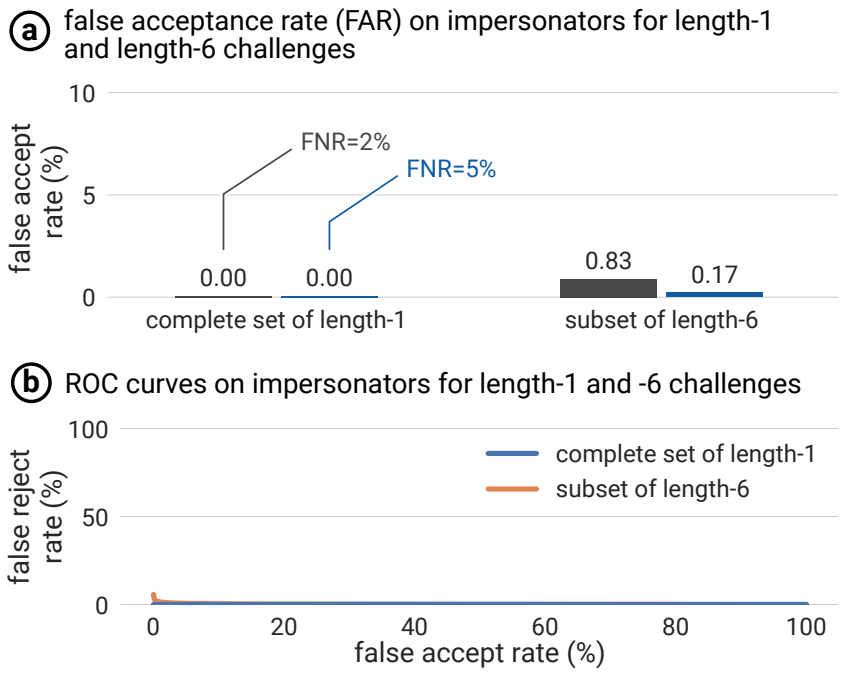

Figure 4.9: ElectricAuth's robustness against impersonation attacks.

4.7.3 Experiment#3: replay and synthesis attacks

In this user study, we measured ElectricAuth's robustness against replay attacks and synthesis attacks, both trying to engineer a response to bypass authentication after obtaining some knowledge on the legitimate user's responses.

We considered three replay attacks, and one synthesis attack, ranging in increased attack complexity:

- **(1) human mimicry**, where the attacker video-tapes and studies a participant's responses and then physically mimics the responses without wearing any EMS;
- **(2) record-replay**, where the attacker compromises the IMUs so that they can *perfectly* record the target participant's response to challenges in previous authentication sessions, then during a new authentication session (i.e., a new challenge), the attacker selects a previous recorded response and directly feeds it to the IMUs;
- **(3) breach-replay**, where the attacker breaches the database or the model to recover

stored challenge-response data, and feeds one response to the IMU's circuit; here ElectricAuth reacts to the breach by asking users to re-register using new challenges and retraining the models;

- **(4) online synthesis**, where the attacker compromises both EMS and IMUs to record both the challenge and the response in previous sessions; then at run-time, the attacker searches through these records and attempts to synthesize and submit in *real-time* an engineered response to the current challenge. For these attacks, we evaluated ElectricAuth using the false acceptance rate (FAR) and the ROC curve.

Participants. We recruited five participants to perform the human mimicry attack: three from our previous study (chosen at random) and two new participants from our local institution (ages: 25 & 22 years old; weights: 55 & 99 kg; heights: 177 & 180 cm; one female and one male). Participants were compensated with 50 USD for their time.

Procedure. In the **human mimicry attack**, we asked participants to study 23 videos of finger movements of a target participant. Each video was a recording of one single response to a length-6 challenge. Participants were allowed to study these videos as many times as they intended and in slow-motion (recorded at 240 fps, with clear and unobstructed view of the finger movements). Once confident and ready, participants were asked to mimic these finger movements while wearing only the IMU component of our system, in their best attempt to impersonate the target participant. Furthermore, as reference, we also asked the target participant that had partaken in Experiment#1 to self-mimic 23 of his own EMS responses after observing and studying them.

Results: robustness against human mimicry. We found that none of the study participants was able to fool our system by mimicking the target participant's responses. Note that these participants were allowed to view the videos in slow motion and as many times as they want. The FAR was 0 for a FRR $\geq 2\%$. This confirms our intuition that the EMS movements are indeed involuntary and incredibly hard to voluntarily replicate.

Results: robustness against record-replay attack. For this we utilized data from Experiment#1. Even assuming perfect recording on the side of the attacker (i.e., their recording channel has access to IMUs without any noise or sample rate issues), we found our system to be robust against these attacks. In particular, for length-6 challenges, the FAR (against any of the 115 challenges) was less than 0.0014% across all 13 participants when FRR $\geq$ 2%. This FAR is significantly smaller than the challenge misclassification rate of our authentication model (0.2%, see Experiment#1).

Results: robustness against breach-replay attack. Again we utilized data from Experiment#1. For each participant, we randomly split the 115 challenges (and their responses) into two equal sets (A and B). We assume that the attacker, via data breach, obtains the dataset A and uses them to launch replay attacks against ElectricAuth. At the same time, ElectricAuth reacts to the data breach by asking users to re-register via a set of new challenges (i.e., dataset B) and retraining the authentication models using dataset B. Like the above, we found our system to be robust against these replay attacks – the FAR was less than 0.0098% when FRR $\geq$ 2%. Moreover, both the anomaly detector and challenge classifier components in the model were able to reject the attack responses.

Results: robustness against online synthesis. We evaluated the success rate of an online **synthesis** attack, using the data from Experiment#1. We assume the attacker has access to the EMS and IMUs without sample rate or noise issues, which is in itself very unlikely. The idea behind a synthesis attack is that the adversary records both challenges and their responses, and segments these into chunks, as in "this impulse at electrode 1, moves this finger by this much", and so forth. We referred to this approach as the simple synthesis attack. A more advanced attack would capture the impact of temporal dependency by segmenting responses into per pair-stimuli chunks, as in "these two impulses at electrodes 1 and 2, move these fingers by this much"). After segmenting the responses, the attacker will observe each incoming impulse of a new challenge and inject a response into the IMUs in

real-time. Note that even assuming best hardware and knowledge, assembling this response will always have some latency.

Figure 4.10(a) plots the FAR of online synthesis attacks considering three latency values, assuming the attacker has observed R=50 challenge-response pairs and the planned FRR is set to 5%. Even under the extreme attack case (zero latency, which is physically impossible), the attack success rate is low (i.e., FAR=2.2% and 7.5% for simple and advanced attacks, respectively). When the synthesis latency reaches 20ms, which still depicts an unlikely extremely fast response, the FAR drops to 0.1-0.2%. The same applies when we raised R to 75 (i.e., the advanced attack's success rate is only 0.25% for latency=20ms).

Results: ROC and EER. Finally, Figure 4.10(b) plots the ROC results for all the replay attacks and synthesis attacks (with latency =20ms). We see that ElectricAuth achieves noticeable EERs only for the synthesis attacks (1.48% for simple synthesis and 1.59% for advanced synthesis). These results show that ElectricAuth is robust against replay and synthesis attacks, even those extreme ones.

4.8 Exploratory longitudinal study

We conducted an exploratory longitudinal study to examine ElectricAuth over time and against various environment and muscle conditions. Specifically, we performed **fixed-model-over-time tests**, which depicts how an authentication model trained using the first three days of data will perform over time and under muscle conditions (e.g., humidity, fatigue, etc.) and other non-predictable environmental factors that were not present in the training data;

Participants. Due to Covid-19, only two co-authors participated in this study (ages: 25 & 24 years old; weights: 70 & 54kg; heights 170 & 163cm; one male and one female).

Procedure. In the day prior to the start of the 24-day period, we conducted an initial calibration session (following the same method and apparatus described in Experiment #1).

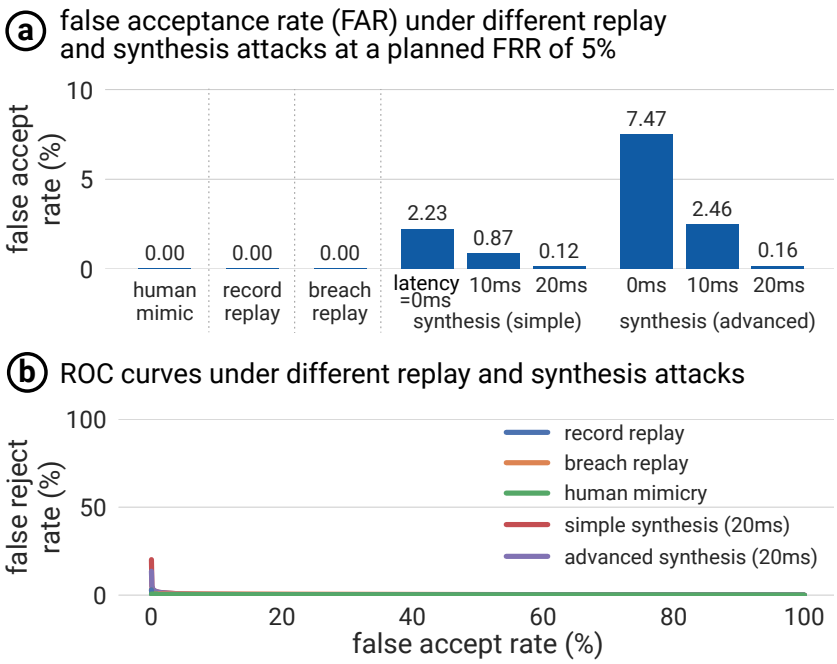

Figure 4.10: ElectricAuth's robustness against different replay and synthesis attacks. For online synthesis, the attacker had perfect records on responses to 50 challenges. Here ElectricAuth operates on length-6 challenges.

