TOWARDS ACCURATE RANGING AND VERSATILE AUTHENTICATION FOR SMART MOBILE DEVICES

By

Lingkun Li

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2022

ABSTRACT

TOWARDS ACCURATE RANGING AND VERSATILE AUTHENTICATION FOR SMART MOBILE DEVICES

By

Lingkun Li

Internet of Things (IoTs) was rapidly developed during past years. Smart devices, such as smartphones, smartwatches, and smart assistants, which are equipped with smart chips as well as sensors, provide users with many easy used functions and lead them to a more convenient life. In this dissertation, we carefully studied the birefringence of the transparent tape, the nonlinear effects of the microphone, and the phase characteristic of the reflected ultrasound, and make use of such effects to design three systems, RainbowLight, Patronus, and BreathPass, to provide users with accurate localization, privacy protection, and authentication, respectively.

RainbowLight leverages observation direction-varied spectrum generated by a polarized light passing through a birefringence material, i.e., transparent tape, to provide localization service. We characterize the relationship between observe direction, light interference and the special spectrum, and using it to calculate the direction to a chip after taking a photo containing the chip. With multiple chips, RainbowLight designs a direction intersection based method to derive the location. In this dissertation, we build the theoretical basis of using polarized light and birefringence phenomenon to perform localization. Based on the theoretical model, we design and implement the RainbowLight on the mobile device, and evaluate the performance of the system. The evaluation results show that RainbowLight achieves 1.68 cm of the median error in the X-axis, 2 cm of the median error in the Y-axis, 5.74 cm of the median error in Z-axis, and 7.04 cm of the median error with the whole dimension. It is the first system that could only use the reflected lights in the space to perform visible light positioning.

Patronus prevents unauthorized speech recording by leveraging the nonlinear effects of commercial off-the-shelf microphones. The inaudible ultrasound scramble interferes recording of unauthorized devices and can be canceled on authorized devices through an adaptive filter. In

this dissertation, we carefully studied the nonlinear effects of ultrasound on commercial microphones. Based on the study, we proposed an optimized configuration to generate the scramble. It would provide privacy protection againist unauthorized recordings that does not disturb normal conversations. We designed, implemented a system including hardware and software components. Experiments results show that only 19.7% of words protected by Patronus' scramble can be recognized by unauthorized devices. Furthermore, authorized recordings have 1.6x higher perceptual evaluation of speech quality (PESQ) score and, on average, 50% lower speech recognition error rates than unauthorized recordings.

BreathPass uses speakers to emit ultrasound signals. The signals are reflected off the chest wall and abdomen and then back to the microphone, which records the reflected signals. The system then extracts the fingerprints from the breathing pattern, and use these fingerprints to perform authentication. In this dissertation, we characterized the challenge of conducting authentication with the breathing pattern. After addressing these challenges, we designed such a system and implemented a proof-of-concept application on Android platform. We also conducted comprehensive experiments to evaluate the performance under different scenarios. BreathPass achieves an overall accuracy of 83%, a true positive rate of 73%, and a false positive rate of 5%, according to performance evaluation results.

In general, this dissertation provides an enhanced ranging and versatile authentication systems of Internet of Things.

To my parents and grandparents for their love and support.

ACKNOWLEDGEMENTS

There are many people I would like to say thanks.

To my advisor, Dr. Yunhao Liu, without your encouragement, I will never come to the U.S. and experience a different culture. I will never have a chance to broaden my horizon, to see a different world. Without your selfless support, I will never have a good life here. I will never forget the days we sit in McDonald's or Pandas Express and discuss my life, my future, and give me your understanding of the research.

To my advisor, Dr. Zhichao Cao, and my master's advisor Dr. Jiliang Wang, I learned a lot from you – research taste, writing styles, and presentation skills. Without your support and guidance, I am not able to finish my Ph.D. study.

I want to thank Professor Eric Torng for his careful edit of my paper. His most professional skills and the hardest working attitude are worth every one of us learning. Thanks to my guidance committee members, Dr. Li Xiao, Dr. Qiben Yan, Dr. Mi Zhang, for their guide and support. I also want to thank the lab mates from the beginning to the current, Dr. Fan Dang, Dr. Pengjin Xie, Dr. Chunyu Qiao, Dr. Yinghui Li, Ye Zhou, Zhao Wang, Qing Zhou, and many others for their collaboration from many aspects. Thanks to my best friends Yue Jiang and Junjie Han during my lifetime in the United States. It is because of their accompany and help that I can successfully complete my studies.

Thanks to my parents and grandparents for their endless love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

Internet of Things (IoTs) was rapidly developed during past years. Smart devices, such as smartphones, smartwatches, and smart assistants, which are equipped with smart chips as well as sensors, provide users with many easy used functions and lead them to a more convenient life.

Many works make use of the device's original function or extract features from them to design systems to serve people, and may or may not avoid side effects of the device. Side effects are not the main function of devices, which are sometimes avoided by users. Recently, there are some applications [1, 2, 3] carefully study the side effects of the devices. For example, LiTell [1] found that because of the manufacturing error, the flashing rates of fluorescent lights varies from one to another. Then, it samples such flashing rates as the landmark and design a localization system. Manufacturing errors are not wanted by people, and usually try to avoid. Before LiTell, many Visible Light Positioning systems [4, 5, 6] either need to modulating landmarks by dynamically changing the flash frequencies, brightness, or need user to perform certain actions and using geometry to calculate the position of the camera. LiTell, however, make use of such errors and regard the flashing frequencies from such errors as the landmark, hence requiring no modulation or using requirement. It reduces both deployment costs and using costs.

Another example is LiShield [2], which exploits the rolling shutter effect of CMOS camera. Rolling Shutter, comparing to Global Shutter, captures one column at a time instead of the whole frame. It is a kind of side effect of cheap camera, whereas the expensive camera, e.g., SLR camera, equipped with Global Shutter usually avoids. LiShield, however, exploits the Rolling Shutter effect to design a visual privacy protection system. Specifically, LiShield designs a light bulb which consists of three color bulbs. Three color bulbs illuminate alternatively with extremely high frequency but can be distinguished by the rolling shutter. Although human eyes cannot sense the color bulbs flicker, the camera with the rolling shutter, however, would capture the column with one bulb illuminate at a time, thus generating a mask with multiple color stripes on the photo captured.

Therefore, it is difficult for human to recognize the content of the photo and prevents unauthorized device to take photos. It also designs a mechanism to remove such a mask on authorized devices, hence allowing authorized devices to take photos.

The nonlinear effect of commercial off-the-shelf (COTS) microphones is another kind of side effects. When a pair of ultrasond captured by the microphone, nonlinearity could generate a shadow spectrum within the audible frequency range with a careful design of the ultrasond. DolphinAttack [7] makes use of the nonlinear effect to break in the voice control system. Many works [8, 9] aims to remove such spectrum in order to get rid of unexpected attacks. UPS+ [3], however, carefully studies the pattern of the nonlinear effect of microphones, and designs a new ultrasonic positioning system, which uses extremely high frequency of the sound to avoid disturbing pets and infants.

In this dissertation, we carefully study two of the side effects, one is the birefringence of the transparent tape, which could blur the underside image when we observe it hence people usually want to avoid. Another is the nonlinearity of COTS microphones, which was discussed above. Based on our carefully study, we propose two systems, RainbowLight and Patronus. RainbowLight uses birefringence to localize a camera. Different from previous works, RainbowLight works even when light bulbs is turned off, hence reducing the deploying and using costs. Patronus leverages the nonlinearity to emit inaudible scramble to interfere unauthorized recordings. We also design a mechanism to cancel out such scrambles with the scramble pattern giving to authorized devices, hence preventing unauthorized recordings while allowing authorized recordings.

From 2019, COVID-19 pandemic brings people into an inconvenient life. COVID-19 virus attacks human's lung and make patients hard to breath. To cope with the COVID-19 pandemic, existing effort [10] implements a mobile application that leverages ultrasound to capture user's breath, and then detects whether the user's lung functionality is normal in a non-invasive manner. In this dissertation, besides the two systems which leverage side effects to provide an enhanced ranging and a privacy protection system, we propose BreathPass, an authentication system leveraging user's breath to cope with the problem that Face-ID is hard to use when a user wears a face cover and

Fingerprint-ID is also hard to use when a user wears a pair of rubber gloves. Compare to existing biometric authentication systems, BreathPass is more resilient to the replay attack and has a high flexibility to mobile devices. In addition, with BreathPass, users are no need to take off their face covers or gloves when they use the application that requires "who you are" authentication; e.g., Apple Pay. It brings users more safety towards the COVID-19 pandemic.

1.1 Proposed techniques and applications

1.1.1 Positioning with birefringence

Ubiquitous existence of lights makes Visible Light Positioning (VLP) become popular and has attracted much research effort. Existing VLP approaches typically need to use a specially designed light bulb as a transmitter or a specially designed receiver to collect light information, or requires a strict user operation (e.g., capturing multiple light bulbs at a time with horizontally holding the smartphone, or needs to keep the light bulb turning on). This results in high deployment, maintenance, and using costs.

In Chapter 2, we present RainbowLight. RainbowLight uses birefringence material to generate a spatial-characterized light pattern. A camera could capture different color pattens from different positions, and achieves a low-cost, high-precision 3D positioning.

We implement RainbowLight and conduct comprehensive experiments. The evaluation results show that RainbowLight achieves 1.68 cm of the median error in the X-axis, 2 cm of the median error in the Y-axis, 5.74 cm of the median error in Z-axis, and 7.04 cm of the median error with the whole dimension.

1.1.2 Audio privacy protection with nonlinearity of microphones

The widespread adoption and ubiquity of smart devices equipped with microphones (e.g., cell-phones, smartwatches, etc.) unfortunately create many significant privacy risks. In recent years, there have been several cases of people's conversations being secretly recorded, sometimes initiated

by the device itself. Although some manufacturers are trying to protect users' privacy, to the best of our knowledge, there is not any effective technical solution available.

In Chapter 3, we present Patronus, a system that can both prevent unauthorized devices from making secret recordings while allowing authorized devices to record conversations. Patronus prevents unauthorized speech recording by emitting what we call a *scramble*, a low-frequency noise generated by inaudible ultrasonic waves. The scramble prevents unauthorized recordings by leveraging the nonlinear effects of commercial off-the-shelf microphones. The frequency components of the scramble are randomly determined and connected with linear chirps, and the frequency period is fine-tuned so that the scramble pattern is hard to attack. Patronus allows authorized speech recording by secretly delivering the scramble pattern to authorized devices, which can use an adaptive filter to cancel out the scramble.

We implement a prototype system and conduct comprehensive experiments. Our results show that only 19.7% of words protected by Patronus' scramble can be recognized by unauthorized devices. Furthermore, authorized recordings have 1.6x higher perceptual evaluation of speech quality (PESQ) score and, on average, 50% lower speech recognition error rates than unauthorized recordings.

1.1.3 Authentication with user's breath

In Chapter 4, we propose BreathPass, a non-invasive authentication system that characterizes the chest/abdomen movement incurred by human breath to enable unlocking smart devices while wearing various types of face covers, clothing, and in different postures. To capture the breathing pattern, BreathPass uses speakers to emit ultrasound signals. The signals are reflected off the chest wall and abdomen and then back to the microphone, which records the reflected signals. The system then extracts the breathing pattern from the reflected signals, and further extracts fingerprints from the breathing pattern, and use these fingerprints to perform authentication. We carefully design a Deep Neural Network model and explore its capacity for feature abstraction in order to address the challenges associated with tiny position changes resulting in different breathing patterns and the

extremely narrow bandwidth of breathing.

We implement a prototype and conduct extensive experiments. BreathPass achieves an overall accuracy of 83%, a true positive rate of 73%, and a false positive rate of 5%, according to performance evaluation results.

1.2 Organization

The reminder of this dissertation is as follows, in Chapter 2, we discuss visible light positioning with birefringence; in Chapter 3, we discuss audio privacy protection with nonlinearity of microphone; in Chapter 4, we discuss authentication with user's breathing; in Chapter 5 we conclude this dissertation.

CHAPTER 2

RAINBOWLIGHT: ENABLING 3D AMBIENT LIGHT POSITIONING WITH MOBILE PHONES AND BATTERY-FREE CHIPS

2.1 Overview

The rapid development of mobile and Internet of Things (IoTs) facilitates the development of a smarter world. More and more smart robots and smart devices are used in different places, such as factories, airports and even at home. Indoor localization significantly expands the capability of these devices, and thus it attracts much research effort, e.g., a large collection of RF-based [11, 12, 13, 14, 15, 16] positioning approaches are proposed.

Visible Light Positioning (VLP) has recently been shown as a promising approach for indoor localization, owing to its potential of high localization precision with ubiquitous existence of light. The basic idea of VLP is to exploit features and information from received light to derive the relative position to light. For example, many approaches use LED light with a controller [17, 18, 19, 5, 20] to modulate the required features. Thus a receiver can use the modulated features for localization. Further, instead of using a controller to actively modulate information in light, many approaches [21, 1, 22, 23] resort to using intrinsic features of light or receiver. Meanwhile, [24, 25, 26, 6, 27, 28] use geometrical relationships among lights for localization.

Existing VLP approaches exhibit high accuracy for indoor localization. However, there still exist the following limitations that hinder their application: (1) Special designed LEDs with controllers [17, 20] or the receiver with sensors [5, 28]. Such kinds of LED/receiver are still not widely used in today's buildings. (2) Pre-collected features for all lights[1, 22]. This introduces a high overhead. It is difficult to ensure the features are stable over time and the system needs to keep updated with all lights. (3) Strict usage requirement. For example, [1] requires to keep the mobile phone horizontal and [24] requires to capture at least 3 lamps in a photo each time. (4) Do not work when the light is turned off in the daytime. During the daytime, people often turn their lights

off and use the ambient light, i.e., sunlight passing through the window, to meet the requirement of illumination. Like DarkLight [29] in the field of visible light communication (VLC) realizes the requirement of communication with extremely-low luminance, we think that perform localization with the light turned off is non-trivial as well. Existing works could not work at all when the light is switched off because they are depended on LEDs or receivers. Those limitations incur a high deployment, maintenance and usage overhead.

To address those limitations, we propose RainbowLight, a low-cost 3D localization approach which significantly reduces the deployment, maintenance, and usage overhead. Our key finding for RainbowLight is that light through a chip containing polarizer and birefringence material will produce different interference patterns and light spectrum in different directions. We go deep into the birefringence principle to analyze the relationship between direction, light interference, and spectrum and derive a model to characterize the relationship. The model builds the foundation of obtaining the direction to a chip based on the received light spectrum. By calculating directions to multiple chips, we can derive the 3D localization of the receiver theoretically.

In the practical design of RainbowLight, we find that the light spectrum is difficult to measure on commercial off-the-shelf (COTS) mobile phones. We use the color extracted from photo to approximate light spectrum and show its effectiveness. To derive light direction for localization, the theoretical model requires various parameters, e.g., optic parameters and thickness of the material, which are difficult to measure in practice. Instead of measuring those parameters, we build a sparse initial mapping between hue value and direction by sampling. Further, we conduct model-based interpolation on the sparse initial mapping to derive a fine-grained mapping. Such a sparse sampling only needs to be performed once for the same type of polarizer and birefringence material. After capturing a photo containing multiple chips, we extract the color pattern of those chips and calculate directions to them. Finally, we leverage a direction based intersection method to calculate the location.

In our implementation, we use transparent adhesive tape as birefringence material. We make small transparent chips by sticking tape with a thin plastic polarizer. In localization, we only need

to place multiple chips to a certain plane (e.g., lamp cover, a glass window) to enable it for 3D localization (see Figure 2.10). It should be noted that RainbowLight does not actively modulate information in the light, and thus it also works for light off scenario in the daytime. We can place chips on a wall, table, or other flat surfaces. This significantly extends the application scenarios.

We evaluate the performance of RainbowLight in different scenarios for different types of light as well as different types of surfaces. The evaluation results show that RainbowLight achieves a high localization accuracy and low cost. It also works well even for light off scenario in the daytime.

The contributions of our work are as follows:

- We show that light through a chip made by polarizer and birefringence material will produce different interference patterns and light spectrum in different directions. We analyze and derive a model to characterize the direction, interference, and light spectrum as the foundation for 3D localization.

- Based on the model, we propose RainbowLight, a low-cost ambient light 3D localization approach with a low deployment, maintenance, and usage cost.