Then, we followed with 24 days of data collection. We collected data once a day. For each participant, we randomly chose 115 length-6 challenges to collect user responses.

For this study, we used fabric sleeves with embedded EMS electrodes at the precalibrated positions for each participant, following a design similar to [101]. Each day, participants were asked to wear their custom electrode-sleeves (depicting their calibrated locations). Participants fitted the sleeve by themselves prior to the trials by aligning markings on the sleeve with their elbow and top of wrist. If the electrode pads were dry, they re-gelled it using conductive gel. Then, they recorded their response to the 115 challenges every day. For each challenge, they collected more than 6 responses per day. After the trials, they removed the sleeve until the next day.

Conditions. To explore the impact of environmental and physiological variations, we conducted data collection under combinations of three conditions: (1) time of the day (morn-

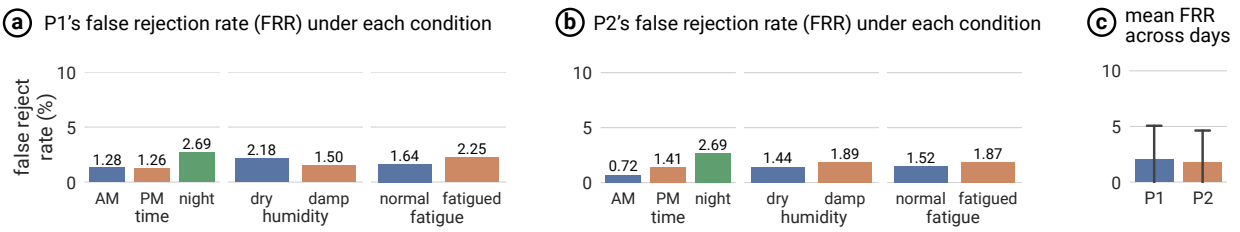

Figure 4.11: Results of fixed-model-over-time tests. (a) and (b) shows for both participants, our system is stable under various conditions; (c) our system is stable over time (21 days) for both participants.

ing/afternoon/late); (2) environment humidity (dry/damp); and (3) muscle fatigue (normal/fatigued). We randomly chose one combination per day, and each combination was tested at least twice during the study. In the damp condition, participants were asked to stay in their bathroom with the humidity at over 80% and temperature over 29 $^{\circ}C$ for more than 20 minutes right before the data collection. For dry condition, participants stayed in an air-conditioned room of humidity 55% and temperature $24^{\circ}C$. To test our system right after the muscles started to fatigue, participants were asked to do a routine of intense forearm muscle training (dumbbell wrist flexion and extension) for a minimum of 15 minutes before collecting data.

During the days in which we tested ElectricAuth under normal muscle conditions, participants still performed their forearm muscle training but after the data collection session. This allowed us to study if extended muscle exercise would affect the system performance.

Training the authentication model. For each participant, we used data collected in the first three days to train the authentication model (the anomaly detector and challenge classifier). For both participants, the training data were collected under the same (dry, pre-workout) condition. The rest of the data (21 days) were used for testing our authentication models. The testing data contained conditions both seen or unseen in the training data. We excluded day 10 and 11 for participant 1 due to need for replenishing the sticky gel on the electrodes, i.e., waiting for gel supply.

For all the trained models, we configured the anomaly detection thresholds to achieve a planned FRR of 2%. As discussed before, such configuration is set using only the training data without the knowledge of any testing data.

Results: fixed-model-over-time tests. To understand the impact of a specific condition (time of the day, environment humidity or muscle fatigue), Fig. 4.11(a)(b) shows the measured false rejection rate (FRR) under each condition. For both participants, the measured FRRs are reasonably consistent across conditions and closely match the planned FRR (2%). But more importantly, while our authentication models are trained only under the combination of dry and pre-workout conditions, they remain accurate under other conditions not seen during training. This provides initial evidence on the generality of ElectricAuth.

For both participants, we also plot the mean FRRs over time in Fig. 4.11(c). We see that the FRR is stable over time, (mean=2.01%, SD=3.13%) for participant 1 and (mean = 1.76%, SD=2.90%) for participant 2. No significant performance degradation over time was found for both participants. These results suggest that ElectricAuth remains relatively stable on a monthly scale.

4.9 Technical evaluation

We deepened our understanding of how future interactive systems might be built based on EMS authentication by measuring system latency, training time, and the feasibility of depth cameras as an alternative tracking modality.

4.9.1 Authentication latency

To measure ElectricAuth's inference latency (i.e., time needed to make a decision in runtime) and the model training, we utilized the data from the participants of Experiment#1, i.e., 115 length-6 challenges, eight response records per challenge for training, two response records for testing.

Run-time inference latency. As we probed the future of EMS-based authentication, we were interested in understanding how ElectricAuth would perform on smaller platforms, such as laptops or even embedded devices. As such, we ran our system on a MacBook Pro with a Intel Core i9-9880H CPU and on a Nvidia Jetson Nano embedded device (measuring 70 x 45 mm). Our results show that our system can authenticate a user in 3ms on laptop's CPU and 35ms on a small embedded device. This result suggests our approach is feasible for quick authentications and even available on mobile or wearable devices.

Training latency. Our results demonstrate that it took 35s (33s for anomaly detector; 2s for the challenge classifier) to train the complete model on a Nvidia Titan RTX GPU and 542s on a laptop's CPU (501s for anomaly detector; 41s for the challenge classifier).

4.9.2 Using camera to capture finger movements

While we used IMUs to capture finger movements in our user study, we believe these movements can also be captured via other modalities, such as depth cameras, a common platform for hand pose estimation [181, 204]. To test our belief, we carried out a simple feasibility experiment. Here, we swapped out IMU sensors with a RGB-D camera (Intel RealSense D435), which operates at 640x480 resolution and 30 frames per second. The camera was placed in front of the participant with a distance of 50cm.

Following the same procedure of Experiment#1, we recorded, via the depth camera, the responses to our 115 length-6 challenges on one participant. We then used an available hand gesture recognition model (from [104]) as our challenge verification model.

We found that the challenge classification accuracy for this simple feasibility experiment was 99.57% using the depth image. We also measure a 0.00% success rate of a record-replay attack against this participant's model.

4.10 Using synthetic data to test attacks at scale

Our user study demonstrated that ElectricAuth was accurate in verifying each of the 13 participants and robust against any attacks in that scale. However, gaining insight into how ElectricAuth would perform in larger deployments (e.g., 100's of users) is impractical by means of user studies at an early stage. To shed light into this, we explore a data-driven approach to evaluate ElectricAuth's robustness against impersonation attacks using synthetic data.

Procedure. We followed the recent approach of generating synthetic data by training deep generative models, which is shown to produce diverse and natural data (e.g., objects [174], human faces [228, 7], faces with emotions [123], and physiological data including ECG, EEG, and so forth [79]) beyond the training set. Specifically, we used the PixelCNN++ model [174], a state-of-the-art deep generative model for images (since we treat each response as an image). Following [174], we trained a generative model for each legitimate user in our experiment #2 (see Section 4.7.2), using the impersonator responses collected for this user (12 subjects and 115 challenges), conditioned on the challenge. Once trained, the generator produces random, natural variations of the training data, emulating responses of potential impersonators beyond our user study. We validated each generator using the well-known negative log likelihood (NLL), which produced results on par with (and often slightly better than) those reported by [174] on object/face images. This indicated that our trained generators are able to learn and follow the actual data distribution rather than overfitting to the training data.

Results: robustness against synthetic impersonators. For each of the 13 users in our experiment #2, we used the corresponding generator to produce 1075 impersonator responses against this user. These include 100 synthetic impersonators for each of 5 randomly selected challenges, and 5 additional impersonators for each of 115 challenges. We then tested these impersonator responses on ElectricAuth's authentication model for this user (i.e., the

same authentication model used in our experiment #2). All impersonator responses were rejected (i.e., 0% FAR at 5% FRR). This result aligns with our user study results, and sheds lights on ElectricAuth's robustness against impersonation attacks at larger scales.

4.11 Conclusions, applications & future work

We proposed, implemented and evaluated the use of electrical muscle stimulation (EMS) as a novel modality for active biometrics. We engineered an interactive system, which we called ElectricAuth, that stimulates the user's forearm muscles with a sequence of electrical impulses (i.e., an EMS challenge) and measures the user's involuntary finger movements (i.e., response to the challenge). The key idea behind ElectricAuth is that it leveraged EMS's *intersubject variability*, i.e., the same electrical stimulation results in different movements in different users because everybody's physiology is unique (e.g., differences in bone and muscular structure, skin resistance and composition, etc.). Moreover, we demonstrated that ElectricAuth is secure against data breaches and replay attacks, as it never reuses the same challenge twice in authentications – the key property that allowed ElectricAuth to achieve this is that in just one second of stimulation our system was able to encode one of 68M possible challenges.

4.11.1 Potential applications

We believe that ElectricAuth is applicable to a range of interactive scenarios in which users authenticate without needing to memorize passwords or PINs. We believe this is of special interest for devices that natively offer motion tracking or finger tracking, such as for virtual reality (which we illustrated in Figure 4.1 using the Oculus Quest), smartwatch-based interaction [129, 203, 229] or even leveraging a smartphone's built in IMUs. Furthermore, we believe our approach is of particular interest for accessibility scenarios, such as authentication for users with motor-impairments (e.g., spinal cord injury, arguably the most impactful

application of EMS in the medical domain [156]) but with intact musculature.

4.11.2 Future work

We believe this first exploration of EMS for user authentication provides fertile grounds for exploring subsequent challenges and opportunities: (1) while we have shown ElectricAuth worked well on the full set of 112 length-2 challenges and a subset of 115 length-6 challenges, growing the size of a challenge might enable new applications, as such, research is needed to demonstrate that this approach works across an even larger set of challenges and over a longer time period; (2) while ElectricAuth worked well on the 13 participants from our user studies, more physiological research is needed to deepen understanding of EMS's intersubject variability; (3) while ElectricAuth worked well on the controlled wrist posture, more investigation is required to understand its performance under other postures and their impacts; lastly, (4) as new EMS systems emerge from the medical domain (e.g., higher resolution electrode arrays [96, 101, 160], implanted devices [159], and so forth), a system like ElectricAuth will likely improve in wearability and performance, which will require further investigations.

CHAPTER 5

CONCLUSION

In this dissertation, we explore protecting users against intrusive sensing. We propose, design and prototype low-cost wearables that users can carry and turn on/off to prevent their private information from being extracted by unauthorized parties. My research produces three distinct wearables for protecting speech, typing content and identity privacy. In this chapter, we summarize the contribution of this dissertation, and discuss the insights learned and future directions for this emerging field.

5.1 Summary of contributions

Speech content privacy. In Chapter 2, we engineer a wearable microphone jammer as a bracelet, which disables surrounding microphones, including hidden ones. Our evaluation demonstrates that (1) our wearable largely outperforms existing jammers in coverage; (2) it remains effective even if the microphones are hidden and covered by various materials; and, (3) our user study participants feel that our wearable protects their speech privacy.

Typing content privacy. In Chapter 3, we study privacy threats against wearable keyboards where the keyboard and its layout are invisible in the physical world. We develop a new, more sophisticated attack that can successfully infer wearable keyboard typing content using just a RGB camera. We demonstrate that wearable keyboard typing is still under threat of keystroke inference attacks and additional protection methods are needed.

Identity privacy. In Chapter 4, we develop a wearable-based authentication method using muscle stimulation, which is robust against data breaches and replay attacks. Our evaluation results demonstrate that our system: (1) authenticates accurately and resists three common attacks: impersonation, replay and synthesis attacks; (2) performs stably over time, against various conditions; (3) runs with low latency and various tracking modalities.

5.2 Discussions

We discuss three key insights we learned throughout the studies in this dissertation.

Human body is a double-edged sword for privacy protection. My research shows that human body can be leveraged to improve protection strength and robustness. In particular, we leverage natural body movements to increase our wearable jammer's protection coverage (Chapter 2), and each user's unique finger movements triggered by EMS signals to identify the user (Chapter 4).

On the other hand, human body is also a source for leaking our private information. We demonstrate in Chapter 3 that by observing a user's typing finger movements, attackers can recover the typed content. In our earlier work [230], we also show that our natural body movements can leak our location inside a building to an attacker outside the property.

Privacy protection is a cat-and-mouse game. As shown in Chapter 3, a seemingly effective protection against existing attacks is eventually defeated by a more advanced adversary. This suggests the need to continuously evaluate (and update) any privacy protection system against evolving attackers.

We follow this principle in the development of our proposed wearable microphone jammer and EMS authentication system. In Chapter 2, to maximize the jammer's coverage and robustness, we opt to use a ring layout and build the jammer in the form of a bracelet. We experimentally show that our jammer is robust against advanced countermeasures, such as covering microphone with a variety of materials and post-processing and denoising the recording using state-of-the-art DNN models. In Chapter 4, to design a biometric authentication system that is breach-resistant, we examine the effectiveness of our system against various advanced countermeasures, including impersonation, replay and synthesis attacks. We also create a synthetic dataset to evaluate our system using data beyond those collected by our user studies.

The use of DNN models is a key factor in the design of privacy protection. In

Chapter 4, we use DNNs to extract useful features from complex human finger movements, where we show features extracted by DNNs outperform traditional hand-craft features by a large margin. Without the help of DNNs, the authentication accuracy could suffer significantly. On the other side, advances in DNNs also enables new attacks as we illustrated in Chapter 3. Further development of privacy protection tools should consider both the integration of DNN models into the design and the evaluation against advanced attacks enabled by DNN models.

5.3 Future directions

We sketch an outline of future directions in the filed of wearable-based privacy protection.

Protecting privacy of other content / behavior categories. While this dissertation explores the two major modalities we use to produce content, there are other ones worth further study. Examples include: (1) hand writing; (2) sign language; (3) lip movements during speech. Beyond content (*what we are*) and identity (*who we are*) privacy, building wearables for behavior (*how we are / where we are*) privacy protection are also important and worth further investigation.

Protecting against intrusive identification. In Chapter 4, we study the impact of intrusive sensing on biometric authentication where the attacker records a user's biometric data and then bypasses the authentication system by replaying the recorded data. On the other hand, intrusive sensing also poses another significant threat where the attacker wants to identify the target using the recorded biometric, such as face and voice. Future wearables can be built to dynamically change one's identity appearance in the physical world by applying time-varying obfuscation signals.

REFERENCES

[1] 2022. Learning to type with mobile keyboards: Findings with a randomized keyboard. *Computers in Human Behavior* (2022).

[2] Muhammad Taher Abuelma'atti. 2003. Analysis of the effect of radio frequency interference on the DC performance of bipolar operational amplifiers. *IEEE Transactions on Electromagnetic compatibility* 45, 2 (2003), 453–458.

[3] Ali Ahanger and Anil Kumar. 2014. Effect of Anthropometric Factors on Motor Nerve Conduction Velovity in Healthy Kashmiri Population. *International Journal of Medical and Applied Sciences* 3 (feb 2014), 125–132.

[4] Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. EyeSpyVR: Interactive Eye Sensing Using Off-the-Shelf, Smartphone-Based VR Headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 57 (July 2018), 10 pages. DOI:http://dx.doi.org/10.1145/3214260

[5] Rawan Alharbi, Tammy Stump, Nilofar Vafaie, Angela Pfammatter, Bonnie Spring, and Nabil Alshurafa. 2018. I Can'T Be Myself: Effects of Wearable Cameras on the Capture of Authentic Behavior in the Wild. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 90 (Sept. 2018), 40 pages. DOI:http://dx.doi.org/10.1145/3264900

[6] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *Proc. of MobiCom*. DOI:http://dx.doi.org/10.1145/2789168.2790109

[7] A. Ali-Gombe, E. Elyan, and C. Jayne. 2019. Multiple Fake Classes GAN for Data Augmentation in Face Image Dataset. In *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[8] Amnesty International. 2021. Ban dangerous facial recognition technology that amp lifies racist policing. (2021).

[9] Leonardo Angelini, Maurizio Caon, Stefano Carrino, Luc Bergeron, Nathalie Nyffeler, Mélanie Jean-Mairet, and Elena Mugellini. 2013. Designing a Desirable Smart Bracelet for Older Adults. In *Proceedings of ACM Conference on Pervasive and Ubiquitous Computing Adjunct*. 425–434.

[10] Xavier Anguera, Chuck Wooters, and Javier Hernando. 2007. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing* 15, 7 (2007), 2011–2022.

[11] Apple Inc. 2021. Gboard-the Google Keyboard. https://apps.apple.com/us/app/gboard-the-google-keyboard/id1091700242. (2021).

[12] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *Proc. of ICML*.

[13] Daniel Arp, Erwin Quiring, Christian Wressnegger, and Konrad Rieck. 2017. Privacy threats through ultrasonic side channels on mobile devices. In *Proc. of EuroS&P*.

[14] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *Proc. of ICLR*.

[15] Dmitri Asonov and Rakesh Agrawal. 2004a. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004.* IEEE, 3–11.