- We implement RainbowLight and evaluate its performance through extensive experiments. RainbowLight achieves an average localization error of 3.3 cm in 2D and 9.6 cm in 3D, and an error of 7.4 cm in 2D and 20.5 cm in 3D for light off scenario in the daytime.

The organization of the remainder is as follows. Section 2.2 introduces the background of our work. Section 2.3 presents 3D localization model of RainbowLight. Section 2.4 and 2.5 introduce the design and implementation of RainbowLight, respectively. Section 2.6 discusses the approach of deploying RainbowLight to enable getting the absolute position in a large area. Section 2.7 presents evaluation results of RainbowLight. Section 2.8 introduces related work.

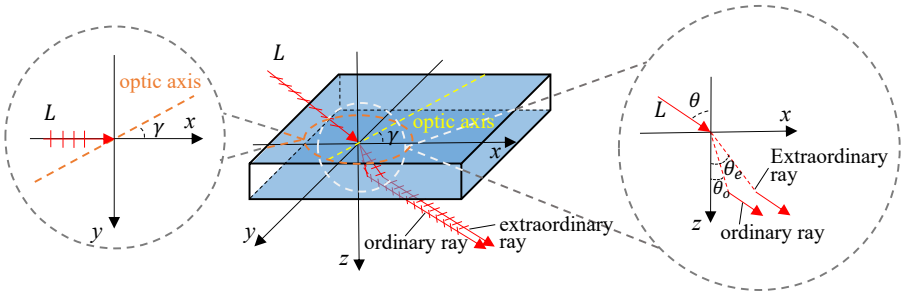


Figure 2.1: Illustration of birefringence.

2.2 Background

2.2.1 Polarization

Polarization is a feature of the transverse wave to specify its oscillation in different directions. Natural light, such as light from a lamp, has different oscillations. Polarizer for light is a kind of device that allows light with the oscillation direction parallel to its *transmission axis*, and blocks light with the oscillation direction perpendicular to its transmission axis. The polarizer is widely used in various applications, *e.g.*, each 3D glasses has two polarizers for two lenses with different transmission axes allowing light with different oscillation to pass.

A polarizer with a single transmission axis is called *linear polarizer*. Light is polarized after passing through a polarizer. The polarized light has an oscillation direction parallel with the transmission axis of the polarizer. Denote the angle between the oscillation direction of light and the transmission axis of a polarizer as ϕ, according to Malus's law[30], the intensity of the light that passes through the polarizer, denoted by I_ϕ, is given by

$$I_\phi = Icos^2\phi, \tag{2.1}$$

where I is the original intensity of light.

Natural light has oscillation in any direction. When natural light passes through a linear polarizer, it becomes linearly polarized light, i.e., light with a single oscillation direction.

2.2.2 Birefringence

Birefringence [31] is a feature for an optically anisotropic material such as plastics, calcite, and quartz. When a ray of light passes through a birefringence material, two refracted rays can be observed. As shown in Figure 2.1, the ray of light is split into two rays taking different paths in the material. Meanwhile, those two rays have orthogonal polarization directions and different refractive indices in the birefringence material. There is a special direction, namely *optic axis*, for each certain type of birefringence material. One of the two rays, called *ordinary ray*, has a polarization direction vertical with the optic axis. Its refractive index is called *ordinary refractive index* and is denoted by n_o. Another ray, called *extraordinary ray*, has a polarization direction along the optic axis. Its refractive index is called *extraordinary refractive index* and is denoted by n_e.

As shown in Figure 2.1, according to Snell's Law [32], we have

$$n_{air} sin\theta = n_e sin\theta_e = n_o sin\theta_o \tag{2.2}$$

where $n_{air} \approx 1$ is the refractive index in air, and θ_o and θ_e are the refractive angle of ordinary ray and extraordinary ray, respectively. Usually, $n_e \neq n_o$, and the refractive angles and refractive indexes of ordinary ray and extraordinary ray are different. Thus there is an optical path difference between the two rays after the birefringence material. For a certain type of material, n_o is fixed determined by the material, while n_e varies depending on the direction of the incident ray. As shown in Figure 2.1, denote the incident angle as θ and the angle between the incident light projection on the incident plane and optic axis as γ. We will show how to obtain n_e and θ_e using θ and γ in practice. Then we can calculate the optical path for ordinary ray and extraordinary ray.

According to Snell's Law, if the incident light L is linearly polarized and the angle between polarization direction and optic axis is ϕ_1, the intensity of ordinary ray I_o and extraordinary ray I_e can be calculated as

$$\begin{aligned} I_o &= I \sin^2 \phi_1 \\ I_e &= I \cos^2 \phi_1. \end{aligned} \tag{2.3}$$

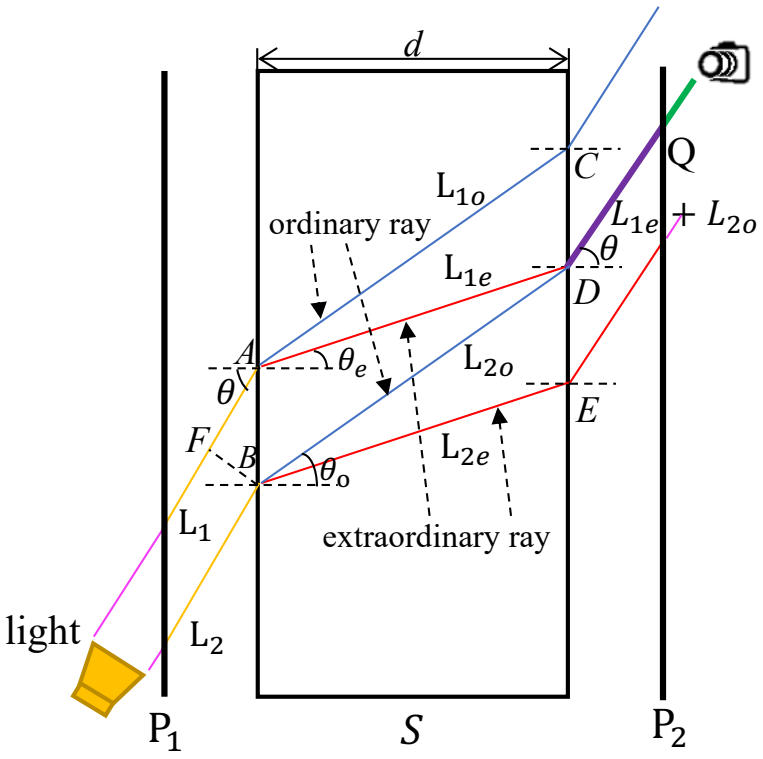


Figure 2.2: Illustration of light interference.

where I is the intensity of L.

2.2.3 Interference

When two light beams L_1 and L_2 have the same frequency, stable phase difference δ and same polarization direction, they can interfere with each other. For a different value of δ, the two light beams can have different interference results. The interference intensity can be calculated as:

$$I_i = I_1 + I_2 + 2\sqrt{I_1 I_2} cos\delta \tag{2.4}$$

where I_i is the light intensity after interference, and I_1 and I_2 are the intensities of L_1 and L_2, and δ is the phase difference between L_1 and L_2 and often derived from the optical path difference.

2.3 Localization Basics

We aim to answer the question of why observing the chip made by polarizers and birefringence material in different directions would get different color patterns. In this section, we firstly build

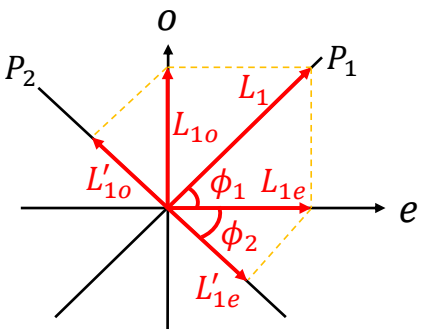


Figure 2.3: Polarization and intensity change through P_1, S and P_2.

a model from the background to show the principle of our 3D positioning approach. Then we conduct an experiment to validate our model. Because some of the parameters are hard to measure, it is difficult to directly use such a model to perform positioning directly. As a result, we show how to address those challenges in our design in section 2.4. Therefore, readers who are not interested in the detailed analysis of RainbowLight can skip this section directly.

As shown in Figure 2.2, a birefringence material S is placed between two polarizers P_1 and P_2. Light from a source (e.g., a lamp) first passes through polarizer P_1 and becomes a linearly polarized light. Consider two rays of the polarized light L_1 and L_2 incident into S at point A and B, respectively. As introduced in Section 2.2, L_1 is separated into two parts: L_{1o} (the ordinary ray) and L_{1e} (the extraordinary ray). The refractive indices of the ordinary ray and the extraordinary ray are n_o and n_e, respectively. Similarly, L_2 is separated into two parts: L_{2o} (the ordinary ray) and L_{2e} (the extraordinary ray). After passing through another polarizer P_2, the light L_{1e} and L_{2o} become L'_{1e} and L'_{2o}. L'_{2o} of L_2 interferes with L'_{1e} of L_1. Then the interference result of light L'_{2o} and L'_{1e} is measured by a camera at Q.

Next, in this section, we analyze the light spectrum of interference results and show its relationship with the angle θ.

2.3.1 Interference Analysis

From Eq. (2.4), we can know that the interference light intensity relies on the two coherent light intensity and their phase difference. We analyze the intensity and phase difference of L'_{1e} and L'_{2o}

in the following part.

2.3.1.1 Intensity

Assume the angles between the optic axis of S and the transmission axes of two polarizers P_1 and P_2 are ϕ_1 and ϕ_2, respectively. Denote the intensity of L_1 as I_1, and assume light rays L_1 and L_2 have equal intensity. According to Eq. (2.3), $I_{1o} = I_1 \sin^2 \phi_1$ and $I_{1e} = I_1 \cos^2 \phi_1$.

Denote the light intensities of L'_{1e} and L'_{2o} as I'_{1e} and I'_{2o}, respectively. According to Eq. (2.1), I'_{1e} and I'_{2o} can be calculated as

$$\begin{aligned} I'_{2o} &= I_{1o} sin^2 \phi_2 = I_1 sin^2 \phi_1 sin^2 \phi_2 \\ I'_{1e} &= I_{1e} cos^2 \phi_2 = I_1 cos^2 \phi_1 cos^2 \phi_2. \end{aligned} \tag{2.5}$$

2.3.1.2 Phase Difference

As shown in Figure 2.2, the incident angles of L_1 and L_2 to S are both θ, the thickness of S is d, and the refraction angles of L_{1e} and L_{2o} are θ_e and θ_o. The optical path difference Δ of L_{1e} and L_{2o} at point Q can be calculated as

$$\begin{aligned} \Delta &= \overline{FA} n_{air} + \overline{AD} n_e - \overline{BD} n_o \\ &= d(tan\theta_o - tan\theta_e)(sin\theta) n_{air} + \frac{d}{cos\theta_e} n_e - \frac{d}{cos\theta_o} n_o \end{aligned} \tag{2.6}$$

where $\overline{FA}$, $\overline{AD}$, and $\overline{BD}$ are the lengths from F to A, from A to D, and from B to D, respectively.

Combining Eq. (2.2) and Eq. (2.6), we have

$$\Delta = d(n_e cos\theta_e - n_o cos\theta_o) \tag{2.7}$$

As aforementioned, for a particular material, n_o is usually fixed, n_e and θ_e are related to the incident angle. We put the details of calculating n_e, θ_e and Δ in Section 2.3.1.3. Therefore, we have

$$\Delta = d(\sqrt{N_e^2 - sin^2\theta(sin^2\gamma + \frac{N_e^2}{N_o^2} cos^2\gamma)} - \sqrt{N_o^2 - sin^2\theta}) \tag{2.8}$$

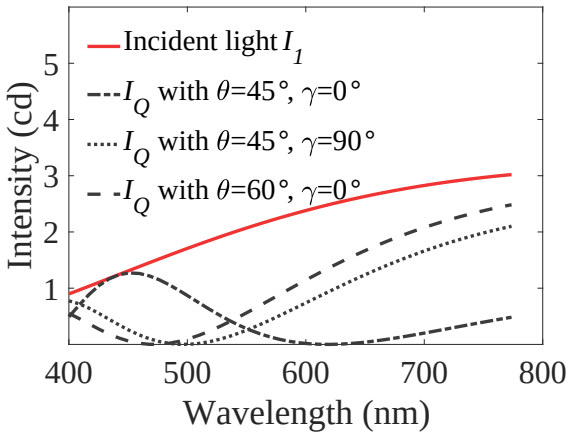


Figure 2.4: Intensity of interference light for different wavelength with different incident angles.

where N_o and N_e are principal refractive indices of S, which are fixed given a certain type of material, θ is the incident angle, and γ is the angle between the projection of incident light on the incident plane and optic axis, which is shown in Figure 2.1.

The optical path difference is for two light beams, the phase difference is different for different wavelength. For light with a specific wavelength λ, we can calculate the phase difference δ of L_{1e} and L_{2o} at point D as

$$\delta_D = \Delta\frac{2\pi}{\lambda}. \tag{2.9}$$

Due to the phase difference of projection on P_2, the phase difference between two coherent lights L'_{1e} and L'_{2o} at point Q is

$$\begin{aligned}\delta &= \delta_D + \delta' \\ &= \begin{cases} \Delta\dfrac{2\pi}{\lambda} & \text{(case 1)} \\ \Delta\dfrac{2\pi}{\lambda} + \pi & \text{(case 2)} \end{cases}\end{aligned} \tag{2.10}$$

where case 1 means the vectors L'_{1o} and L'_{1e} are in the same direction on P_2, and case 2 means they have reverse directions.

2.3.1.3 Calculation of n_e, θ_e, and Δ

Inspired by [33], as shown in Figure 2.1, the directional vector of optical axis, ordinary ray, and extraordinary ray in the birefringence are

$$e_a = (cos\gamma, sin\gamma, 0) \tag{2.11}$$

$$e_{ko} = (sin\theta_o, 0, cos\theta_o) \tag{2.12}$$

$$e_{ke} = (sin\theta_e, 0, cos\theta_e) \tag{2.13}$$

We assume the angle between optic axis and extraordinary ray is α, i.e. angle between e_a and e_{ke}. So according to (2.11),(2.13), we have:

$$cos\alpha = e_a \cdot e_{ke} = cos\gamma sin\theta_e \tag{2.14}$$

Because the refractive index of extraordinary ray varies with different incident angles, according to the relationship between α and the refractive index of extraordinary ray n_e in [34], we have

$$n_e = \frac{N_o N_e}{\sqrt{N_o^2 sin^2\alpha + N_e^2 cos^2\alpha}} = \frac{N_o N_e}{\sqrt{N_o^2 + (N_e^2 - N_o^2)cos^2\alpha}} \tag{2.15}$$

where N_o and N_e are principal refractive indices and are fixed for each type of material. According to (2.14),(2.15), we have:

$$n_e = \frac{N_o N_e}{\sqrt{N_o^2 + (N_e^2 - N_o^2)cos^2\gamma sin^2\theta_e}} \tag{2.16}$$

According to Snell's Law, we have:

$$n_{air} sin\theta = n_e sin\theta_e = n_o sin\theta_o \tag{2.17}$$

where $n_{air} \approx 1$ is the refractive index in air. Then we have

$$n_e = \frac{sin\theta}{sin\theta_e}. \tag{2.18}$$

According to (2.16),(2.18), we have:

$$\theta_e = \arcsin \sqrt{\frac{sin^2\theta}{N_e^2 - sin^2\theta(\frac{N_e^2}{N_o^2}cos^2\gamma - cos^2\gamma)}} \tag{2.19}$$

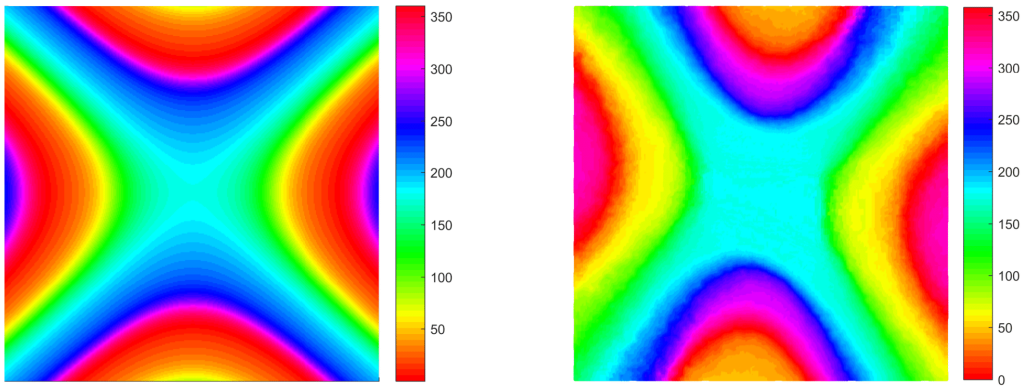


Figure 2.5: (a) Hue values on x-y plane by simulation, (b) Hue values measured by mobile phone on x-y plane.