[16] D. Asonov and R. Agrawal. 2004b. Keyboard acoustic emanations. In *Proc. of IEEE S&P*. DOI:http://dx.doi.org/10.1109/SECPRI.2004.1301311

[17] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: Using the Doppler Effect for Multi-device Interaction. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[18] Kazuki Awaki, Chun-Hao Liao, Makoto Suzuki, and Hiroyuki Morikawa. 2016. Speakerless Sound-based 3D Localization with Centimeter-level Accuracy. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp)*.

[19] Aware. 2020. Voice Authentication. https://www.aware.com/voice-authentication/.

[20] Michael Backes, Markus Dürmuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. 2010. Acoustic Side-Channel Attacks on Printers.. In *USENIX Security symposium*. 307–322.

[21] Davide Balzarotti, Marco Cova, and Giovanni Vigna. 2008. ClearShot: Eavesdropping on Keyboard Input from Video. In *Proc. of IEEE S&P*. DOI:http://dx.doi.org/10.1109/SP.2008.28

[22] Salil P Banerjee and Damon L Woodard. 2012. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research* (2012).

[23] Xueliang Bao, Yuxuan Zhou, Yunlong Wang, Jianjun Zhang, Xiaoying Lü, and Zhigong Wang. 2018. Electrode placement on the forearm for selective stimulation of finger extension/flexion. *PloS one* 13, 1 (2018).

[24] Dom Barnard. 2018. Average Speaking Rate and Words per Minute. VIRTUALSPEECH. (January 2018). https://virtualspeech.com/blog/average-speaking-rate-words-per-minute.

[25] Leonard E Baum and others. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* (1972).

[26] Victoria Bellotti and Abigail Sellen. 1993. Design for Privacy in Ubiquitous Computing Environments. In *Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work (ECSCW'93)*. Kluwer Academic Publishers, Norwell, MA, USA, 77–92. `http://dl.acm.org/citation.cfm?id=1241934.1241940`

[27] Big Brother Watch. 2021. Stop facial recognition. `https://bigbrotherwatch.org.uk/campaigns/stop-facial-recognition`. (2021).

[28] Joseph A. Boales, Farrukh Mateen, and Pritiraj Mohanty. 2017. Micromechanical microphone using sideband modulation of nonlinear resonators. *Applied Physics Letters* 111, 9 (2017), 093504.

[29] Thomas Brewster. 2019. We Broke Into A Bunch Of Android Phones With A 3D-Printed Head. `https://www.forbes.com/sites/thomasbrewster/2018/12/13/we-broke-into-a-bunch-of-android-phones-with-a-3d-printed-head/#4a9a79213307`.

[30] Matthew Brocker and Stephen Checkoway. 2014. iSeeYou: Disabling the MacBook Webcam Indicator LED. In *Proceedings of the 23rd USENIX Conference on Security Symposium (SEC'14)*. USENIX Association, Berkeley, CA, USA, 337–352. `http://dl.acm.org/citation.cfm?id=2671225.2671247`

[31] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K. Varshney, and Dawn Song. 2020. Anomalous Instance Detection in Deep Learning: A Survey. (2020).

[32] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Proc. of CHI*.

[33] Joseph P Campbell. 1997. Speaker recognition: A tutorial. *Proc. IEEE* 85, 9 (1997), 1437–1462.

[34] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[35] Arpan Chakraborty, Brent Harrison, Pu Yang, David Roberts, and Robert St. Amant. 2014. Exploring Key-Level Analytics for Computational Modeling of Typing Behavior. In *Proc. of HotSoS*.

[36] R. Chandler, C. Clauser, J. McConville, Herbert Reynolds, and J. Young. 1975. Investigation of Inertial Properties of the Human Body. (03 1975), 171.

[37] Theocharis Chatzis, Andreas Stergioulas, Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. 2020. A comprehensive study on deep learning-based 3D hand pose estimation methods. *Applied Sciences* (2020).

[38] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Tracking Keystrokes Using Wireless Signals. *In Proc. of MobiSys* (2015).

[39] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: Sharing Files Between Multiple Devices Using In-air Gestures. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[40] H. Chung, M. Iorga, J. Voas, and S. Lee. 2017. Alexa, Can I Trust You? *Computer* 50, 9 (2017), 100–104.

[41] Hyunji Chung and Sangjin Lee. 2018. Intelligent Virtual Assistant knows Your Life. *CoRR* abs/1803.00466 (2018).

[42] Charles E Clauser, John T McConville, and J W Young. 1971. Weight, Volumn, and Center of Mass of Segments of The Human Body. *Journal of Occupational and Environmental Medicine* (1971).

[43] Clearview.AI Inc. 2017. Clearview.AI. `https://www.clearview.ai`. (2017).

[44] William W. Cohen. 2015. Enron Email Dataset. `https://www.cs.cmu.edu/~enron/`. (2015).

[45] Gregg D Colton. 1997. High-Tech Approaches to Breeching Examination Security. Espionage 101. (1997).

[46] CopyLeaks. 2015. Plagiarism checker API - integrate AI powered API, Copyleaks. `https://api.copyleaks.com/`. (2015).

[47] Cory Cornelius, Ronald Peterson, Joseph Skinner, Ryan Halter, and David Kotz. 2014. A Wearable System That Knows Who Wears It. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '14)*. Association for Computing Machinery, New York, NY, USA, 55–67. `DOI:http://dx.doi.org/10.1145/2594368.2594369`

[48] LibriSpeech Dataset. 2017. `http://www.openslr.org/12`. (2017).

[49] Luigi De Russis, Dario Bonino, and Fulvio Corno. 2013. The Smart Home Controller on Your Wrist. In *Proceedings of ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp)*. 8.

[50] Silent Ultrasonic Microphone Defeater. 2019. `https://www.uspystore.com/silent-ultrasonic-microphone-defeater`. (2019).

[51] A Desplantez, C Cornu, and F Goubel. 1999. Viscous properties of human muscle during contraction. *Journal of biomechanics* 32, 6 (1999), 555–562. DOI:http://dx.doi.org/https://doi.org/10.1016/S0021-9290(99)00039-1

[52] K.N. Dewangan, G.V. [Prasanna Kumar], P.L. Suja, and M.D. Choudhury. 2005. Anthropometric dimensions of farm youth of the north eastern region of India. *International Journal of Industrial Ergonomics* 35, 11 (2005), 979 – 989. DOI:http://dx.doi.org/https://doi.org/10.1016/j.ergon.2005.04.003

[53] Vivek Dhakal. 2017. Identification of typing behaviors from large keystroke dataset. Master Thesis, Aalto University. (2017).

[54] Hidden Microphone dictaphone Bug Recording supressor ultrasonic + Noise Generator by i4 Technology. 2019. https://www.amazon.com/Microphone-dictaphone-Recording-supressor-ultrasonic/dp/B01MG4WACJ/. (2019).

[55] Laura Dipietro, Angelo M Sabatini, and Paolo Dario. 2008. A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 4 (2008), 461–482.

[56] Carl Doersch. 2016. Tutorial on Variational Autoencoders. (2016).

[57] C. Doersch and A. Zisserman. 2017. Multi-task Self-Supervised Visual Learning. In *Proc. of ICCV*.

[58] R Drillis, R Contini, and M Bluestein. 1964. Body Segment Parameters; A Survey of Measurement Techniques. *Artificial limbs* 8 (1964), 44–66. http://europepmc.org/abstract/MED/14208177

[59] John J Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)* 25, 6 (2018), 1–40.

[60] Simon Eberz, Nicola Paoletti, Marc Roeschlin, Andrea Patané, Marta Kwiatkowska, and Ivan Martinovic. 2017. Broken Hearted: How To Attack ECG Biometrics. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*. The Internet Society. https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/broken-hearted-how-attack-ecg-biometrics/

[61] EDUCBA. 2022. OpenCV perspectivetransform. https://www.educba.com/opencv-perspectivetransform/. (2022).

[62] Basic VAE Example. 2019. https://github.com/pytorch/examples/tree/master/vae.

[63] Hugging Face. 2022. Wer - a hugging face space by evaluate-metric. https://huggingface.co/spaces/evaluate-metric/wer. (2022).

[64] Farzam Farbiz, Zhou Hao Yu, Corey Manders, and Waqas Ahmad. 2007. An Electrical Muscle Stimulation Haptic Feedback for Mixed Reality Tennis Game. In *ACM SIGGRAPH 2007 Posters (SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 140–es. DOI:http://dx.doi.org/10.1145/1280720.1280873

[65] Al Faruque, Mohammad Abdullah, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan. 2016. Acoustic side-channel attacks on additive manufacturing systems. In *Proceedings of the 7th International Conference on Cyber-Physical Systems*. IEEE Press, 19.

[66] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. 2020. Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–7.

[67] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. 2016. How We Type: Movement Strategies and Performance in Everyday Typing. In *Proc. of CHI*.

[68] Carl Fischer, Kavitha Muthukrishnan, Mike Hazas, and Hans Gellersen. 2008. Ultrasound-aided Pedestrian Dead Reckoning for Indoor Navigation. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT '08)*.

[69] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.

[70] Michael J. Grey. 1997. *Viscoelastic properties of the human wrist during the stabilization phase of a targeted movement*. Master's thesis. Simon Fraser University, Burnaby, Canada.

[71] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. Qwertyring: Text entry on physical surfaces using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–29.

[72] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops Surfaces (ITS '15)*. Association for Computing Machinery, New York, NY, USA, 59–62. DOI: http://dx.doi.org/10.1145/2817721.2817722

[73] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. SoundWave: Using the Doppler Effect to Sense Gestures. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI)*.

[74] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov, and Yuval Elovici. 2017. SPEAKE(a)R: Turn Speakers to Microphones for Fun and Profit. In *11th USENIX*

Workshop on Offensive Technologies (WOOT 17). USENIX Association, Vancouver, BC. https://www.usenix.org/conference/woot17/workshop-program/presentation/guri

[75] SDR Harridge, R Bottinelli, M Canepari, MA Pellegrino, C Reggiani, M Esbjörnsson, and B Saltin. 1996. Whole-muscle and single-fibre contractile properties and myosin heavy chain isoforms in humans. *Pflügers Archiv* 432, 5 (1996), 913–920.

[76] Hasomed. 2020. https://hasomed.de/en/.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of CVPR*.

[78] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *In Proc. of NIPS* (2015).

[79] Andres Hernandez-Matamorosb, Hamido Fujita, and Hector Perez-Meanac. 2020. A novel approach to create synthetic biomedical signals using BiRNN. *Elsevier Information Sciences* 541 (Dec 2020), 218–241.

[80] AL Hof. 1998. In vivo measurement of the series elasticity release curve of human triceps surae muscle. *Journal of biomechanics* 31, 9 (1998), 793–800.

[81] Christian Holz and Patrick Baudisch. 2013. Fiberio: A Touchscreen That Senses Fingerprints. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 41–50. DOI:http://dx.doi.org/10.1145/2501988.2502021

[82] Christian Holz and Marius Knaust. 2015. Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 303–312. DOI:http://dx.doi.org/10.1145/2807442.2807458

[83] Jayakumar Hoskere. 2019. Everyday ai: Beyond spell check, how google docs is smart enough to correct grammar — google cloud blog. https://cloud.google.com/blog/products/g-suite/everyday-ai-beyond-spell-check-how\-google-docs-is-smart-enough-to-correct-grammar. (2019).

[84] Christopher-Eyk Hrabia, Katrin Wolf, and Mathias Wilhelm. 2013. Whole Hand Modeling Using 8 Wearable Sensors: Biomechanics for Hand Pose Prediction. In *Proceedings of the 4th Augmented Human International Conference (AH '13)*. Association for Computing Machinery, New York, NY, USA, 21–28. DOI:http://dx.doi.org/10.1145/2459236.2459241

[85] Report: Data Breach in Biometric Security Platform Affecting Millions of Users. 2019. https://www.vpnmentor.com/blog/report-biostar2-leak/.

[86] Google is permanently nerfing all Home Minis because mine spied on everything I said 24/7 [Update x2]. 2017. https://www.androidpolice.com/2017/10/10/google-nerfing-home-minis-mine-spied-everything-said-247/. (2017).

[87] Steven A Israel and John M Irvine. 2012. Heartbeat biometrics: a sensing system perspective. *International Journal of Cognitive Biometrics* 1, 1 (2012), 39–65.

[88] Speech jammer TOWER-A for blocking proffessional microphones / counter surveillance. 2019. https://www.detective-store.com/speech-jammer-tower-a-for-blocking-proffessional-microphones-counter-surveillance-1516.html. (2019).

[89] Wolfgang Jank. 2006. The EM algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In *Perspectives in operations research*.

[90] Jemine, Corentin. 2019a. Real-Time Voice Cloning. https://github.com/CorentinJ/Real-Time-Voice-Cloning. (2019).

[91] Jemine, Corentin. 2019b. Resemblyzer. https://github.com/resemble-ai/Resemblyzer. (2019).

[92] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proc. of ICML*.

[93] Wenqiang Jin, Srinivasan Murali, Huadi Zhu, and Ming Li. 2021. Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations. In *Proc. of ACM SIGSAC CCS*. DOI:http://dx.doi.org/10.1145/3460120.3484549

[94] Zach Jorgensen and Ting Yu. 2011. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 476–482.

[95] Ilias Kaperonis. 1984. Industrial espionage. *Computers & Security* 3, 2 (1984), 117–121.

[96] T. Keller, M. Lawrence, A. Kuhn, and M. Morari. 2006. New Multi-Channel Transcutaneous Electrical Stimulation Technology for Rehabilitation. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. 194–197.

[97] Gary S. Kendall, Christopher Haworth, and Rodrigo F. Cádiz. 2014. Sound Synthesis with Auditory Distortion Products. *Computer Music Journal* 38 (2014), 5–23. Issue 4.

[98] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on*

User Interface Software and Technology (UIST'12). Association for Computing Machinery, New York, NY, USA, 167–176. DOI:http://dx.doi.org/10.1145/2380116.2380139

[99] Yoon Sang Kim, Byung Seok Soh, and Sang-Goog Lee. 2005. A new wearable input device: SCURRY. *IEEE Transactions on Industrial Electronics* 52, 6 (2005), 1490–1499.

[100] Tomi Kinnunen, Md. Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In *Proc. Interspeech 2017*. 2–6. DOI:http://dx.doi.org/10.21437/Interspeech.2017-1111

[101] Jarrod Knibbe, Paul Strohmeier, Sebastian Boring, and Kasper Hornbæk. 2017. Automatic Calibration of High Density Electric Muscle Stimulation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article Article 68 (Sept. 2017), 17 pages. DOI:http://dx.doi.org/10.1145/3130933

[102] Michinari Kono, Takumi Takahashi, Hiromi Nakamura, Takashi Miyaki, and Jun Rekimoto. 2018. Design Guideline for Developing Safe Systems That Apply Electricity to the Human Body. *ACM Trans. Comput.-Hum. Interact.* 25, 3, Article Article 19 (June 2018), 36 pages. DOI:http://dx.doi.org/10.1145/3184743

[103] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. Real-time hand gesture detection and classification using convolutional neural networks. In *Proc. of IEEE FG 2019*.

[104] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. 2019. Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks. In *14th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2019, Lille, France, May 14-18, 2019*. IEEE, 1–8. DOI:http://dx.doi.org/10.1109/FG.2019.8756576

[105] Krishna, Arvind. 2020. Ibm ceo's letter to congress on racial just ice reform. (2020).

[106] Ernst Kruijff, Dieter Schmalstieg, and Steffi Beckhaus. 2006. Using Neuromuscular Electrical Stimulation for Pseudo-Haptic Feedback. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '06)*. Association for Computing Machinery, New York, NY, USA, 316–319. DOI:http://dx.doi.org/10.1145/1180495.1180558

[107] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. 2020. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proc. of CVPR*.

[108] Vladimir I Levenshtein and others. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*.

[109] Jingjie Li, Kassem Fawaz, and Younghyun Kim. 2019. Velody: Nonlinear Vibration Challenge-Response for Resilient User Authentication. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. Association for Computing Machinery, New York, NY, USA, 1201–1213. DOI: http://dx.doi.org/10.1145/3319535.3354242

[110] Junhong Li, Chenghao Wang, Wei Ren, and Jun Ma. 2017. ZnO thin film piezoelectric micromachined microphone with symmetric composite vibrating diaphragm. *Smart Materials and Structures* 26, 5 (2017), 055033.

[111] Mengyuan Li, Yan Meng, Junyi Liu, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. 2016. When CSI Meets Public WiFi: Inferring Your Mobile Phone Password via WiFi Signals. In *Proc. of ACM SIGSAC CCS*. DOI:http://dx.doi.org/10.1145/2976749.2978397

[112] Chong L. Lim, Chris Rennie, Robert J. Barry, Homayoun Bahramali, Ilario Lazzaro, Barry Manor, and Evian Gordon. 1997. Decomposing skin conductance into tonic and phasic components. *International Journal of Psychophysiology* 25, 2 (1997), 97 – 109. DOI:http://dx.doi.org/https://doi.org/10.1016/S0167-8760(96)00713-1

[113] John Lim, True Price, Fabian Monrose, and Jan-Michael Frahm. 2020. Revisiting the Threat Space for Vision-Based Keystroke Inference Attacks. In *Proc. of ECCV*.