Finally, according to (2.18) and (2.19), we have:

$$n_e = \sqrt{N_e^2 - sin^2\theta(\frac{N_e^2}{N_o^2}cos^2\gamma - cos^2\gamma)} \tag{2.20}$$

Because the optical path difference is:

$$\Delta = d(n_e cos\theta_e - n_o cos\theta_o) \tag{2.21}$$

We substitute n_e, θ_e, and n_o, θ_o into Eq. (2.21), we can have the expression of Δ using known parameters:

$$\Delta = d(\sqrt{N_e^2 - sin^2\theta(sin^2\gamma + \frac{N_e^2}{N_o^2}cos^2\gamma)} - \sqrt{N_o^2 - sin^2\theta}) \tag{2.22}$$

2.3.1.4 Summary

According to Eq. (2.4), the intensity spectrum of the interference light at Q can be calculated as

$$\begin{aligned} I_Q = {} & I_1 cos^2\phi_1 cos^2\phi_2 + I_1 sin^2\phi_1 sin^2\phi_2 \\ & + 2I_1 cos\phi_1 cos\phi_2 sin\phi_1 sin\phi_2 cos\delta. \end{aligned} \tag{2.23}$$

where δ can be calculated according to Eq. (2.10).

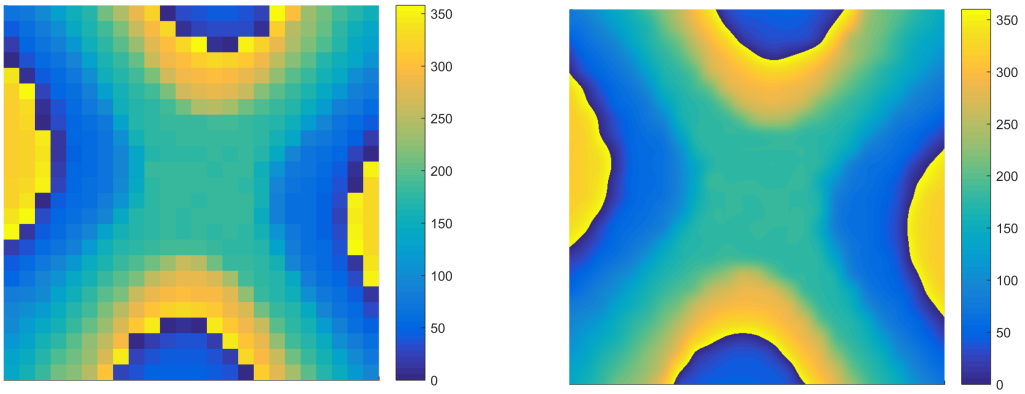


Figure 2.6: (a) Hue matrix sampled, (b) Hue matrix after interpolation.

According to Eq. (2.23), given the intensity spectrum on frequency domain of light source I_1, the angle ϕ_1 between optic axis of the birefringence material and the polarizer P_1, the angle ϕ_2 between optic axis of the birefringence material and the polarizer P_2, the incident direction parameters θ and γ, and birefringence material parameters principal refractive indices and thickness d, we can calculate the value of the light intensity I_Q at Q.

Figure 2.4 shows the light spectrum of interference for different parameters. Given the value of I_1, ϕ_1, ϕ_2 and d, different combinations of θ and γ result in different spectrum of I_Q. This makes the foundation of obtaining light incident angles based on different interference results. As long as we can get the incident angles from multiple points, we can use the AoA-based method for localization.

2.3.2 Validation

2.3.2.1 Choose The Light Spectrum Feature

Mobile cameras usually do not have the capability of measuring the light spectrum directly. However, the direction is represented by the interference light spectrum, and we have to distinguish different light spectrums to distinguish different directions. There is a challenge for us to find a proper light feature, which satisfies two conditions in the meantime: it can be measured by the COTS camera and can indicate the direction from the source to the chip. It is well-known that

different light spectrums result in different colors of the mixed light. A straightforward approach is to measure the RGB color and map RGB vectors to different directions. However, we find this is not feasible in practice as the spectrum could not be effectively represented in RGB color. Instead, we use the HSL (Hue, Saturation, Lightness) color space and find that the H (i.e., Hue) component from HSL is much more suitable for representing the color of mixtures of lights [35].

2.3.2.2 Measurement Result

We conduct an experiment to validate the model. We measure the hue value on different positions after P_2. Figure 2.5b shows the measurement hue values for different positions on a plane with a certain distance to the light source. Then we compare the measurement result with the simulation result based on Eq. (2.23). In our simulation, we use the parameters of quartz crystal (a type of birefringence material) chip with thickness of 0.6 mm. We measure the intensity spectrum of interference result on different direction. We leverage the color wheel [35] to approximate intensity spectrum with hue value. Figure 2.5a shows the hue value with respect to positions on a surface parallel with the birefringence chip. We can see that the color regularities of Figure 2.5a and Figure 2.5b are very similar. This coincides with our analysis and Eq. (2.23). This also means that hue value is effective for representing the intensity spectrum.

2.4 RainbowLight Design

2.4.1 Design Overview

Figure 2.7 illustrates the system overview of RainbowLight. The chips used in RainbowLight are a combination of two polarizers and one birefringence chip as shown in Figure 2.2. With one chip, we can calculate direction information. Combining the direction information from multiple chips, we can derive the 3D location. The main design of RainbowLight consists of two parts. The first part is mapping initialization. This part is to build an initial mapping between the direction and hue value for a certain type of chip. The mapping initialization only needs to be performed once for a certain type of chip. The second part is the 3D localization component. In this part, a mobile

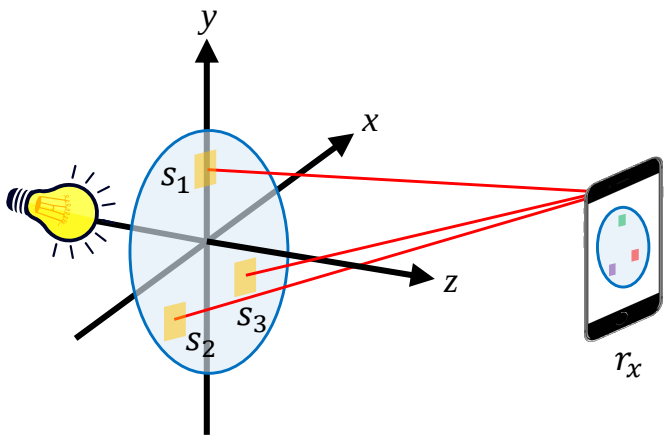


Figure 2.7: Overview of RainbowLight.

camera will take a photo containing multiple chips. Based on the hue value of the initial mapping, the direction to those chips can be calculated. Then we also propose a direction intersection based method to calculate the final 3D location.

2.4.2 Mapping Initialization

The mapping between light directions and hue values can be built by sampling in different positions. We put a chip at the origin O of the coordinate system, and the chip is parallel with the x-y plane. A mobile phone moves in a grid at a certain plane ($z = 1m$) and captures a photo containing the chip at each position. For a sampling position r, it derives the hue value h of the color for the chip from the captured photo. It means that the hue values for all points on the ray $\overrightarrow{Or}$, i.e., the ray with the direction from the chip to the position on the plane in the space, are h.

Therefore, we build a map $R_S \longrightarrow H_S$ from sampling positions $R_S = (r_1, r_2, \ldots, r_n)$ to hue values

$$H_S = \left(\ h_1, h_2, \cdots, h_n \ \right) \tag{2.24}$$

where h_i denotes the hue value observed by mobile phone from points on line $\overrightarrow{Or_i}$.

For a higher sampling density, the map should be more accurate. On the other hand, a higher density also indicates a higher sampling overhead. To reduce the initial sampling overhead, we propose an interpolation-based method to improve the granularity of initial map. We leverage the color regularity to interpolate a coarse-grained sampling matrix H_S and build a fine-grained

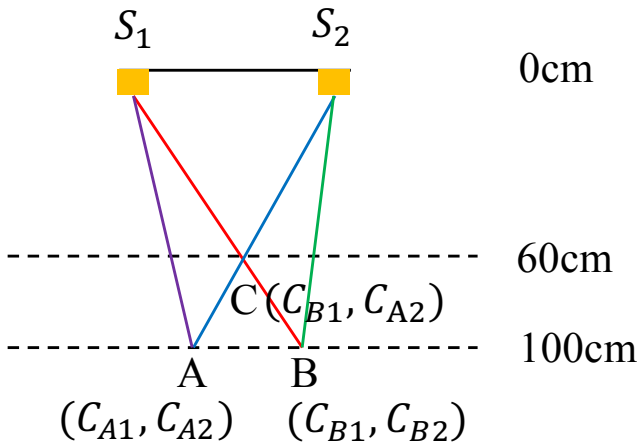


Figure 2.8: Illustration of localization algorithm.

map $R \rightarrow H$. We examine the performance of interpolation under different sampling density in Section 2.7.3.

As shown in Figure 2.5, the color gradually changes with the position. As the hue value ranges from 0 to 360, in interpolation we should carefully deal with the hue value cross the hue range boundary. More specifically, for two hue values h_1 and h_2 ($h_1 > h_2$) for two adjacent sampling positions, we first calculate the hue value gap $h_\Delta = h_1 - h_2$. If h_Δ is smaller than a pre-defined threshold thr (e.g., $thr = 350$), the interpolation can be performed between h_1 and h_2. If h_Δ is larger than the pre-defined threshold thr, we consider the hue value between those two sampling positions crosses the hue value boundary. The interpolated hue value should be performed for h_1 and $h_2 + 360$. All the hue value should be calculated from the interpolation result modulo by 360 to guarantee the hue values are in $[0, 360)$. Figure 2.6a shows the original hue matrix. Figure 2.6b shows the interpolation result.

In practice for the same type of chip, we only need to build the initial map $R \rightarrow H$ once. This could significantly reduce the initialization overhead for RainbowLight. Later, we will show how to leverage the map for localization in 3D space.

2.4.3 3D Localization

2.4.3.1 Localization Design

To enable 3D localization, we simply stick several chips on a transparent surface. Without loss of generality, we assume three chips S_1, S_2 and S_3 are used. Later, in Section 2.7, we will show the impact of number of chips. Denote the position of the center of S_1, S_2, and S_3 as p_1, p_2 and p_3, respectively. The position p_1, p_2 and p_3, namely reference points, can be measured in advance.

A mobile phone with a camera at the position r_x simply captures a photo containing S_1, S_2, and S_3. We calculate the hue values $\tilde{h}_1$, $\tilde{h}_2$ and $\tilde{h}_3$ from the photo for those three chips. Based on the initial map between colors and directions, RainbowLight can obtain the possible directions from p_1, p_2 and p_3, respectively. Thus we have three groups of ray directions from three reference points, respectively. Then we can obtain the position r_x based on the intersection of those ray directions.

2.4.3.2 Intersection Based Localization

The goal of localization is to calculate the position r_x based on $\tilde{h}_1$, $\tilde{h}_2$ and $\tilde{h}_3$ and $R \rightarrow H$.

Find line group candidates: The initial map is built using a chip at coordinate origin O. In practical, chips are usually attached at other positions. In order to make the map $R \rightarrow H$ suitable for the deployment of a specific chip, we need to do coordinate translation for the initial mapping. The map becomes $R^j \rightarrow H$ for $j = 1, 2, 3$, where $R^j = R + p_j$ is the transformed sampling position for S_j.

Due to the color error for the camera on a mobile phone, there may be multiple lines with hue close to $\tilde{h}_1$, $\tilde{h}_2$, and $\tilde{h}_3$. Meanwhile, according to Eq. (2.23), we also find that there are multiple combinations of θ and γ leading to the same hue value. It indicates that there may be multiple directions of the same hue value. Therefore, for each chip, we can calculate a group of lines. Overall, we obtain three groups of lines denoted by G_1, G_2, and G_3. We have $G_j = \{\overrightarrow{r_i^j p_j} || h_i - \tilde{h}_j | < \epsilon_h\}$ for $j = 1, 2, 3$ where $r_i^j \in R^j$ and ϵ_h is the maximum allowed hue error.

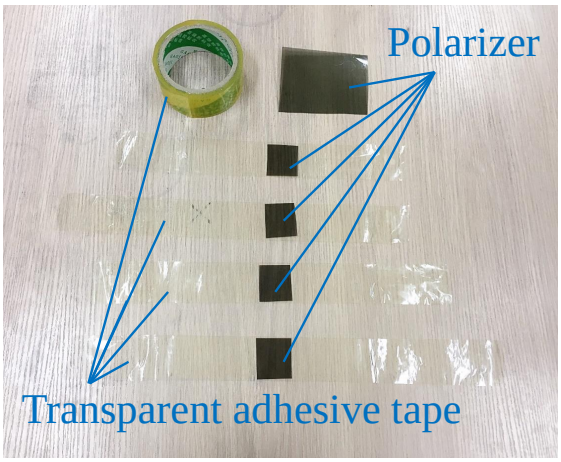


Figure 2.9: Chips in RainbowLight.

Line intersection: The main idea is calculating the localization based on the intersection point of those three sets of lines G_1, G_2, and G_3 as the localization result r_x. There should exist three lines from G_1, G_2 and G_3, respectively, that intersect at point r_x. Due to hue value measurement error, those three lines may be very close to each other but not directly intersect in practice. Therefore, we could use an algorithm based on the contrary thinking. The idea is based on the principle that light travels in a straight line. As shown in Figure 2.8, not without generality, suppose we want to perform localization in a 2D plane. If we put two chips, namely S_1 and S_2, at localization 0 cm and perform initialization at 100 cm, i.e. at point A we observe S_1 and S_2 and get hue values C_{A_1} and C_{A_2} respectively, and at point B we get hue values C_{B_1} and C_{B_2}, according to the principle, we will get hue values C_{B_1} and C_{A_2} at point C at 60 cm. Therefore, if we have sampled all points at 100 cm, hue values of nearly all points in the plane will be derived ideally. We regard all those hue values as 2D coordinates. After that when we capture a photo contains those two chips, we extract hue values, for example, (C_1, C_2), then we can get the final positioning result by calculating the minimum distance between (C_1, C_2) and all coordinates we derived. We can easily extend the algorithm above from 2D plane to 3D space.

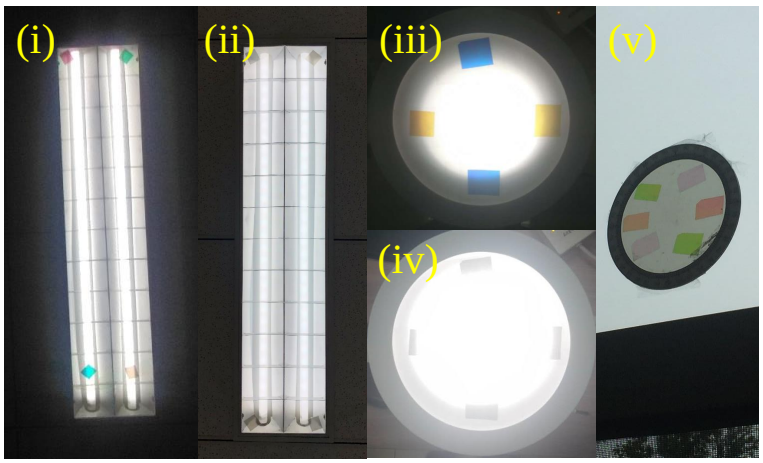


Figure 2.10: Anchor with chips made by two polarizers and one transparent adhesive tape (i): near to fluorescent (iii) on LED lamp cover, anchor with chips made by one polarizer and one transparent adhesive tape(ii): near to fluorescent (iv): on LED lamp cover, (v): anchor on a glass window.

2.5 Implementation

RainbowLight consists of two components: anchor and receiver. In this section, we present the details of those two components. We also discuss a variant of RainbowLight, which put polarizer P_2 in front of the camera to eliminate color observed by human eyes. Since RainbowLight performs relative localization for a given anchor, it needs to identify which anchor is captured by camera hence can be used in a large region. We also discuss how to provide identifiers to anchors in this section.

2.5.1 Anchor

The anchor of RainbowLight is composed of a group of chips. Each chip consists of two linear polarizers and a thin birefringence material chip. We stick the birefringence material chip between two linear polarizers. As shown in Figure 2.9, we use the everyday transparent adhesive tape as the birefringence material. RainbowLight does not require to stick the anchors on a lamp. We can put anchors on different surfaces as long as light can pass through the chips. For example, as shown in Figure 2.10 (i), (iii) and (v), we put anchor near lamps or on a lamp cover or a window. As shown in Figure 2.10(i) and (iii), despite chips display colors, each chip made by polarizers and transparent

adhesive tape is very small. It would not disturb human eyes. To enable RainbowLight, we also need to record the relative position for those chips.

2.5.2 Receiver

We use the smartphone as the receiver side. The camera can capture a photo containing the anchor. We implement software on the mobile phone based on Android. While the camera is taking a photo, RainbowLight launches automatic exposure to fit the luminance of the environment. After obtaining the photo, we use the algorithm of white balance to eliminate color shift among different camera models, then use OpenCV to localize the position of each chip in the image based on features such as shapes and derive HSL information from the photo. To address hue value estimation error in practice, we use the averaged hue value for each chip as the hue value for localization. Then we use the 3D localization algorithm mentioned in section 2.4.3.2 to get the position of the camera.

Now we present a variant of RainbowLight to eliminate the color which can be observed by human eyes directly. We put polarizer P_2 in front of the camera. In such a case, human eyes cannot observe the color displayed by chips directly as shown in Figure 2.10 (ii) and (iv), but cameras can capture chips with different colors. However, if we put P_2 in front of the camera, the camera's rotation would result in the change of color of the chips, thus color-direction map could not be used. Fortunately, since the hue value instead of RGB represents color in RainbowLight, chips only show two complementary hue values with the camera's rotation as shown in Figure 2.11. Therefore, we measure the camera's rotation angle firstly, if it results in complementary hue values of initialization, we can transform them into original hue values hence performing localization. Attaching polarizer in front of the camera will bring in extra costs, and brings error of accuracy with the camera's rotation. We will present the accuracy in Section 2.7. Users who deploy the RainbowLight can choose where to put the polarizer P_2 according to their conditions and requirements.

We measure the latency of RainbowLight. In the measurement, we let RainbowLight process 10 photos to measure the average latency. The mobile phone we used is Huawei Nexus 6P. It takes 236 ms on average to find chips and extract hue values. It takes 503 ms on average for 3D localization

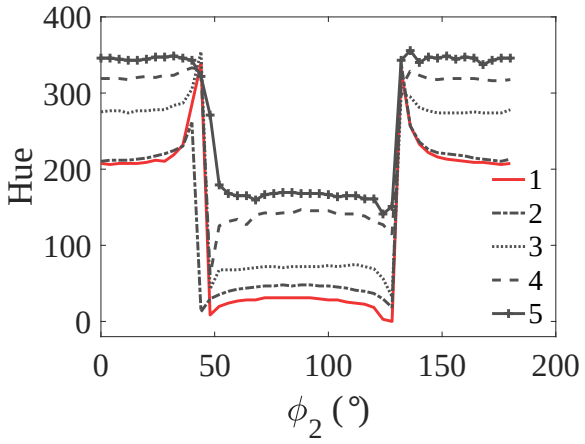


Figure 2.11: Complementary hue observed as rotating mobile phone for different tape thickness (1 ~ 5).

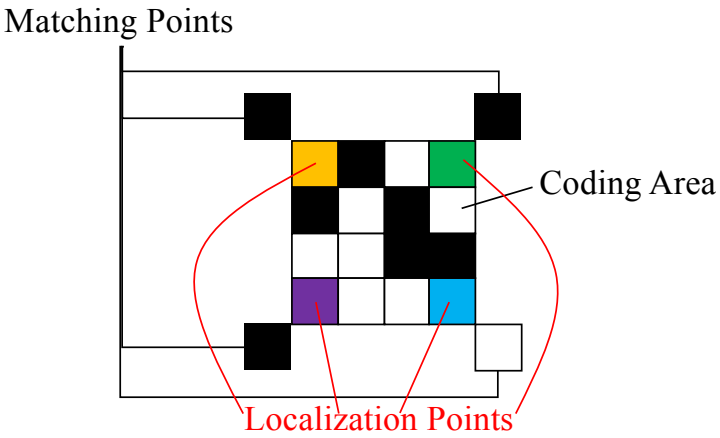


Figure 2.12: RainbowLight anchor with identifier.

from hue values. We optimize RainbowLight 3D localization to parallel the processing in our implementation of localization. With such an optimization, the time for 3D localization reduces to 123 ms on average. This would apply to most VLP based applications such as navigation. We also use Power Monitor to measure the power consumption of RainbowLight with Nexus 5x, the result shows that our algorithm takes 1.122J to process one photo and perform localization.

2.6 Apply RainbowLight to Localization in a Large Area

We have presented a novel relative localization approach, RainbowLight, which can derive the camera's relative position to an anchor. However, one small RainbowLight anchor only can be

captured by the camera in a small region, thus it is difficult to apply to localization in a large area such as a shopping mall. To address this issue, we can use the idea similar to use multiple lamps to illuminate an entire room, in other words, we give each anchor a unique identifier and extract the identifier from the anchor firstly to get a coarse-grained area where camera located, then derive precise relative location to the anchor. Therefore, we can use RainbowLight to get the camera's location in a large area.

2.6.1 Providing Identifier to RainbowLight Anchor

We can use the existing method such as iLAMP [22] to distinguish different light sources in a large area if we put an anchor on the lamp. We can also attach the QR code on each anchor to identify them. Considering iLAMP cannot be used with light-off, we also design a QR-code-like method to use our localization chips for providing ID.

As shown in Figure 2.12, after modification, an anchor consists of 3 components. While the *Localization Points* used to derive the relative position is made by polarizers and transparent adhesive tape, *Matching Points* and *Coding Area* are only made by polarizer. We make them by two perpendicular polarization directions. Similar to the QR code, 3 of matching points are in the same direction, and another is not, therefore, it can be decoded even if the anchor is rotated in the photo. We use those two directions to represent 0 or 1 in the coding area. Therefore, after taking a photo behind another polarizer either covered on the anchor or put before the camera, we can compare the brightness of each polarizer in the coding area to polarizers of matching point to recognize each of them representing 0 or 1, hence decode the identifier. In this case, the anchor can encode $2^{12} = 4096$ identifiers in the coding area.

2.6.2 Localization in a Large Area

As Figure 2.13 shown, without loss of generality, suppose we have 3 anchors in a large area, we can store each anchor's identifier and its real position in a database in advance. During the localization process, for example, after a camera in area #3 which is a valid area of anchor #3 taking a photo

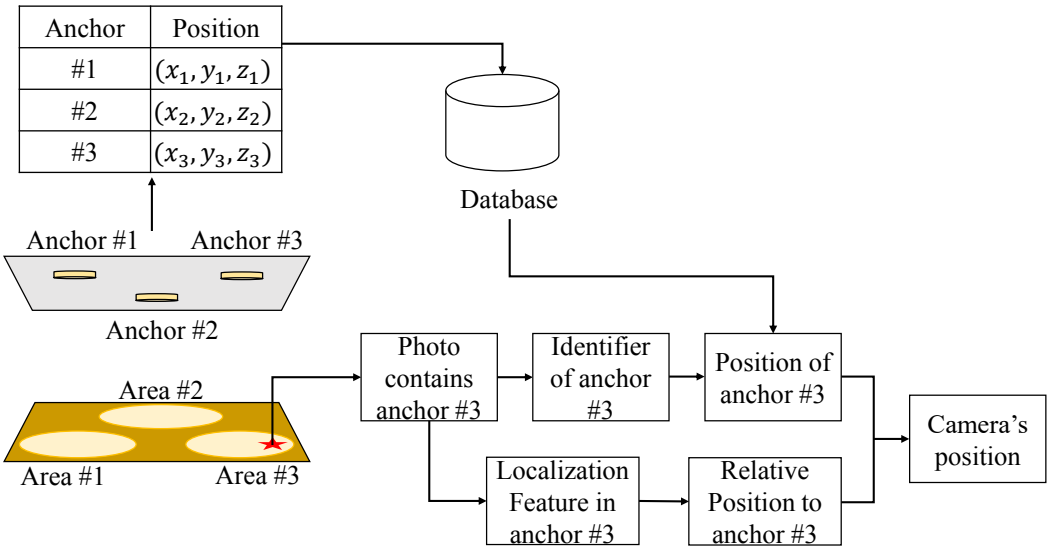


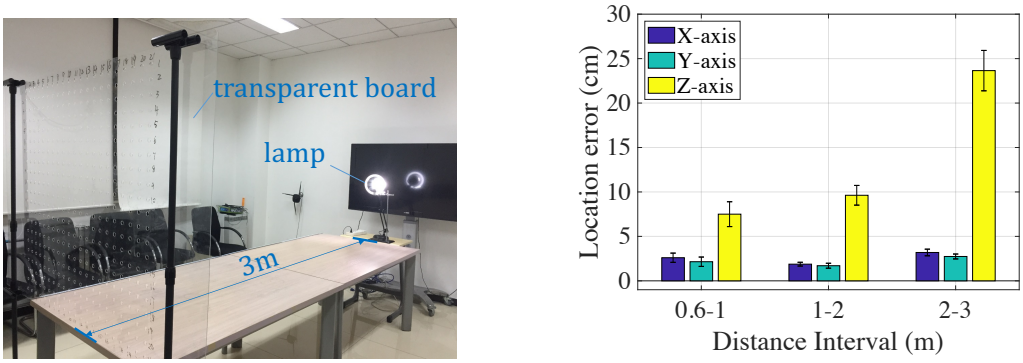
Figure 2.13: Overview of localization in building.

containing anchor #3, the system firstly decodes the identifier of the anchor in the photo, then get the real position of the anchor from the database. Combining with the relative position from the camera to the anchor, we could get the camera's real position.

2.7 Evaluation

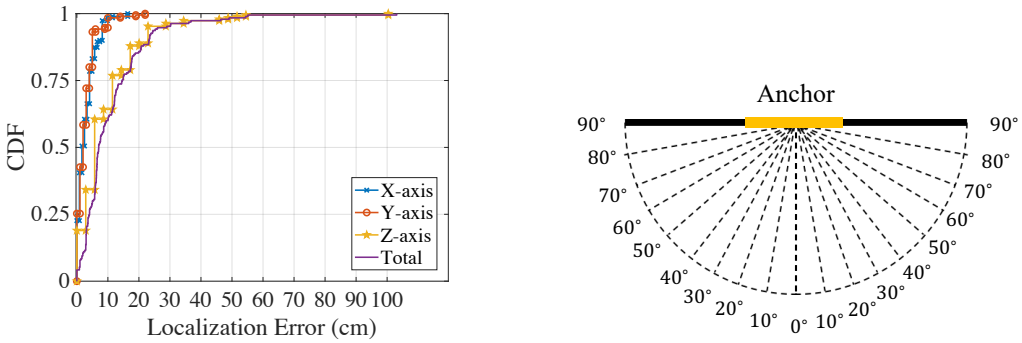
We evaluate the performance of RainbowLight from the following aspects:

- Localization accuracy for different distances.
- The performance of mapping the position related to the landmark to the absolute position.
- The impact of system parameters on localization accuracy.
- System performance under different light sources (different manufacturers, color temperatures, lamp types, and powers).
- System performance under different mobile phone models.
- System performance with the light on/off.
- System performance with different angles of mobile phone orientation.



(a) (b)

Figure 2.14: (a) Experiment environment. (b) Localization precision on different distance.



(a) (b)

Figure 2.15: (a) Localization precision map relative position to absolute position. (b) Capture in different angles.

Through the evaluation, we aim to show the effectiveness of RainbowLight in practice. It should be noted that for all experiments we use the same initial mapping unless otherwise specified. This means that we only need to perform initialization once, which significantly reduces the initialization overhead compared with existing approaches.

2.7.1 Localization Accuracy

Figure 2.14a shows the experiment environment. In the experiment, we move a transparent board to different distances to the light source. For each distance, we move the mobile phone on the board at different positions. We can measure the position of the mobile phone on the board as the ground truth. Meanwhile, we also use RainbowLight to calculate the position of the mobile phone. We

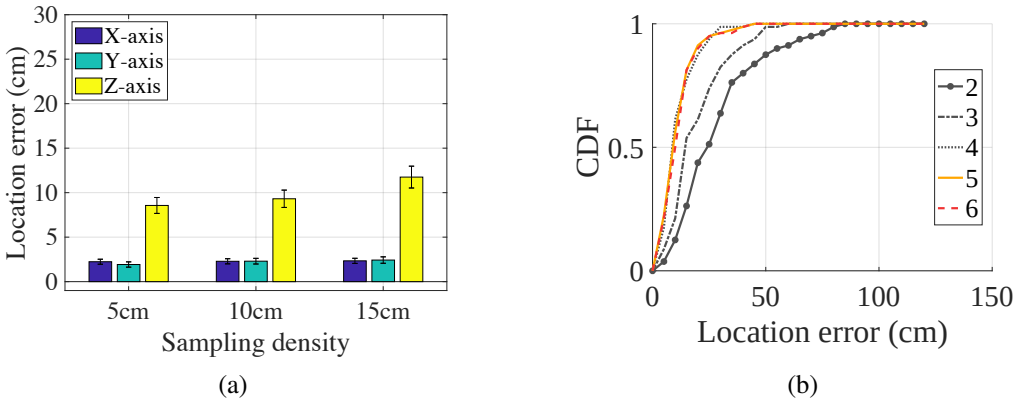


Figure 2.16: (a) Localization precision on different sampling density. (b) Localization precision on different number of chips.

switch off other lamps during our experiment at night. Figure 2.14b shows the localization error for the mobile phone moving on the board out of 230 random points. The x-axis denotes the range of the distance between the board and the lamp. We can see that the localization error increases as distance increases. This is mainly because hue value is less sensitive to the position for a larger distance.

We can also observe that the error on z-axis is larger than that on x-y plane. The major reason is that the angle from the chip to the mobile phone varies by a smaller value when we move the mobile phone along the z-axis than that along the x-y plane. This phenomenon is more evident when chips are close to each other. However, even when those chips are all in a circle with diameter less than 16 cm, the localization accuracy for different distance is still high. This indicates RainbowLight can work for different distance with the lamp of small size.

Overall, in the 2m - 3m distance interval, the mean error of localization is 3.19 cm on x-axis, 2.74 cm on y-axis, and 23.65 cm on z-axis. This performance is better than SmartLight with a localization error of about 60 cm on z-axis for distance from 1m - 3m. The localization accuracy of RainbowLight is enough for most of today's application scenarios such as navigation.

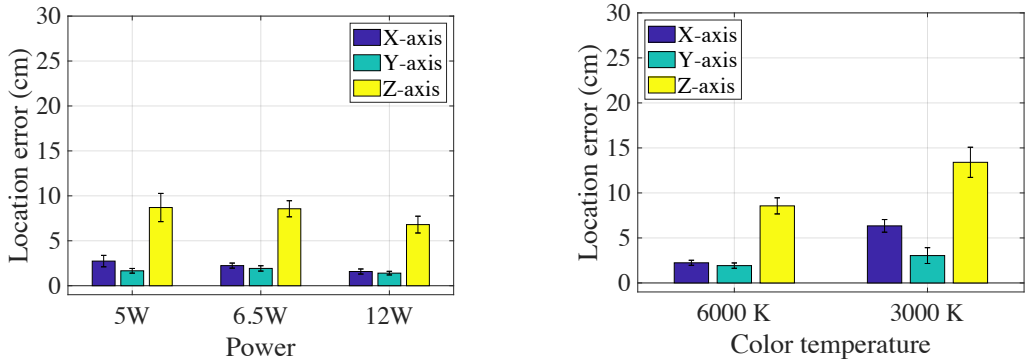


Figure 2.17: Localization accuracy for different (a) power of lamp, (b) color temperature of lamp.