[114] Feng Lin, Kun Woo Cho, Chen Song, Wenyao Xu, and Zhanpeng Jin. 2018. Brain Password: A Secure and Truly Cancelable Brain Biometrics for Smart Headwear. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*. Association for Computing Machinery, New York, NY, USA, 296–309. DOI:http://dx.doi.org/10.1145/3210240.3210344

[115] Kang Ling, Yuntang Liu, Ke Sun, Wei Wang, Lei Xie, and Qing Gu. 2020. SpiderMon: Towards Using Cell Towers as Illuminating Sources for Keystroke Monitoring. In *Proc. of IEEE INFOCOM*. DOI:http://dx.doi.org/10.1109/INFOCOM41043.2020.9155447

[116] Pedro Lopes and Patrick Baudisch. 2013. Muscle-Propelled Force Feedback: Bringing Force Feedback to Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2577–2580. DOI:http://dx.doi.org/10.1145/2470654.2481355

[117] Pedro Lopes, Alexandra Ion, and Patrick Baudisch. 2015a. Impacto: Simulating Physical Impact by Combining Tactile Stimulation with Electrical Muscle Stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 11–19. DOI:http://dx.doi.org/10.1145/2807442.2807443

[118] Pedro Lopes, Alexandra Ion, Willi Mueller, Daniel Hoffmann, Patrik Jonell, and Patrick Baudisch. 2015b. Proprioceptive Interaction. In *Proceedings of the 33rd Annual*

ACM Conference on Human Factors in Computing Systems (CHI '15). Association for Computing Machinery, New York, NY, USA, 939–948. DOI:http://dx.doi.org/10.1145/2702123.2702461

[119] Pedro Lopes, Patrik Jonell, and Patrick Baudisch. 2015c. Affordance++: Allowing Objects to Communicate Dynamic Use. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2515–2524. DOI:http://dx.doi.org/10.1145/2702123.2702128

[120] Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing Haptics to Walls & Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1471–1482. DOI:http://dx.doi.org/10.1145/3025453.3025600

[121] Pedro Lopes, Sijing You, Alexandra Ion, and Patrick Baudisch. 2018. Adding Force Feedback to Mixed Reality Experiences and Games Using Electrical Muscle Stimulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 446, 13 pages. DOI:http://dx.doi.org/10.1145/3173574.3174020

[122] Pedro Lopes, Doăa Yüksel, François Guimbretière, and Patrick Baudisch. 2016. Muscle-Plotter: An Interactive System Based on Electrical Muscle Stimulation That Produces Spatial Output. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 207–217. DOI:http://dx.doi.org/10.1145/2984511.2984530

[123] Y. Luo and B. Lu. 2018. EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2535–2538.

[124] Dominique Machuletz, Stefan Laube, and Rainer Böhme. 2018. Webcam Covering As Planned Behavior. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 180, 13 pages. DOI:http://dx.doi.org/10.1145/3173574.3173754

[125] Sapna Maheshwari. 2018. Hey, Alexa, What Can You Hear? And What Will You Do With It? New York Times. (March 2018). https://mobile.nytimes.com/2018/03/31/business/media/amazon-google-privacy-digital-assistants.html.

[126] Robert J Mailloux. 1982. Phased array theory and technology. *Proc. IEEE* 70, 3 (1982), 246–291.

[127] Majestic. 2022. The Majestic Million. https://majestic.com/reports/majestic-million". (2022).

[128] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. 2009. *Handbook of fingerprint recognition*. Springer Science & Business Media.

[129] Meethu Malu, Pramod Chundury, and Leah Findlater. 2018. Exploring Accessible Smartwatch Interactions for People with Upper Body Motor Impairments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 12. DOI: http://dx.doi.org/10.1145/3173574.3174062

[130] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (Sp)IPhone: Decoding Vibrations from Nearby Keyboards Using Mobile Phone Accelerometers. In *Proc. of ACM CCS*. DOI:http://dx.doi.org/10.1145/2046707.2046771

[131] SMM Martens, Joris M Mooij, N Jeremy Hill, Jason Farquhar, and Bernhard Schölkopf. 2011. A graphical model framework for decoding in the visual ERP-based BCI speller. *Neural computation* (2011).

[132] Asier Marzo, Sue Ann Seah, Bruce W. Drinkwater, Deepak Ranjan Sahoo, Benjamin Long, and Sriram Subramanian. 2015. Holographic acoustic elements for manipulation of levitated objects. *Nature Communications* 6, 1 (2015), 8661.

[133] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. 2019. The jester dataset: A large-scale video dataset of human gestures. In *Proc. of IEEE/CVF ICCVW*.

[134] R. Mayrhofer and H. Gellersen. 2007. On the Security of Ultrasound as Out-of-band Channel. In *Proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium*.

[135] MediaPipe Hands. 2022. JavaScript Solution API. https://google.github.io/mediapipe/solutions/hands#javascript-solution-api. (2022).

[136] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid touch interaction in virtual reality using wearable sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 519–528.

[137] Meta Platforms Inc. 2020. META QUEST 2. https://store.facebook.com/quest/products/quest-2. (2020).

[138] Meta Platforms Inc. 2021. Introducing Horizon Workrooms: Remote Collaboration Reimagined. https://about.fb.com/news/2021/08/introducing-horizon-workrooms-remote-collaboration-reimagined. (2021).

[139] Microsoft. 2021. Keyboard - Mixed Reality — Microsoft Docs. https://docs.microsoft.com/en-us/windows/mixed-reality/design/keyboard. (2021).

[140] Wayne J Millar. 1986. Distribution of body weight and height: comparison of estimates based on self-reported and observed measures. *Journal of Epidemiology & Community Health* 40, 4 (1986), 319–323.

[141] Saeed Mirzamohammadi and Ardalan Amiri Sani. 2016. Viola: Trustworthy Sensor Notifications for Enhanced Privacy on Mobile Systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '16)*. ACM, New York, NY, USA, 263–276. DOI:http://dx.doi.org/10.1145/2906388.2906391

[142] Rafael Morales, Asier Marzo, Sriram Subramanian, and Diego Martínez. 2019. LeviProps: Animating Levitated Optimized Fabric Structures using Holographic Acoustic Tweezers. In *Proc. of UIST*.

[143] Tim Moynihan. 2016. Alexa and Google Home Record What You Say. But What Happens to That Data? Wired. (December 2016). https://www.wired.com/2016/12/alexa-and-google-record-your-voice/.

[144] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In *Proc. of CVPR*.

[145] Tahrima Mustafa, Richard Matovu, Abdul Serwadda, and Nicholas Muirhead. 2018. Unsure How to Authenticate on Your VR Headset? Come on, Use Your Head!. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (IWSPA '18)*. Association for Computing Machinery, New York, NY, USA, 23–30. DOI:http://dx.doi.org/10.1145/3180445.3180450

[146] Gergana Stefanova Nikolova and Yuli Emilov Toshev. 2007. Estimation of male and female body segment parameters of the Bulgarian population using a 16-segmental mathematical model. *Journal of Biomechanics* 40, 16 (2007), 3700–3707. DOI:http://dx.doi.org/https://doi.org/10.1016/j.jbiomech.2007.06.016

[147] Oculus. 2019. Introducing Hand Tracking on Oculus Quest-Bringing Your Real Hands into VR. https://www.oculus.com/blog/introducing-hand-tracking-on-oculus-quest-bringing-your-real-hands-into-vr/.

[148] New Generation of High Grade Smartphone Scrambler. 2019. https://www.globaltscmgroup-usa.com/. (2019).

[149] Wayne O Olsen. 1998. Average speech levels and spectra in various speaking/listening conditions: A summary of the Pearson, Bennett, & Fidell (1977) report. *American Journal of Audiology* 7, 2 (1998).

[150] Patrick Howell O'Neill. 2019. Data leak exposes unchangeable biometric data of over 1 million people. https://www.technologyreview.com/f/614163/data-leak-exposes-unchangeable-biometric-data-of-over-1-million-people/.

[151] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors (Basel)* 16, 1 (Jan 2016). DOI:http://dx.doi.org/10.3390/s16010115

[152] Marcos Ortega, M.G. Penedo, J. Rouco, N. Barreira, and M.J. Carreira. 2009. Personal verification based on extraction and characterisation of retinal feature points. *Journal of Visual Languages & Computing* 20, 2 (2009), 80–90. DOI:http://dx.doi.org/https://doi.org/10.1016/j.jvlc.2009.01.006

[153] Minna Pakanen, Ashley Colley, Jonna Häkkilä, Johan Kildal, and Vuokko Lantz. 2014. Squeezy Bracelet: Designing a Wearable Communication Device for Tactile Interaction. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational (NordiCHI '14)*. 305–314.

[154] Leysia Palen and Paul Dourish. 2003. Unpacking "Privacy" for a Networked World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 129–136. DOI:http://dx.doi.org/10.1145/642611.642635

[155] Shwetak N. Patel, Jeffrey S. Pierce, and Gregory D. Abowd. 2004. A Gesture-Based Authentication Scheme for Untrusted Public Terminals. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. Association for Computing Machinery, New York, NY, USA, 157–160. DOI:http://dx.doi.org/10.1145/1029632.1029658

[156] P. Hunter Peckham and Jayme S. Knutson. 2005. Functional Electrical Stimulation for Neuromuscular Applications. *Annual Review of Biomedical Engineering* 7, 1 (2005), 327–360. DOI:http://dx.doi.org/10.1146/annurev.bioeng.6.040803.140103

[157] Ken Pfeuffer, Matthias J. Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 2019. Behavioural Biometrics in VR: Identifying People from Body Motion and Relations in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 110, 12 pages. DOI:http://dx.doi.org/10.1145/3290605.3300340

[158] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215 – 249. DOI:http://dx.doi.org/https://doi.org/10.1016/j.sigpro.2013.12.026

[159] D. Popovic, L. L. Baker, and G. E. Loeb. 2007. Recruitment and Comfort of BION Implanted Electrical Stimulation: Implications for FES Applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15, 4 (2007), 577–586.