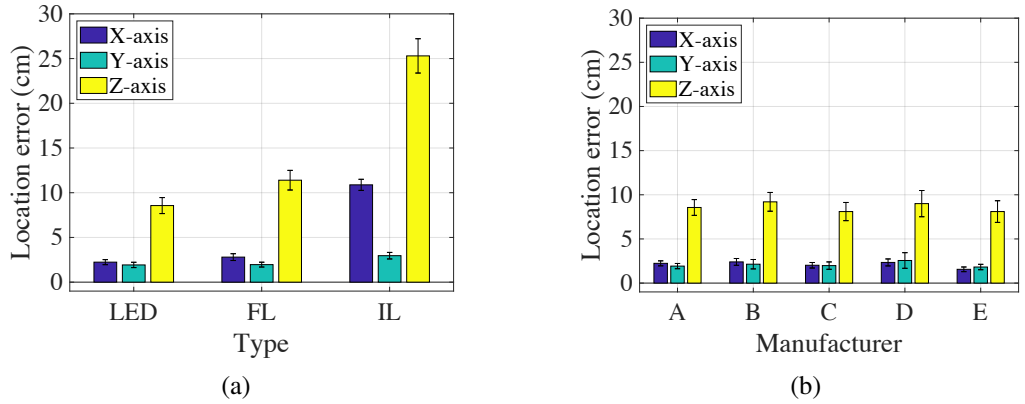


Figure 2.18: Localization accuracy for different (a) types of lamp, (b) manufacturers of lamp.

2.7.2 Performance with Identifier

As discussed in section 2.6, we design an approach to map the position of a camera relative to the anchor to the absolute position in an area by providing an identifier to each anchor. To evaluate the performance of this approach, we randomly choose 190 points in an area of the meeting room, and calculate the accuracy of localization. Figure 2.15a shows the performance. We can find that RainbowLight achieves 1.68 cm of the median error in the X-axis, 2 cm of the median error in the Y-axis, 5.74 cm of the median error in Z-axis, and 7.04 cm of the median error with the whole dimension. It also achieves 7.37cm, 5 cm, 22.9 cm, 23.20 cm of the 90% error in X-axis, Y-axis, Z-axis, and with the whole dimension, respectively. The localization accuracy is also enough for most of today's application scenarios.

To evaluate the performance of decoding of the identifier on the anchor, we use the camera to capture photos with different angles to the anchor. As shown in figure 2.15b, we put an anchor in a plane, and use the camera to capture photos from 0° to 90°, then try to decode the identifier on the anchor. With the identifier we designed in section 2.6, it could not be decoded when the angle is above to 60°. Since the ceiling of a room is often with a height of 3 m, so users should deploy an anchor in every 9.42 m^2 with the code we designed in section 2.6.

2.7.3 Impact of Sampling Density

We examine the impact of sampling density in building the initial map. Figure 2.16a shows the localization accuracy with respect to different sampling densities. We build the initial map on a plane parallel to $x - y$ plane with $z = 100$ cm. We examine the performance with different inter-distance of sampling position, i.e., 5 cm, 10 cm, and 15 cm, respectively. It can be seen that low sampling density still works well for RainbowLight. Even when the inter-distance is 15 cm, the localization error is only around 10 cm. This is mainly because hue value distribution is smooth in the 3D space and thus interpolation is effective in building initial mapping.

2.7.4 Impact of Number of Transparent Chips

As shown in Section 2.4, the hue value from a single chip determines a candidate group of rays from the chip. With more chips, the localization accuracy will be improved as the intersection point can be refined with more groups of rays. We explore the relationship between localization accuracy and the number of chips. Figure 2.16b shows the CDF of 3D localization error while increasing the number of chips from 2 to 6. It can be seen that the localization accuracy increases when the number of chips increases from 2 to 4. Further, the performance becomes relatively stable when the number increases from 4 to 6. This means 4 chips is enough in practice to achieve a good localization accuracy.

Figure 2.19: Different light sources.

2.7.5 Impact of Different Light Sources

We examine the performance of RainbowLight with different light sources. As shown in Figure 2.19, we use lamps of different types, i.e fluorescent (FL), LED and incandescent bulb (IL), from different manufacturers (A - E), with different color temperature (3000 K, 6000 K) and different power (5 W, 6.5 W, 12 W). In all the following experiments, we use a Philips (manufacturer A) 6.5 W LED with the color temperature of 6000 K for initialization.

In our daily life, the power of LED mainly ranges from 5 W to 20 W. Figure 2.17a shows localization error of LED (manufacturer A) of power 5 W (500 lm), 6.5 W (600 lm), and 12W (1100 lm) out of 150 random points. There is no significant difference in terms of error for different power. This is mainly because as long as γ and θ are fixed, our approach captures the major property of light spectrum and also removes other noise such as brightness, as explained in Section 2.3.

There are mainly two different color temperatures (6000 K and 3000 K) for typical lamps in our daily lives. Intuitively, 6000 K generates white color while 3000 K generates yellow. The light spectrums from those two temperatures are slightly different. We initialize with a 6000 K lamp and measure the localization error for 3000 K and 6000 K out of 100 random points. As shown in Figure 2.17b, we can see that the localization error of 3000 K is slightly larger than that of 6000 K

because of spectrum difference. However, the accuracy of both color temperature is still acceptable. In practical applications, we only need to build the initial map with one color temperature, and RainbowLight performs well under other color temperatures.

We examine the performance of RainbowLight for the three most commonly used lamps, i.e., LED, fluorescent, and incandescent bulb out of 150 random points. As shown in Figure 2.18a, the accuracy for fluorescent is high. The accuracy of the incandescent bulb is relatively low. This is because those two types of lamps have different light spectrums. However, as long as we use the incandescent bulb for initialization, the accuracy of RainbowLight remains high for incandescent bulb.

We also examine the performance of RainbowLight among different brands of lamps. The light spectrum emitted slightly varies for lamps from different manufacturers. We choose 5 LEDs from 5 different popular manufacturers, marked as A-E. The power of all lamps is 5 W and the lumens are 500 lm, 380 lm, 450 lm, 400 lm, 280 lm, respectively. The color temperature is 6000 K. Figure 2.18b shows that the error is small for all brands out of 250 random points and the performance is similar for all brands. It also indicates we only need to initialize with a certain brand, and the accuracy of RainbowLight is acceptable under other brands.

Summary. RainbowLight achieves a high accuracy under different circumstances with commonly used lamps. For most scenarios, RainbowLight only needs to be initialized once, and almost can be used for all other lamps. This significantly reduces the deployment cost and makes RainbowLight practical.

2.7.6 Impact of Different Mobile Phone Models

Because different cameras have different parameters of light sensors, so they might get different hue values to the same light beam. We use the white balance algorithm to reduce the impact from different parameters of sensors, and examine the impact of different mobile phones. We use two branches of mobile phones, i.e., Huawei Nexus 6P and Vivo X7 to measure the accuracy of RainbowLight. We randomly choose 10 points in the range of z-axis between 100 cm and 150 cm

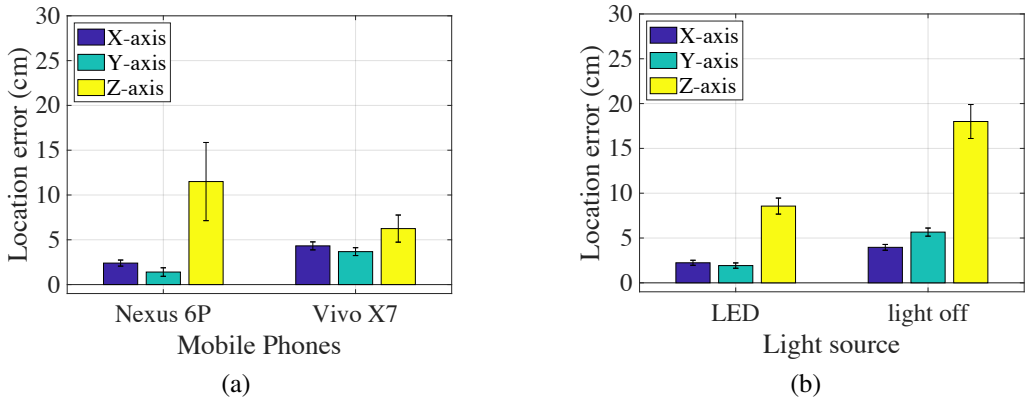


Figure 2.20: Localization accuracy for different (a) mobile phones, (b) lamp status.

for each mobile phone, the result is shown in figure 2.20a. We find that the error doesn't change much, so RainbowLight could be used on different mobile phone models.

2.7.7 Localization with Light Off

Most existing visible light positioning systems, e.g., LiTell[1], SmartLight[20], and CELLI[17], only work when the light is turned on, as those systems require modulating information in the light ray or measuring special features from the light ray. This significantly hinders their applications in the daytime when light is usually switched off. RainbowLight can work even when light is switched off during the daytime as it does not need to modulate information in light or measure light features. Figure 2.20b shows the performance of RainbowLight out of 50 random points with the light turned off. Similar to Section 2.7.1, we examine the accuracy in the environment as shown in Fig. 2.14a. In the experiment, sunlight passes through the window and we switch all lamps off. We can see that the error for the light turned off is still less than 20 cm. The error for the light turned off is very small and is similar to the scenario of the light turned on. This is mainly because RainbowLight can generate obvious features from different light sources, and can also effectively extract those features. This significantly extends the application for visible light-based localization and make it more practical in everyday life.

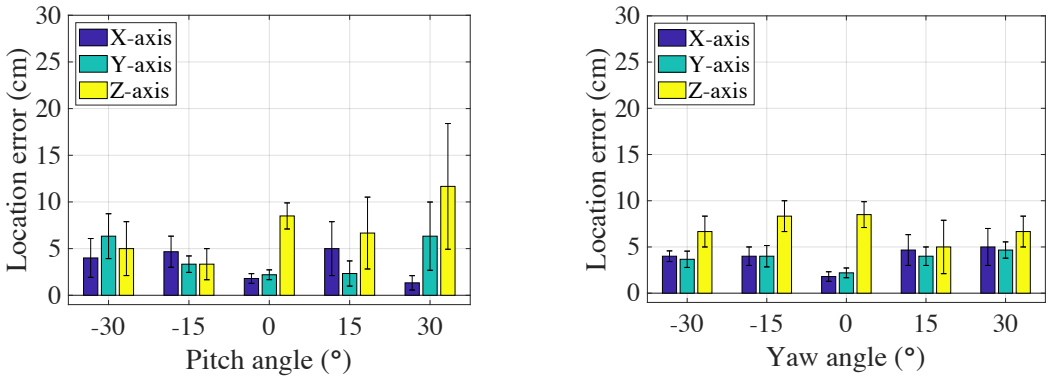


Figure 2.21: Localization precision of different (a) pitch angles, (b) yaw angles.

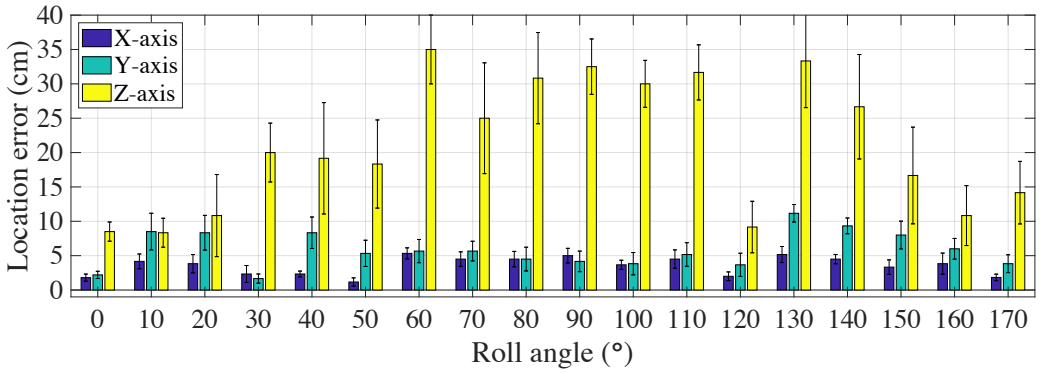


Figure 2.22: Localization precision of different roll angles of mobile phone.

2.7.8 Impact of Mobile Phone Orientation

To verify the influence of pitch and yaw, we measure error at distance 60 cm with different pitch and yaw angles. Figure 2.21a and Figure 2.21b shows the result. We select range from $-30°$ to $30°$ because the mobile cannot capture the lamp with pitch and yaw angle out of this range. We can see that when we change pitch and yaw angle, error changes slightly. This is mainly because when we change the pitch and yaw angle, ϕ_1, ϕ_2, γ, and θ does not change.

If P_2 is attached to the chip, mobile phone roll will have no impact on the hue value. If we put the polarizer P_2 in front of the camera, RainbowLight needs to confirm if chips on anchor show complementary hue value and its impact on localization accuracy. We also examine the accuracy of localization in this scenario. The error of different roll angles of camera as shown in Figure 2.22a.

Therefore, no matter which position we are, as long as we can capture the lamp with any 3D

orientation, RainbowLight shows a high localization accuracy. This extends application scenarios of today's VLP systems.

2.8 Related Work

2.8.1 Visible Light Based Localization

The first category of work is to use a special designed LED light to generate identifiable features [21, 18]. Those works usually need to use an MCU to control the lamp to modulate information by change the frequency, voltage, etc. Spotlight [19] generates a sequence of on/off the pattern and uses such a pattern as landmarks for localization. Spinlight [5] uses a hemispherical shade to encode position information with holes. CELLI [17] designs a structure with LCD to modulate polarization direction of emitting light. It generates two sweeping lines with special light properties and uses sweeping lines for localization.

Recently, SmartLight [20] proposes an interesting idea to use a digital modulated LED array with a lens to achieve single light 3D localization. It modulates different LED lights with different frequency on the LED array. Then it emits the light through a lens to the 3D space. Then it derives the location based on the frequency of received light. Pulsar [28] uses the inherent features of photodiode diversity. It builds a map from angle to RSS. It designs a special receiver with two photodiodes. Most of those approaches in this category require a specially designed lamp or receiver. Thus it may not apply to most scenarios in our daily lives.

Further, many attempts are proposed to remove the requirements with specially controlled light. Existing methods such as [6, 27] use geometrical relationships among lights with the known position for triangulation based localization. PIXEL [24] leverages the inherent feature of optical rotatory dispersion for localization. When a linearly polarized light passes through a disperser, the color observed through a polarizer with different transmission directions should be different at different locations. By fixing the orientation of a mobile phone, [24] derive the identifier by the observed color, then calculates location with the geometrical relationship. It requires to capture more than one light in one photo.

LiTell [1] and iLAMP [22] use inherent features of fluorescent such as frequency and color spectrum to identify each light. Given the position of the light, the location can be derived by triangulation. Those two approaches are very nice as they do not need any extra modification to the lamp. However, they require to sample the features for each light. It is also highly related to the environment and cannot work when a lamp is changed. Recently, [36] proposes an interesting method of using light to correct inertial measurement unit errors. As introduced in [36], it leverages the property that a polarized light ray going through transparent tape is rotated by an amount related to wavelength. Then it tries to derive the location change by sensing the color after a polarizer with different directions. It detects color changes by edge crossing between four types of blocks hence serve as landmarks to correct IMU drift errors.

Luxapose [6] localizes the relative position from lamps. The main idea is to build a geometrical model and calculate the position based on the relationship between lamps' positions both in the real world and in the photo. Such a model is also used in iLAMP [22]. However, the model needs extra-parameters, e.g., focal length or data from other sensors. Since different cameras hold different parameters like the focal length, they are not easy to use. RainbowLight only uses the color pattern to derive the relative position to the tag, which is more general. Travi-Navi [37] using the computer vision-based approach to launch the navigation. It stores guider's video and uses sensors to calibrate the position, and those data can be further used for followers in navigation.

2.8.2 Other Localization Approaches

Localization has attracted many research efforts. Besides visible light based localization, there exist a large collection of localization approaches using wireless signal, such as [11, 12, 13, 38, 14, 39, 40, 41, 42, 15, 16, 43], using acoustic signal [44, 45, 46], using environment information and cell tower signal [47], FM signal [48], stride information [49], inertial sensors [50] etc. Those approaches are usually based on a signal attenuation model or pre-collecting a large number of fingerprints. Meanwhile, many wireless signal based approaches need to analyze signal properties such as CSI, which further leads to a high computation overhead. Thus they usually require specially designed

hardware at the receiver or sender, making it difficult to implement on the mobile phone. Multiple path effect also affects the localization accuracy for many of those approaches. Our approach is largely inspired by those approaches.

CHAPTER 3

PATRONUS: PREVENTING UNAUTHORIZED SPEECH RECORDINGS WITH SUPPORT FOR SELECTIVE UNSCRAMBLING

3.1 Overview

Human beings have long used acoustic signals to exchange information with each other. Human beings now use acoustic signals, which is speech, to exchange information with ubiquitous smart devices such as smartphones, smartwatches, and digital assistants that are equipped with embedded microphones. While these speech detection and recognition capabilities make possible many convenient features, they also introduce many privacy risks such as secret, unauthorized recordings of our private speech [51, 52] that can have real world consequences. For example, the Ukrainian prime minister offered his resignation after an unauthorized recording was leaked [53].