[160] Ana Popović-Bijelić, Goran Bijelić, Nikola Jorgovanović, Dubravka Bojanić, Mirjana B. Popović, and Dejan B. Popović. 2005. Multi-Field Surface Electrode for Se-

lective Electrical Stimulation. *Artificial Organs* 29, 6 (2005), 448–452. DOI:http://dx.doi.org/10.1111/j.1525-1594.2005.29075.x

[161] Rebecca S. Portnoff, Linda N. Lee, Serge Egelman, Pratyush Mishra, Derek Leung, and David Wagner. 2015. Somebody's Watching Me?: Assessing the Effectiveness of Webcam Indicator Lights. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1649–1658. DOI:http://dx.doi.org/10.1145/2702123.2702164

[162] William K Pratt. 1972. Generalized Wiener filtering computation techniques. *IEEE Trans. Comput.* 100, 7 (1972), 636–641.

[163] Lawrence Rabiner and Biinghwang Juang. 1986. An introduction to hidden Markov models. *IEEE ASSP magazine* (1986).

[164] Rahul Raguram, Andrew M. White, Dibyendusekhar Goswami, Fabian Monrose, and Jan-Michael Frahm. 2011. ISpy: Automatic Reconstruction of Typed Input from Compromising Reflections. In *Proc. of ACM CCS*.

[165] Vijay Rajanna, Seth Polsley, Paul Taele, and Tracy Hammond. 2017. A Gaze Gesture-Based User Authentication System to Counter Shoulder-Surfing Attacks. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 1978–1986. DOI:http://dx.doi.org/10.1145/3027063.3053070

[166] Kasper Bonne Rasmussen, Marc Roeschlin, Ivan Martinovic, and Gene Tsudik. 2014. Authentication Using Pulse-Response Biometrics. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society. https://www.ndss-symposium.org/ndss2014/authentication-using-pulse-response-biometrics

[167] Brian Reed. 1997. The physiology of neuromuscular electrical stimulation. *Pediatric Physical Therapy* 9, 3 (1997), 96–102. https://journals.lww.com/pedpt/Fulltext/1997/00930/The_Physiology_of_Neuromuscular_Electrical.2.aspx

[168] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* (2014).

[169] Dario Rethage, Jordi Pons, and Xavier Serra. 2018. A wavenet for speech denoising. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5069–5073.

[170] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. 2017. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of ACM MobiSys*.

[171] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible Voice Commands: The Long-Range Attack and Defense. In *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

[172] Mohd Sabra, Anindya Maiti, and Murtuza Jadliwala. 2020. Zoom on the keystrokes: exploiting video calls for keystroke inference attacks. *arXiv preprint arXiv:2010.12078* (2020).

[173] U.S. Occupational Safety and Health Administration (OSHA). 2013. Occupational Safety and Health Administration Technical Manual. `https://www.osha.gov/dts/osta/otm/new_noise/#appendixc`. (August 2013).

[174] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. 2017. PixelCNN++: A PixelCNN Implementation with Discretized Logistic Mixture Likelihood and Other Modifications. In *ICLR*.

[175] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-available Input with Muscle-computer Interfaces. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*. 167–176.

[176] Munehiko Sato, Rohan S. Puri, Alex Olwal, Yosuke Ushigome, Lukas Franciszkiewicz, Deepak Chandra, Ivan Poupyrev, and Ramesh Raskar. 2017. Zensei: Embedded, Multi-Electrode Bioimpedance Sensing for Implicit, Ubiquitous User Recognition. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3972–3985. `DOI: http://dx.doi.org/10.1145/3025453.3025536`

[177] SeleniumHQ. 2022. SeleniumHQ/selenium: A browser automation framework and ecosystem. `https://github.com/SeleniumHQ/selenium`. (2022).

[178] Woon Seob Lee and Seung S. Lee. 2008. Piezoelectric microphone built on circular diaphragm. 144 (06 2008), 367–373.

[179] Diksha Shukla, Rajesh Kumar, Abdul Serwadda, and Vir V. Phoha. 2014. Beware, Your Hands Reveal Your Secrets!. In *Proc. of ACM CCS*. `DOI:http://dx.doi.org/10.1145/2660267.2660360`

[180] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. 2017. Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4645–4653. `DOI:http://dx.doi.org/10.1109/CVPR.2017.494`

[181] Ayan Sinha, Chiho Choi, and Karthik Ramani. 2016. DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed With Deep Features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4150–4158. `DOI: http://dx.doi.org/10.1109/CVPR.2016.450`

[182] Ivo Sluganovic, Marc Roeschlin, Kasper B. Rasmussen, and Ivan Martinovic. 2016. Using Reflexive Eye Movements for Fast Challenge-Response Authentication. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1056–1067. DOI:http://dx.doi.org/10.1145/2976749.2978311

[183] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on TNNLS* (2022).

[184] Liwei Song and Prateek Mittal. 2017. Inaudible Voice Commands. *CoRR* abs/1708.07238 (2017).

[185] Spiceworks. 2018. Spiceworks Study Reveals Nearly 90 Percent of Businesses Will Use Biometric Authentication Technology by 2020. https://www.spiceworks.com/press/releases/spiceworks-study-reveals-nearly-90-percent-businesses-will-use-biometric-authentication-technology-2020/.

[186] Diana S. Stetson, James W. Albers, Barbara A. Silverstein, and Robert A. Wolfe. 1992. Effects of age, sex, and anthropometric factors on nerve conduction measures. *Muscle & Nerve* 15, 10 (1992), 1095–1104. DOI:http://dx.doi.org/10.1002/mus.880151007

[187] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *CHI Conference on Human Factors in Computing Systems*. 1–16.

[188] Primož Strojnik, Alojz Kralj, and I Ursic. 1979. Programmed six-channel electrical stimulator for complex stimulation of leg muscles during walking. *IEEE Transactions on Biomedical Engineering* 2 (1979), 112–116.

[189] Fangmin Sun, Chenfei Mao, Xiaomao Fan, and Ye Li. 2019a. Accelerometer-Based Speed-Adaptive Gait Authentication Method for Wearable IoT Devices. *IEEE Internet of Things Journal* 6, 1 (2019), 820–830. DOI:http://dx.doi.org/10.1109/JIOT.2018.2860592

[190] Fangmin Sun, Chenfei Mao, Xiaomao Fan, and Ye Li. 2019b. Accelerometer-Based Speed-Adaptive Gait Authentication Method for Wearable IoT Devices. *IEEE Internet of Things Journal* 6, 1 (2019), 820–830. DOI:http://dx.doi.org/10.1109/JIOT.2018.2860592

[191] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. 2020. TelemetRing: A Batteryless and Wireless Ring-Shaped Keyboard Using Passive Inductive Telemetry. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1161–1168.

[192] Emi Tamaki, Takashi Miyaki, and Jun Rekimoto. 2011. PossessedHand: Techniques for Controlling Human Hands Using Electrical Muscles Stimuli. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems (CHI '11). Association for Computing Machinery, New York, NY, USA, 543–552. DOI:http://dx.doi.org/10.1145/1978942.1979018

[193] Tap System Inc. 2021. Tap Strap 2 - Wearable Keyboard, Mouse & Air Gesture Controller. https://www.tapwithus.com/. (2021).

[194] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue. 2013. A survey of keystroke dynamics biometrics. *The Scientific World Journal* 2013 (2013). DOI:http://dx.doi.org/10.1155/2013/408280

[195] The Ghost Howls. 2021. Horizon Workrooms review: nice, but not compelling for work yet. https://skarredghost.com/2021/08/26/horizon-workrooms-review. (2021).

[196] IBM Speech to Text. 2018. https://www.ibm.com/watson/services/speech-to-text/. (Jul. 2018).

[197] Your Phone Is Listening Literally Listening to Your TV. 2015. https://www.theatlantic.com/technology/archive/2015/11/your-phone-is-literally-listening-to-your-tv/416712/. (2015).

[198] The Muscle Physiology Laboratory UCSD. 2008. Fundamental Functional Properties of Skeletal Muscle. http://muscle.ucsd.edu/musintro/props.shtml.

[199] Kai Uffmann, Stefan Maderwald, Waleed Ajaj, Craig G Galban, Serban Mateiescu, Harald H Quick, and Mark E Ladd. 2004. In vivo elasticity measurements of extremity skeletal muscle with MR elastography. *NMR in Biomedicine: An International Journal Devoted to the Development and Application of Magnetic Resonance In Vivo* 17, 4 (2004), 181–190.

[200] University of Notre Dame. 1995. The frequency of the letters of the alphabet in English. https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html. (1995).