Manufacturers claim that they are trying their best to protect users' privacy, but there is no effective and user-friendly technical anti-recording solution available despite the fact that anti-recording is not a new problem. One existing anti-recording solution is to talk near a white noise source, e.g., near an FM radio tuned to unused frequencies, so that the conversation cannot be clearly recorded. This approach is not user-friendly because the people having the conversation must put up with the white noise that interferes with their normal communication. A similar solution [54] emits high frequency noise near the upper bound of human sensitivity; most people do not notice the interference, but pets and infants may notice it [3], so this solution is not environment-friendly. Electromagnetic interference was an effective anti-recording solution [55] in the past, but modern microphones are immune to electromagnetic interference. Moreover, all of these traditional anti-recording approaches cannot allow authorized devices to clearly record conversations.

Any effective anti-recording solution must provide the following three key properties: (1) normal human conversation should be unaffected by the anti-recording solution meaning the anti-recording solution should not change what humans hear while having a conversation; (2) unauthorized devices

should not be able to make a clear recording of any conversation protected by the anti-recording solution; (3) authorized devices should be able to make a clear recording of any conversation protected by the anti-recording solution.

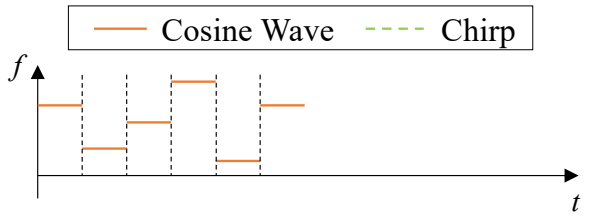
One potential solution that can satisfy all three properties is to generate multiple ultrasonic frequency sound waves because of the following two properties of ultrasonic waves. First, humans cannot hear ultrasonic sound waves. Second, commercial off-the-shelf (COTS) microphones exhibit nonlinear effects, which means that when these microphones receive multiple ultrasonic sound waves, they generate low-frequency sound waves that can be heard by humans and thus interfere with the clarity of recordings made with those microphones [8, 56, 7, 57, 58, 3, 59]. There are three main challenges that must be overcome in order to develop an ultrasonic anti-recording solutions that satisfies the three key properties:

(1) First, any ultrasonic anti-recording solution must defend against potential attacks such as using Short-time Fourier transform (STFT) to analyze unauthorized recordings and using filters to cancel out the low-frequency sound waves that interfere with recording clarity.

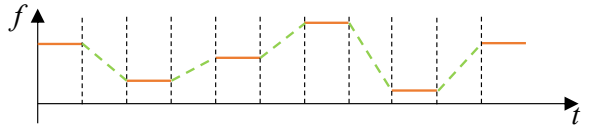
(2) Second, ultrasound travels along a straight line [60], which means a single ultrasonic wave generator can only interfere with recording devices within a limited range of angles from the generator. In practice, it is difficult to design an ultrasonic anti-recording solution that can neutralize all recording devices within a large coverage area.

(3) Finally, the performance of authorized devices could be affected by the ringing effect due to electronic behaviors. Such ringing impulses are hard to be canceled and may remain in authorized recordings, severely downgrading the quality of the descrambled recordings.

In this chapter, we present Patronus, an ultrasonic anti-recording system that satisfies the three key properties. Patronus has two key components: the *scramble* that is the pseudo-noise generated at all microphones, and *descrambling* that is the process to remove the scramble for authorized devices. We form the scramble by randomly picking frequencies from the human voice frequency band and then shifting them to the ultrasonic band. To thwart STFT attacks, we further fine-tune



(a) Discrete frequency scramble components.



(b) Continuous frequency changing scramble with chirps.

Figure 3.1: Using chirps to smooth the frequency changing components of the scramble.

the period of the scramble so that it cannot be easily analyzed and canceled. We add a reflection layer with a curved surface to create a reflected ultrasonic wave that can cover a wider area. Finally, to mitigate ringing effects, i.e., sudden hardware impulses due to discrete frequency changes of current waves, we use chirps to smooth the frequency changing components of the scramble, as shown in Figure 3.1.

Patronus lets authorized devices clearly record audio conversations by sending them the scramble pattern. With scramble pattern, the authorized device applies the Normalized Least-Mean-Square (NLMS) adaptive filter [61] to cancel the scramble and thus produce a clear audio recording of the conversation.

We implement a prototype of Patronus and conduct comprehensive experiments to evaluate its performance. We use the Perceptual Evaluation of Speech Quality (PESQ) [62], the Speech Recognition Vocabulary Accuracy (SRVA, see Section 3.6), and speech recognition error rates (1 - SRVA) to evaluate the performance of Patronus. Our results show that only 19.7% of the words protected by Patronus' scramble can be recognized by unauthorized devices. Furthermore, authorized recordings have 1.6x higher PESQ and, on average, 50% lower speech recognition error rates than unauthorized recordings.

In this chapter, we provide several unique technical contributions when compared to existing works. First, to the best of our knowledge, Patronus is the first system to leverage the nonlinear effect

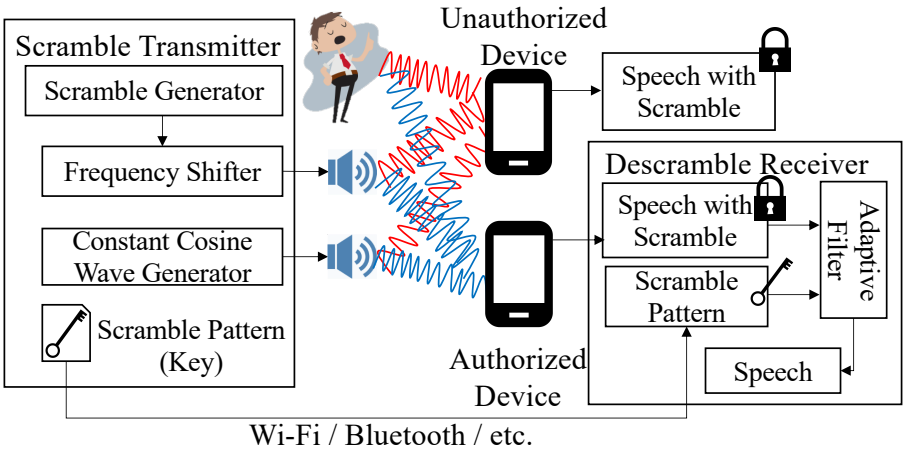


Figure 3.2: System Overview.

of COTS microphones to prevent unauthorized recordings while allowing authorized recordings. Second, we perform a thorough study of the nonlinear effects of ultrasound frequencies including the effects of higher orders whereas recent works[8, 7, 56, 9] only consider the order up to 2. This is critical for descrambling when the signal components with order higher than 2 will likely lie in the human voice frequency band, which means simply cutting off the high frequency components will result in message loss. Instead, our descrambling solution carefully removes these higher order frequencies using an NLMS filter. Third, we mitigate ringing effects by connecting scramble segments with chirps. This simplifies learning the coefficients of impulse response in existing work [8], especially when we deploy multiple ultrasonic transducers in a large space. In general, our contributions are as follows:

- We propose a novel ultrasound modulation approach to provide privacy protection against unauthorized recordings that does not disturb normal conversation.
- We do a thorough study around the nonlinear effect of ultrasound on commercial microphones and propose an optimized configuration to generate the scramble.
- To overcome the fact that ultrasound travels in a straight line, we design a low cost reflection layer to effectively enlarge the coverage area of Patronus in a cost-effective way.
- We present Speech Recognition Vocabulary Accuracy, a new metric to measure the recording

quality. Our experimental results with both PESQ and SRVA show that Patronus effectively prevents unauthorized devices from making secret recordings.

The organization of the rest of this chapter is as follows. Section 3.2 introduces related work. Section 3.3 introduces the nonlinear effect of common microphones, which we analyze more thoroughly than existing works. Section 3.4 presents the design of Patronus. Section 3.5 presents the prototype implementation of Patronus. Section 3.6 presents our evaluation results of Patronus. Section 3.7 discusses the limitations of Patronus and future work.

3.2 Related Works

3.2.1 Nonlinear Effect of Microphones

There has been a lot of research into the nonlinear effect of microphones. For many years, the development of ultrasonic systems on smartphones was restricted due to being limited to a roughly 4 kHz range of frequencies between the high end of human hearing to the cutoff frequency of typical microphones. Furthermore, some infants and pets can actually perceive frequencies within this small band. Roy et.al. [8] performed detailed research on the nonlinear effects of microphones to break through these limitations and expand the working frequency band for ultrasonic systems on smartphones. DolphinAttack [7] leverages the nonlinear effect to generate audio commands that are inaudible to humans. After being recorded by the microphone, the input ultrasonic signals would generate a shadow signal that could be recognized by VCS. Therefore, attackers can perform unauthorized commands without being discovered. SurfingAttack [59] uses oscillation of a surface such as a table to transmit inaudible commands. With this modality, attackers can deploy their speakers in hidden spots such as the back of the surface being used to transmit the secret commands. LipRead [56] extends the attack range by leveraging characteristics of human hearing. It also puts forward a model to filter out such commands generated by the nonlinear effect. Metamorph [57] injects inaudible commands into human-made commands to achieve unauthorized actions. AIC [9] presents a mechanism that fundamentally cancels inaudible commands against

VCS, which we will discuss as an attack model in Section 3.4.2. NAuth [58] uses the nonlinear effect to authenticate devices. Unlike most of these methods, Patronus aims to preserve privacy by adding a removable scramble generated by ultrasonic signals to the recorded human speech. From a technical perspective, Patronus is unique in that it takes into account third and higher order terms from the nonlinear effect. Our experiments show those high order terms can affect recordings whereas most existing methods (e.g., AIC) only consider the second order term and assume the higher order sub-band of the microphone is clean.

3.2.2 Dual Channel Applications

Some applications leverage the difference between humans and devices. For example, human eyes and devices have different perceptions of flicker frequency. Technologies exist that use this phenomena to communicate between the screen and the camera without affecting human vision [63, 64, 65, 66]. Likewise, some technologies modulate acoustic signals in ways that no human can detect to communicate between devices [67, 68].

The difference between the sensitivity of humans and devices is also used in privacy protection. Kaleido [69] protects a movie's copyright by adding a flashing distractor with very high frequency into movie frames that cannot be seen by human eyes. If such a protected movie is subsequently recorded by an unauthorized camera equipped with a rolling shutter, the distractor will be visible on the unauthorized recording because of its high sample rates making the pirated recording a low quality recording. LiShield [2] also uses the Rolling Shutter effect to reduce the quality of photos. Lights with different colors are set to flash in alternating high frequencies that provide normal lighting because human eyes cannot sense the flashing. However, cameras are influenced because the Rolling Shutter samples column by column meaning unexpected color stripes will appear on the photo. In the end, it prevents unauthorized cameras from taking photos. Although Patronus has a similar motivation to prevent unauthorized recordings, Patronus is different from the two papers as it targets acoustics rather than visuals.

3.3 Nonlinear Behavior of Common Microphones

In this section, we provide a brief primer about nonlinearity of common microphones; a more comprehensive introduction can be found in recent papers [8, 56]. Ideally, COTS microphones are linear systems. Given the input signal $s(t)$, the output signal $y(t)$ is expected to be linear combinations of the input signal, i.e., $y(t) = A_1 s(t)$ where A_1 is the complex gain quantifying the change of the phase and amplitude. Due to the physical properties of materials and variations in manufacturing, the components of a common microphone, such as the diaphragm and the pre-amplifier, are imperfect and typically do not constitute a linear system. As a result, COTS microphones, which are widely equipped on smartphones and smartwatches, typically exhibit nonlinear behavior. Specifically, the output signal $y(t)$ is under nonlinear effect, where $y(t) = A_1 s(t) + A_2 s^2(t) + A_3 s^3(t) + \cdots$, and the power gains of each component satisfy $|A_m| > |A_n| (m < n)$.

When the input signals are composed of two different ultrasonic frequencies, the output from a nonlinear microphone would contain several new shadow sounds with frequencies that are a linear combination of the two input frequencies. Assuming that the input signal is $s(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$ where f_1 and f_2 are the ultrasonic frequencies, the output signal would be $y(t) = \sum_{i=1}^{+\infty} A_i s^i(t)$. Without loss of generality, we assume $f_1 > f_2$ in the following discussion. For each component $A_i s^i(t)$,

$$
\begin{aligned}
s^i(t) &= (\cos(2\pi f_1 t) + \cos(2\pi f_2 t))^i \\
&= \mu + \sum_{j=1}^{i} [\alpha_j \cos(2\pi j f_1 t) + \beta_j \cos(2\pi j f_2 t)] \\
&+ \sum_{j=1}^{i-1} [\lambda_j \cos(2\pi (j f_1 - (i-j) f_2) t) + \gamma_j cos(2\pi (j f_1 + (i-j) f_2) t)],
\end{aligned}
$$

where α_j, β_j, λ_j and γ are coefficients of the polynomial expansion, and μ is the consequent constant.

After the pre-amplifer, the signals would pass through an embedded low-pass filter whose cut-off frequency is usually 24 kHz. Since f_1 and f_2 are both ultrasonic frequencies, $j f_1$ and $j f_2$ are all ultrasonic frequencies. However, if $i = 2j$, $j f_1 - (i-j) f_2 = j(f_1 - f_2)$ may be

a non-ultrasonic frequency when j is small enough. Therefore, when the input signal is $s(t) = \cos(2\pi f_1 t)+\cos(2\pi f_2 t)$, new audible cosine waves $\cos(2\pi j(f_1-f_2)t)$ appear, where $j = 1, 2, \ldots, k$, $k \leq i$, and $k(f_1 - f_2) \leq 24\,\text{kHz}$. Existing works like BackDoor[8] and DolphinAttack[7] make use of $A_2 s^2(t)$ but ignore higher-order components; they essentially assume that for $i > 2$, $|A_i|$ is relatively small and has little effect on the output signal. However, in our experiments, we find that more high-order components should be taken into consideration as they do affect the output signal.

3.4 Design

3.4.1 Overview

As shown in Figure 3.2, there are three parties involved in Patronus: the Scramble Transmitter, authorized devices with descramble receivers, and unauthorized devices.

The Scramble Transmitter sends a series of scramble signals with randomly varying frequencies. To ensure that unauthorized voice recordings will be affected, the frequencies of the recorded scrambles should be located in the human voice band. Therefore, we use the Scramble Generator to generate random frequencies in the target range, store them as a secret key, and send them to the Descramble Receivers through Wi-Fi, Bluetooth, or other media. The Scramble Generator then generates cosine wave segments according to these frequencies. The generated segments are then sent to the Frequency Shifter and their frequencies will be increased by f_0, which is an ultrasonic frequency. To ensure the scramble signal is picked up by microphones of unauthorized devices because of the nonlinear effect, we design a Constant Cosine Wave Generator to transmit a cosine wave with a constant ultrasonic frequency of f_0.

During human talking protected by Patronus, the actual human conversation plus two ultrasonic signals will arrive essentially simultaneously at recorders (both authorized and unauthorized) and human ears. Human ears will not detect the ultrasonic signals and thus receive the human conversation with no additional noise. As discussed in Section 3.3, the two ultrasonic signals will generate a shadow audible signal that will be included in any recording made by a COTS microphone due to nonlinear effects. This applies to both authorized and unauthorized devices.

Authorized devices, which receive a secret key from the Scrambling Transmitter, can generate the scramble waveform. They can then feed the scramble waveform along with the scrambled recording into an adaptive filter to extract clear speech from the scrambled speech. The details of descrambling will be discussed in Section 3.4.5.

We must overcome three challenges in order to design Patronus. First, we must design a system whose working area is as large as possible. This is difficult because a sound wave of high frequency typically travels along a straight line meaning a straightforward implementation of ultrasonic generators will only cover a small area defined by a limited range of angles. Second, there is a trade-off between a shorter and a longer period of scramble frequencies. As the period increases, the system is more vulnerable to unauthorized recordings using STFT attacks. As the period decreases, the difficulty of descrambling increases. Our goal is to maximize the information recovered by authorized devices over unauthorized ones without exposing the scramble pattern to STFT. These details are discussed in Section 3.4.3.4. Third, when frequency changes frequently, a severe ringing effect (Section 3.4.3) occurs in the scrambled recording, which affects even the recordings made by authorized devices after descrambling. We use chirps to connect each frequency component of the scramble to eliminate the sudden change of the input to ultrasonic speakers, hence minimizing the ringing effect and enhancing the quality of the recovered speech by authorized devices.