[201] Valentin Bazarevsky and Fan Zhang. 2021. On-Device, Real-Time Hand Tracking with MediaPipe. (2021). https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html

[202] Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory* (1967).

[203] Tran Huy Vu, Archan Misra, Quentin Roy, Kenny Choo Tsu Wei, and Youngki Lee. 2018. Smartwatch-Based Early Gesture Detection 8 Trajectory Tracking for Interactive Gesture-Driven Applications. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1 (March 2018), 27. DOI:http://dx.doi.org/10.1145/3191771

[204] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2019. Self-Supervised 3D Hand Pose Estimation Through Training by Fitting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 10845–10854. DOI:http://dx.doi.org/10.1109/CVPR.2019.01111

[205] Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. 2015. MagnifiSense: Inferring Device Interaction Using Wrist-worn Passive Magneto-inductive Sensors. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[206] Richard P Wildes. 1997. Iris recognition: an emerging biometric technology. *Proc. IEEE* 85, 9 (1997), 1348–1363.

[207] Davey Winder. 2019a. Apple's iPhone FaceID Hacked In Less Than 120 Seconds. https://www.forbes.com/sites/daveywinder/2019/08/10/apples-iphone-faceid-hacked-in-less-than-120-seconds/#449ff4b621bc.

[208] Davey Winder. 2019b. Hackers Claim Any Smartphone Fingerprint Lock Can Be Broken In 20 Minutes. https://www.forbes.com/sites/daveywinder/2019/11/02/smartphone-security-alert-as-hackers-claim-any-fingerprint-lock-broken-in-20-minutes/#588a90116853.

[209] Charlie Wood. 2017. Devices sprout ears: What do Alexa and Siri mean for privacy? Christian Science Monitor. (January 2017). https://www.csmonitor.com/Technology/2017/0114/Devices-sprout-ears-What-do-Alexa-and-Siri-mean-for-privacy.

[210] Candid Wueest. 2017. Everything You Need to Know About the Security of Voice-Activated Smart Speakers. Symantec. (Nov. 2017). https://www.symantec.com/blogs/threat-intelligence/security-voice-activated-smart-speakers.

[211] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated Residual Transformations for Deep Neural Networks. *arXiv preprint arXiv:1611.05431* (2016).

[212] Chenren Xu, Sugang Li, Gang Liu, Yanyong Zhang, Emiliano Miluzzo, Yih-Farn Chen, Jun Li, and Bernhard Firner. 2013b. Crowd++: Unsupervised Speaker Count with Smartphones. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[213] Yi Xu, Jared Heinly, Andrew M White, Fabian Monrose, and Jan-Michael Frahm. 2013a. Seeing double: Reconstructing obscured typed input from repeated compromising reflections. In *Proc. of ACM SIGSAC CCS*.

[214] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, Thorsten Holz and Stefan Savage (Eds.). USENIX Association, 497–512.

[215] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. 2019. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE/CVF ICCVW*.

[216] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proc of UIST*.

[217] Tuo Yu, Haiming Jin, and Klara Nahrstedt. 2016. WritingHacker: Audio Based Eavesdropping of Handwriting via Mobile Devices. In *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[218] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *Proc. of ICML*.

[219] Qinggang Yue, Zhen Ling, Xinwen Fu, Benyuan Liu, Kui Ren, and Wei Zhao. 2014. Blind Recognition of Touched Keys on Mobile Devices. In *Proc. of ACM SIGSAC CCS*. DOI:http://dx.doi.org/10.1145/2660267.2660288

[220] Qinggang Yue, Zhen Ling, Wei Yu, Benyuan Liu, and Xinwen Fu. 2015. Blind recognition of text input on mobile devices via natural language processing. In *Proc. of the Workshop on Privacy-Aware Mobile Computing*.

[221] Chen Yunfang, Zhu Yihong, Zhou Hao, Chen Wei, and Zhang Wei. 2018. Enhanced Keystroke Recognition Based on Moving Distance of Keystrokes Through WiFi. In *Proc. of NSS*.

[222] Clint Zeagler. 2017. Where to Wear It: Functional, Technical, and Social Considerations in On-body Location for Wearable Technology 20 Years of Designing for Wearability. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers (ISWC '17)*.

[223] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. (2020).

[224] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible voice commands. In *Proceedings of ACM Conference on Computer and Communications Security (CCS)*.

[225] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[226] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. 2020. How does mixup help with robustness and generalization? *arXiv preprint arXiv:2010.04819* (2020).

[227] Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu. 2018. EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia* (2018).

[228] Jian Zhao, Lin Xiong, Karlekar Jayashree, Jianshu Li, Fang Zhao, Zhecan Wang, Sugiri Pranata, Shengmei Shen, Shuicheng Yan, and Jiashi Feng. 2017. Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 65–75.

[229] Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. 2016. AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 81–86. DOI: http://dx.doi.org/10.1145/2984511.2984568

[230] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y Zhao, and Haitao Zheng. 2018. Et tu alexa? when commodity wifi devices turn into adversarial motion sensors. *arXiv preprint arXiv:1810.10109* (2018).

[231] Li Zhuang, Feng Zhou, and J. Tygar. 2005. Keyboard acoustic emanations revisited. In *Proc. of ACM CCS*.

[232] Li Zhuang, Feng Zhou, and J Doug Tygar. 2009. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)* 13, 1 (2009), 3.

APPENDIX A

TYPING PRIVACY THREATS AGAINST WEARABLE KEYBOARDS

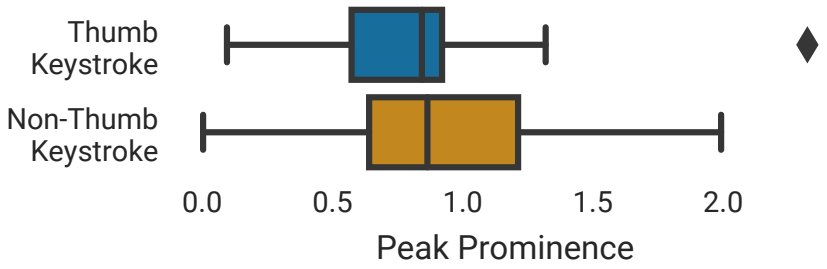

Figure A.1: Distribution of negative acceleration's peak prominence for thumb-based and non-thumb keystrokes.

	Left hand	Right hand	Multi Touch
P0	I	I	No
P1	T, I, M, R, P	I, M, R, P	No
P2	I	I	No
P3	I, M, R	I, M	Yes
P4	I	I	No
P5	I, M, R	T, I, M, R	No
P6	I, M, R, P	T, I, M, R, P	Yes
P7	I, M, R	T, I, M, R	No
P8	I, M, R, P	T, I, M, R, P	Yes
P9	I, M, R, P	T, I, M, R	No
P10	I, M, R	I, M, R	Yes
P11	T, I, M, R, P	I, M, R, P	Yes
P12	I, M, R	I, M, R	No
P13	I, M, R, P	T, I, M, R, P	Yes
P14	I, M	I, M, R	No
P15	I, M, R P	T, I, M, R	No

Table A.1: Typing behaviors of our 16 participants. We refer to each hand's fingers by Thumb(T), Index(I), Middle(M), Ring(R) and Pinky(P)).

Unsup. Infer	*DNN Detector*	*Label Filter*	*DNN Classifier*	*Noise Train*	**CER** (%)	**WER** (%)	**Sim.** (%)
✔					26.3	67.0	0.0
✔	✔				23.7	61.2	18.8
✔	✔	✔	✔		14.9	45.2	50.1
✔		✔	✔	✔	10.8	34.6	73.0
✔	✔	✔	✔	✔	6.9	17.8	79.9

Table A.2: The contribution of each component of the pipeline tested on P9

Inferred Text	Ground Truth	CER (%)
once we know exactly what contracts we are looking at we can fine tune the calculation	once we know exactly what contracts we are looking at we can fine tune the calculation	0
each trader will be us ed to manage their individual position and profitability goals for the simulation n	each trader will be asked to manage their individual position and profitability goals for the simulation	3.8
traders will be managing their individual booms and associates product	traders will be managing their individual books and associated products	5.5
the acatt ached draft is fairly legalistic in tone	the attached draft is fairly legalistic in tone	6.3
the lle ongoing uncertainty about our future coupled with the contestant media scrutiny makes the situation difficult for all of uw	the ongoing uncertainty about our future coupled with the constant media scrutiny makes this situation difficult for all of us	7.1
hbur message will be scanned an nd hec ed for viruses prior to requested release	your message will be scanned and checked for viruses prior to requested release	7.6
i attach a letter of intent ent w ich i hope covers all the points we discussed this m ing	i attach a letter of intent which i hope covers all the points we discussed this morning	9.9
os one of the enhanced sec rit measur s we have ecentl emppo edm we will be e ch c ing emp o ee badges at the entrance to the ballr o om	as one of the enhanced security measures we have recently employed we will b e checking employee badges at the entrance to the ballroom	11.8

Figure A.2: Examples of the final recovered text compared to the ground truth text.