3.4.2 Attack Model

Based on common acoustic processing technologies and known properties of nonlinearity effects, we consider the following types of attacks:

3.4.2.1 Short-Time Fourier Transform (STFT)

One natural way for an unauthorized device to try to extract a useful recording from its scrambled recording is to analyze the scrambled recording with STFT and filter out suspicious frequencies. We address this attack model by changing the scramble frequency according to a finely-tuned period model, making it impossible for the attacker to obtain each exact scramble frequency along with its

start and end time. Detailed analysis is provided in Section 3.4.3.4. Even with the correct scramble frequencies available, bandpass filters will not work because the scramble frequencies are selected from the human voice band. The frequencies from chirps and those from human speaking are mixed together. To prove Patronus can defeat this attack model, we simulate the attack scenario when (1) the attacker is aware that our scramble pattern is varying continuous waves smoothed by chirps (2) the attacker calculates approximate scramble frequencies with STFT (3) the attacker applies NLMS adaptive filter (Section 3.4.5.4) to remove the scramble with the approximate scramble frequencies they obtained from STFT. Our simulated attack experiments, provided in Section 3.6.8, show that this attack will fail because the approximate scramble frequencies are not accurate enough.

3.4.2.2 Extra Ultrasonic Transmitter Attack

After DolphinAttack[7] proposes to inject malicious commands into ultrasound, AIC [9] adds three more ultrasonic transmitters to cancel the malicious commands and protect Voice Control Systems (VCS). AIC assumes the legitimate as well as malicious commands are within the lower sub-band of the microphone sensible frequency band. Their added ultrasonic transmitters project only the malicious commands onto the higher sub-band, which can be used to filter the malicious commands in the low sub-band. With a fast changing of scramble frequencies, we can cover the whole frequency band, and make sure no clean band is left for attackers.

3.4.2.3 Wi-Fi/Bluetooth Snifing

Attackers can sniffer the Wi-Fi or Bluetooth channel to get the scramble pattern transmitted from the Scramble Transmitter to the authorized device. However, there are many cryptographic approaches to prevent attackers from sniffing channels. For example, we can encrypt the scramble pattern by AES-CTR using a pre-shared key and then directly send it to authorized devices.

3.4.2.4 Physical Attacking

There are also some physical attack models. First, attackers can place an obstacle before the Scramble Transmitter. However, attackers cannot do it secretly and nobody would like to do so. Second, attackers may just wrap a cover on their microphones. However, the cover itself may defeat the attackers objective of making a good recording. Although Patronus cannot perfectly handle such attack models, it enhances the difficulty of making an unauthorized recording. Finally, attackers may conduct experiments to discover where Patronus fails. This can be fixed by enlarging the working area through some methods that we will discuss later.

3.4.3 Ultrasonic Scramble Modulation

Two ultrasonic signals will be superimposed at the recorders to create the desired low-frequency component. In the design of the scramble using ultrasonic signals, we mainly consider the following issues:

3.4.3.1 Range of Frequency

The first issue is how to make it hard to cancel out the scramble without the key. Basically, the range of human speech frequency is from 85 Hz to 255 Hz [70, 71]. If the scramble consists of multiple random frequencies from this range, it is hard for attackers to cancel the scramble using linear filters. The application of a linear filter, e.g., highpass filter, will not only cancel the scramble, it will also change the original human speech. To ensure the scramble covers all human speech frequencies in practice, we modulate the scramble with a wider frequency band than $[85, 255]$ Hz.

3.4.3.2 Random Frequencies

If we always use specific frequencies to generate the scramble, attackers could analyze the frequency spectrum of their recordings to infer the scramble frequencies; with those, they could then recover the original audio signals. To address this issue, we choose scramble frequencies randomly. We

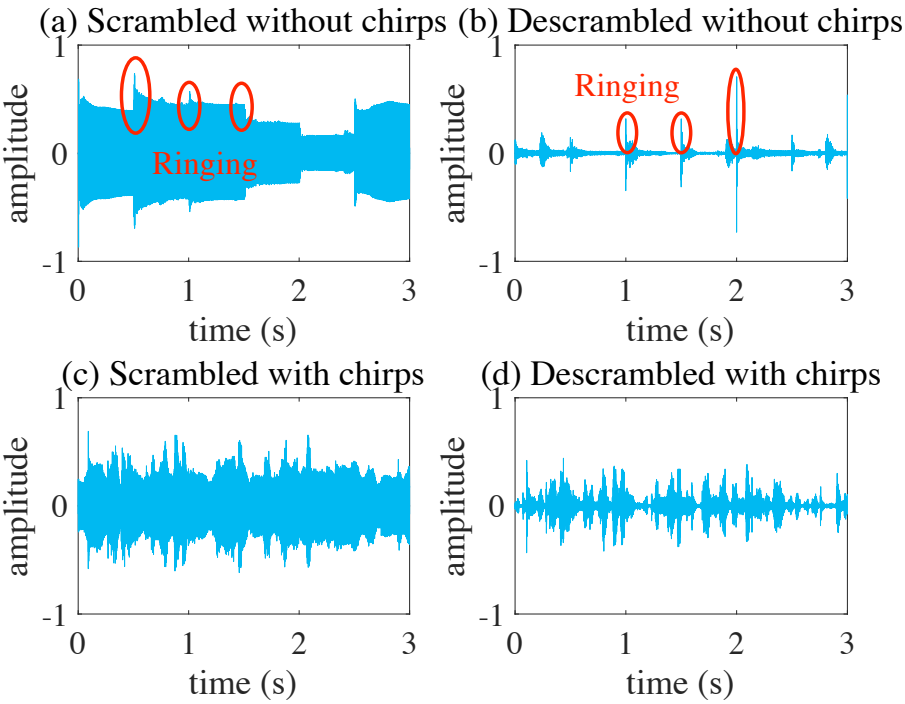


Figure 3.3: Illustration of how linear chirps mitigate the ringing effect.

also periodically change the scramble frequencies over time. The sequence of scramble frequencies can be thought of as a one-time pad key. Without the sequence, it would be difficult for attackers to remove the scramble.

3.4.3.3 Ringing Effect

Frequent changing of the scramble frequencies produces a ringing effect [8] that makes it challenging for authorized devices to produce a high-quality descrambled recording. Specifically, the ringing effects incur heavy-tailed impulse responses that will remain in descrambled recordings as shown in Figure 3.3 (a) and (b). Since the ringing effect occurs when the input changes suddenly, we use a chirp signal to connect two adjacent segments with different frequencies in the scramble to smooth such a sudden change. Specifically, when the scramble changes from frequency A to frequency B, we add a transition signal that starts at frequency A and moves linearly to end with frequency B.

The impulse incurred by ringing effects can have a very high amplitude or power. It will suppress other signals due to the microphone Passive Gain Suppression [8]. Figure 3.3 confirms

that the ringing effect is mitigated by chirps. Figure 3.3 (a) shows a scrambled recording with no chirp, the resulting descrambled recording in Figure 3.3 (b) has many areas where most of the signal is suppressed. In contrast, Figure 3.3 (c) exhibits a scrambled recording with chirp signals, the resulting descrambled recording in Figure 3.3 (d) does not have the peak signals corresponding to the ringing effect and the rest of the signal is not suppressed.

3.4.3.4 Duration of each frequency

The next challenge is choosing the proper duration for each frequency in the sequence of scramble frequencies. Intuitively, if we give each frequency a long duration, unauthorized devices could easily split the record into multiple segments where each segment is only protected by a constant frequency scramble. They could then apply simple techniques such as using a linear bandpass filter to the scrambled recording to extract a clear speech recording.

More generally, there are two competing issues in choosing the duration of each scramble frequency, namely, defending against STFT attacks that are discussed in Section 3.4.2.1, and ensuring that authorized devices can obtain high-quality descrambled recordings. We first consider defending against STFT attacks. An STFT attack can successfully remove the scramble waveform if it can both accurately infer the frequencies and time periods for each scramble frequency in the sequence of frequencies. When the window length is n, the frequency resolution would be $\Delta f = \frac{f_s}{n} = \frac{f_s}{f_s \times t} = \frac{1}{t}$ where f_s is the sampling rate and t is the duration of the window. Taking 0.1s as an example, the offset of STFT can reach 10Hz. If the attacker tries to improve the frequency resolution by lengthening the window, the accuracy of the estimated time periods for the given scramble frequency will diminish. If the scramble frequency duration is long, scramble frequency will exhibit fewer changes within any given window, thus STFT attacks can use longer windows to accurately estimate the frequency with exact estimates of the frequency time period. Therefore, to thwart STFT attacks, we should make the frequency duration as short as possible. However, a too-short duration may misshape the scrambled recording due to imperfect hardware. A typical microphone and speaker use a diaphragm to sense and generate the vibration; this diaphragm moves

continuously and can not change its position instantaneously. Circuit latency also makes it hard for the system to respond to frequent and instant changes. As a result, the scrambled waveform would be slightly distorted. This means the NLMS adaptive filter at authorized devices may not correctly descramble the scrambled waveform because it does not expect the distortion caused by frequent frequency changes. Therefore, the frequency duration cannot be too short. In summary, to balance these competing concerns, we must find a frequency duration that maximizes the information recovered by authorized devices compared to the information recovered by unauthorized devices. To identify a good frequency duration, we measure the descrambling performance with different frequency durations in Section 3.6.8.

3.4.3.5 Key Construction

We have two choices to construct the key for granting the privilege of recording the audio to authorized devices. One is directly using the scramble waveform generated by the Scramble Generator as the key. After getting the scramble waveform, authorized devices remove the scramble from the recorded audio. But there are some issues we need to consider. First, the sampling rate of authorized devices may vary from one to another. It means that in terms of the digital signal, devices having different sampling rates will get different presentations of the same scramble waveform. To grant the privilege to devices, the Scramble Transmitter should generate different digital scramble waveforms according to different sampling rates of authorized devices. This results in high computational overheads. Second, in addition to different sampling rates from different authorized devices, the sampling rates of the Scramble Generator and an authorized device may be also different. As a result, the scramble that the speaker emitted might have a different presentation of the recorded waveform.

In Patronus, we choose another way to construct the key. We select the frequency sequence used to generate the scramble as the key. After receiving the frequency sequence, an authorized device can reconstruct the scramble waveform with their sampling rates, which we discuss in more detail later. After that, an authorized device can use the reconstructed scramble waveform to remove the

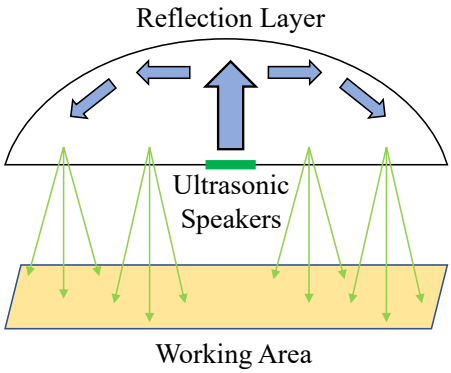


Figure 3.4: Enlarge working area with reflection.

scramble from the recording and get the clear speech.

With the discussion above, we formally describe the scramble generation. We set one speaker to transmit an ultrasonic continuous wave $S_1(t) = \cos(2\pi f_0 t)$, while the other speaker transmits continuous waves linked by chirps $S_2(t) = \cos(2\pi f(t)t)$, where

$$f(t) = \begin{cases} f_i, & (2i-2)\Delta t \le t < (2i-1)\Delta t, \\ f_i + \frac{f_{i+1}-f_i}{\Delta t}t, & (2i-1)\Delta t \le t < 2i\Delta t, \end{cases} \tag{3.1}$$

and $f_i (i = 1, \ldots, n)$ are randomly generated constant frequencies. Δt is the duration of a single sine wave or a chirp. The induced low-frequency noise will be

$$R(t) = \cos(2\pi(f(t) - f_0)t). \tag{3.2}$$

To ensure $R(t)$ covers human voice, $f_i (i = 1, \ldots, n)$ are sampled from $[f_{low} + f_0, f_{high} + f_0]$ where $[f_{low}, f_{high}]$ covers the human voice band.

3.4.4 Enlarge Scramble Working Area

The scramble signal is generated by two ultrasonic signals, which incurs another issue as the ultrasonic wave typically propagates in a straight line. In other words, if you want to prevent a certain device from recording, the ultrasonic speaker should be pointed directly towards that device. This results in a limited coverage area for ultrasonic anti-recording solutions.

Inspired by lamps that often use a bow-shaped cover to reflect the light beam in many directions, we build a reflection layer that reflects the ultrasonic wave in many directions. As Figure 3.4 shows, we put ultrasonic speakers near the center of the reflection layer and place the devices (authorized and unauthorized) in the working area. When the ultrasonic wave hits the reflection layer, it gets reflected in many directions leading to a much larger cover area.

3.4.5 Grant Recording Privilege

The goal of Patronus is not only to block unauthorized devices from recording audio, but also to provide authorized devices with a mechanism to recover speech. Patronus achieves this by creating a way for authorized devices to remove the scramble from the scrambled recording. Specifically, Patronus grants the clear recording privilege to authorized devices using the following steps.

3.4.5.1 Key Transmission

The Descramble Receiver needs the waveform of the scramble generated by the Scramble Generator before it can remove the scramble. Intuitively, if it had the pure scramble waveform, it could remove the scramble from the recorded audio by subtracting the scramble waveform from the recorded audio waveform. The scramble waveform here acts as the key for deciphering the recorded audio. We send the key through non-acoustic channels such as Wi-Fi or Bluetooth with cryptographic protection to prevent eavesdroppers from getting the key. Additionally, because of the randomness of scramble frequencies, they cannot get a usable scramble waveform by listening to the acoustic channel. Instead, they can get either the combination of interfered speech with scramble, or get the scramble without speech but independent of the successive scramble waveform.

3.4.5.2 Scramble Reconstruction

As discussed in Section 3.4.3, the Scramble Transmitter sends the random frequency sequence instead of the scramble waveform to authorized devices as the key. Patronus needs to use these

frequencies to reconstruct the scramble waveform before removing the scramble. An authorized device uses Equation (3.2) and its recording sampling rate to generate the scramble waveform.

3.4.5.3 Synchronization

We need to synchronize the reconstructed scramble with the recorded scramble before removing it from recordings. Specifically, we choose a segment from the reconstructed scramble as the template, e.g., the beginning segment. Then we use cross-correlation to find the segment that is the most similar to the template. We then synchronize the recorded scramble and the reconstructed scramble by aligning the two segments.

3.4.5.4 Adaptive Filtering

Now we have the waveform of the scramble. The next task is to remove the scramble from the recorded audio with the known waveform of the scramble. Practically, we cannot directly subtract the scramble from the recorded audio because when the sound propagates through the air, it will be distorted due to reflection and attenuation. We use adaptive filter to remove the waveform-known scramble.

Adaptive filter is widely used in Active Noise Cancellation (ANC) headsets. Technically, there is a reference microphone outside the headset. The reference microphone captures the noise, and the digital signal processor (DSP) generates the anti-noise wave according to the captured noise. When the noise wave and the anti-noise wave arrive at the ear, they eliminate each other. In Patronus, we denote the speech as x_1. It propagates through the acoustic channel h_1, arrives at the authorized device and becomes $h_1 * x_1$, where the operator $*$ denotes the convolution operation. Additionally, we denote the scramble waveform that is generated by non-linear effects and recorded by the authorized device as x_2. It propagates through another channel h_2, arrives at the authorized device and becomes $h_2 * x_2$. Therefore, the audio recorded by the authorized device is

$$y = h_1 * x_1 + h_2 * x_2. \tag{3.3}$$

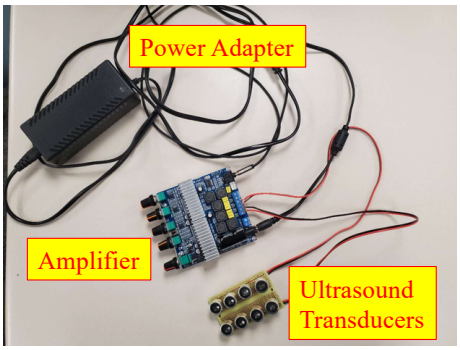


Figure 3.5: Implementation of Scramble Transmitter.

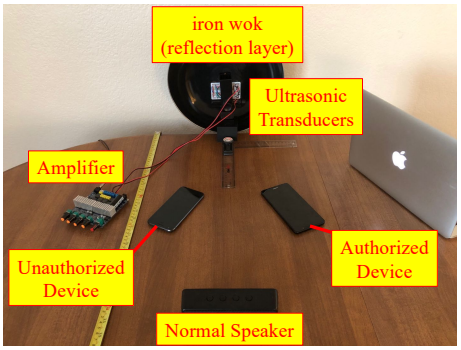


Figure 3.6: Prototype of Patronus.

Similar to ANC headsets, here we see the scramble x_2 as the noise in ANC headsets. Different from ANC headsets, the noise here is generated from the key as we discussed in Section 3.4.5.2. Therefore, we can use the Normalized Least-Mean-Square (NLMS) Adaptive Filter [61] to remove the scramble. Formally, we are trying to find a channel vector h_2' to solve the optimization problem

$$\min E[(y - h_2' * x_2)^2]. \tag{3.4}$$

When the expectation in Equation (3.4) is minimized, $h_2 \approx h_2'$. Therefore, $h_1 * x_1 \approx y - h_2' * x_2$, and it can be regarded as the speech without the scramble. Stochastic gradient descent is usually adopted to solve the optimization problem defined by Equation (3.4), but it is hard to derive the gradient of the expectation. Researchers thus use $(y - h_2' * x_2)^2$ instead of the expectation to solve the problem. In this way, the noise gets canceled [72].

Following this design, we can develop a mechanism that prevents unauthorized recording while supporting authorized recording. The mechanism also prevents attackers from descrambling without authorization. Figure 3.7 gives an example. A piece of VOA news audio is used as the original record, the attack result has severe scramble effects just like the unauthorized record, but the authorized record removes almost all scrambles.

3.5 Implementation

This section discusses the details of the implementation of Patronus, which contains two parts, the Scramble Transmitter and the Descramble Receiver for authorized devices. We use an ordinary

smartphone with its built-in audio recorder as the Unauthorized Device or Authorized Device.

3.5.1 Scramble Transmitter

3.5.1.1 Hardware Implementation

As Figure 3.5 shows, we use eight TCT40-16R/T 16 mm ultrasonic transducers. Half of them play the frequency-shifted scramble and they are connected in parallel. The other half play the fixed-frequency cosine wave and are connected in parallel as well. We utilize an AOSHIKE DC12V-24V 2.1 Channel TPA3116 Subwoofer Amplifier Board to enhance the power of output ultrasonic signals. The two waveforms are played through a stereo channel. The frequency-shifted scramble uses the left channel, and the constant-frequency cosine wave uses the right channel.

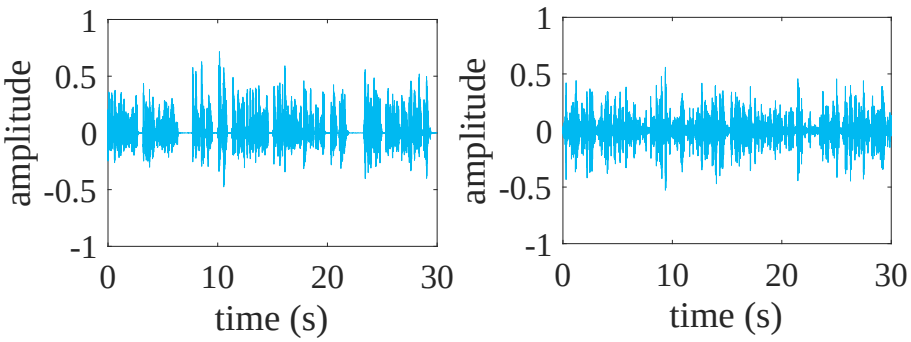
As we have discussed in Section 3.4.4, we use a reflection layer to enlarge the working area. In this prototype, we use an iron wok as the reflection layer. The opening diameter of the iron wok is 30 cm, and the depth is 10 cm. As shown in Figure 3.6, the ultrasonic transducers are placed towards the center of the iron wok.

3.5.1.2 Format of Key

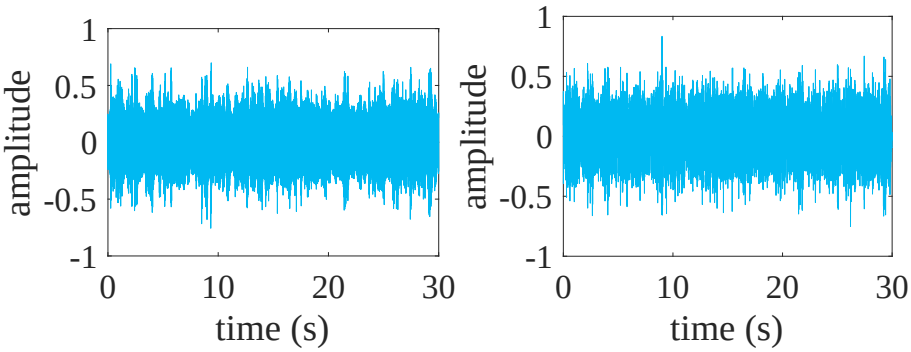
As we have mentioned in Section 3.4, Patronus uses the frequency sequence as the key. This key must include the duration of each frequency in addition to the frequency itself in order for the Descramble Receiver to generate the scramble waveform. Thus, our key file includes the frequency sequence plus the sample rate of the Scramble Transmitter and the number of samples of each frequency.

3.5.2 Descramble Receiver for Authorized Devices

We use an ordinary smartphone as an authorized device. The authorized device receives the key from the Scramble Transmitter. After the audio is recorded, the smartphone reconstructs the



(a) Original waveform (b) Authorized waveform



(c) Unauthorized waveform (d) Descrambled by STFT attack

Figure 3.7: Illustration of original waveform, authorized waveform, unauthorized waveform, and descrambled waveform by STFT attack.

scramble waveform with the given key and leverages NLMS Adaptive filter to cancel the scramble. Formally, it takes the following steps:

3.5.2.1 Reconstruct Scramble Waveform

As we mentioned, in addition to the frequency sequence, the received key also contains the sampling rate of the Scramble Transmitter, which is denoted by f_{st}, as well as the number of samples of each frequency n_t. With the known sampling rate of the authorized device f_{sr}, the number of its recovered samples for each scramble frequency component can be calculated through the equation

$$n_r = \frac{f_{sr} n_t}{f_{st}}, \tag{3.5}$$

After getting n_r, the authorized device uses the same process as the Scramble Transmitter to generate the scramble, i.e., generating the discrete cosine signal with the frequency f_i and f_{i+1}, and

connecting them by a chirp signal with start frequency f_i and end frequency f_{i+1}, where f_i and f_{i+1} are from the frequency sequence in the key.

3.5.2.2 Normalized Least-Mean-Square (NLMS) Adaptive Filter

After reconstructing the scramble waveform, we can use the Normalized Least-Mean-Square Adaptive Filter to cancel the scramble from the scrambled record. Specifically, we put the scrambled record rec_s and the scramble waveform s into the NLMS Adaptive Filter to get the descrambled waveform e by removing s from rec_s. According to the discussion in Section 3.3, the scramble wave is not only generated by frequencies in the given frequency sequence but also generated by high-order frequencies that are multiples of the target frequencies. Therefore, after getting e from the NLMS Adaptive filter, we still need to iteratively remove the multiples of the frequency sequence scramble by NLMS Adaptive filter. It means that we iteratively put e and the scramble waveform generated by k-times multiple of the frequency sequence into NLMS Adaptive Filter, where $k = 2, 3, 4, 5, 6$ in our prototype.

In summary, the procedure of authorized devices for removing the scramble from the record is shown in Algorithm 3.1.

Input: rec_s, f_{sr}, f_{st}, n_t,
the frequency sequence $f[1..n]$
Output: Speech Record without Scramble e

1: $n_r \leftarrow f_{sr} n_t / f_{st}$
2: $e \leftarrow rec_s$
3: **for** $k = 1$ to 6 **do**
4: $s \leftarrow$ ScrambleGenerator($k \times f[1..n]$, n_r).
5: $e \leftarrow$ NLMS-Adaptive-Filter(e, s)
6: **end for**
7: **return** e

Algorithm 3.1: Remove Scramble from the record.

The NLMS-Adaptive-Filter can be found in many open-source libraries, e.g., MATLAB, Python, etc. Due to the selective frequency response of different smart devices, each model has its own parameter setting. In the implementation, we choose 500 as the number of taps and 0.005 as the

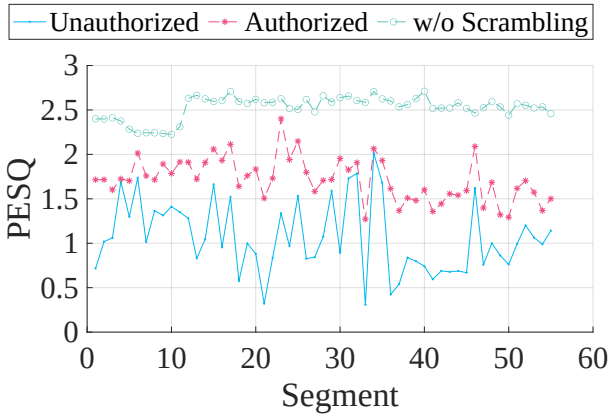
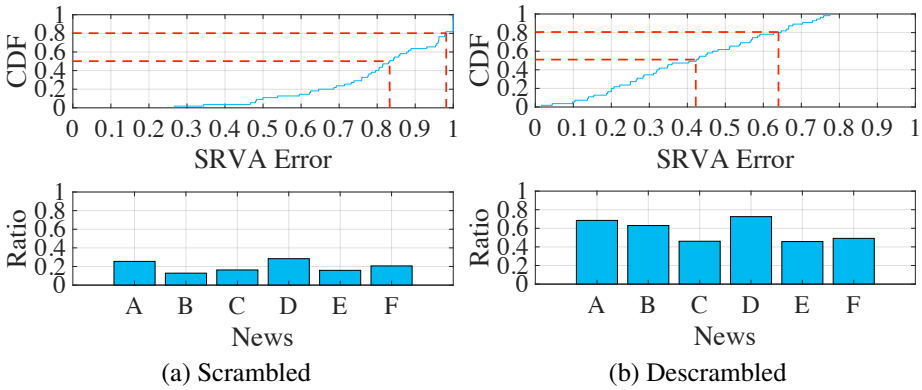


Figure 3.8: PESQ of recordings captured by unauthorized and authorized devices, and PESQ of recordings without scrambling by turning off Patronus as the baseline.

step size for an iPhone, 100 as the number of taps and 0.003 as the step size for a Pixel, and 300 as the number of taps and 0.005 as the step size for a Galaxy S9.

3.5.3 Simulated STFT Attacker

We also simulate an STFT attacker to verify whether or not Patronus can prevent such an attack. Specifically, as discussed in Section 3.4.2.1, we apply STFT to the scrambled recording using the MATLAB function *stft* to infer its frequency sequence. We then feed the frequency sequence to an NLMS adaptive filter to get the descrambled recording. Experiment results are shown in Section 3.6.8. Here, we illustrate an example, which contains the original waveform, authorized waveform, unauthorized waveform and the waveform descrambled by STFT, in Figure 3.7. As illustrated by the figure, we observe that the authorized waveform is similar to the original waveform, the unauthorized waveform is different from the original one, and the unauthorized waveform is similar to the waveform descrambled by STFT attack. Therefore, our prototype proves that Patronus can block the unauthorized recording while allowing authorized recording, and it can prevent STFT attacks.



(a) Scrambled (b) Descrambled

Figure 3.9: (a) Upper half: The CDF of SRVA Error of scrambled recordings from the unauthorized device. Lower half: The ratio of SRVA between scrambled recordings and original waveforms. (b) Upper half: The CDF of SRVA Error of descrambled recordings from the authorized device. Lower half: The ratio of SRVA between descrambled recordings and original waveforms.

3.6 Evaluation

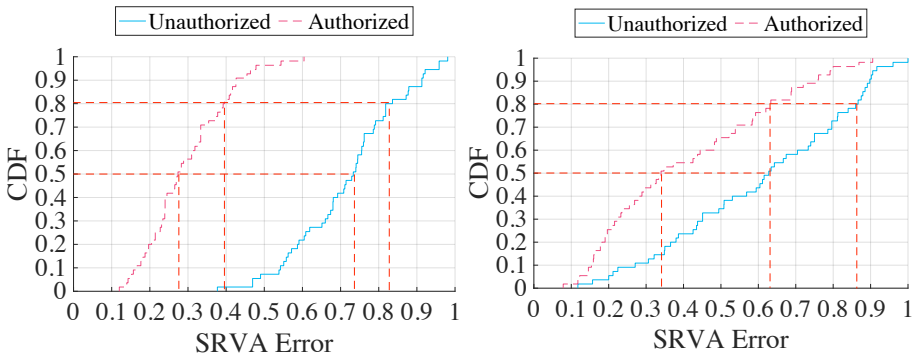
3.6.1 Overview

To evaluate the performance of Patronus, we select six news speech waveforms from Voice of America (VOA) and note these waveforms as A - F. The news speeches are read by a male, a female, or both alternatively, sometimes with background music.

A normal speaker (shown in Figure 3.6) is set to play these news waveforms, and we also read the news ourselves. While the news waveforms are played under different conditions, we start Patronus to interfere with the unauthorized recording device. Meanwhile, an authorized device is recording too. Later we apply scramble cancellation to recordings from the authorized device. After getting the scrambled recordings and scramble-canceled recordings, the following metrics are adopted to measure the performance of Patronus.

3.6.1.1 Perceptual Evaluation of Speech Quality (PESQ)

PESQ is a common-used metric of speech quality [62]. It is widely adopted by phone manufacturers, network equipment vendors, and telecom operators. Technically, the inputs include a clear speech signal as the reference and a signal that needs to be measured. The output is a Mean Opinion



(a) Human reading (b) Human hearing

Figure 3.10: (a) Compare SRVA between before and after descrambling for the human voice. (b) Compare SRVA between before and after descrambling for human recognition.

Score (MOS) [73] ranging from −0.5 to 4.5. A high PESQ score means that the corresponding speech has a high hearing quality and vice versa. Typically, PESQ values ranging from 1.00 to 1.99 means "No meaning understood with any feasible effort" while those ranging from 3.80 to 4.50 meaning "Complete relaxation possible; no effort required" [74]. However, we cannot regard the audio recording as strict as lossless communication. To fit PESQ to characterize the performance of Patronus, we measure the PESQ of recordings without scrambling by turning off Patronus, and use that result as the baseline. As shown in Figure 3.8, such recordings have PESQ between 2.2 and 2.7. We regard them as the upper bound of both unauthorized and authorized recordings. In the following experiments, we use the PESQ implementation written in MATLAB [75] to compute the PESQ score.

3.6.1.2 Speech Recognition Vocabulary Accuracy (SRVA)

We also use a Speech Recognition service to measure the effectiveness of scrambling and descrambling. Specifically, we apply Google's Speech To Text (STT) service to transform the acoustic signals to text. We first use the STT service to recognize the original speech without interference and treat the recognized word sequence w_c as the ground truth. Then we use the STT service to recognize the scrambled speech and descrambled speech, and use w_s and w_d to denote their results, respectively. We name $\frac{\sum_{i \in w_s} isTrue(i \in w_c)}{|w_c|}$ (or $\frac{\sum_{i \in w_d} isTrue(i \in w_c)}{|w_c|}$) as the Speech Vocabulary Recognition

Accuracy (SRVA) and use it to quantify the effectiveness of scrambling and descrambling. Note that $isTrue(i \in w_c)$ returns 1 when i is a word from w_c, and 0 when i is not a word from w_c. We define SRVA Error as 1 – SRVA which indicates the error rates of recognition with the STT service.

Using the above metrics, we try to answer the following questions:

- Can Patronus effectively scramble the unauthorized speech recordings?
- Can Patronus permit authorized devices to record the speech?
- Can Patronus work on different mobile devices?
- What is the impact of the distance between Patronus and a recorder?
- What is the impact of the reflection layer?
- What is the impact of the frequency switching time?
- Is it possible to perform real-time descrambling?

3.6.2 Effectiveness of Scrambling and Descrambling

We split the 6 news speech waveforms into 55 segments (1650 seconds in total), each 30 seconds long. Both the authorized and unauthorized device are Apple iPhone X in this experiment, so do the following experiments except that of Section 3.6.5. As shown in Figure 3.8, with Patronus's scrambling, the hearing qualities of most segments are extremely low. Specifically, 44 out of 55 (80.0%) segments have PESQ scores lower than 1.5. For SRVA, overall, only 551 out of 2796 (19.7%) words are recognized correctly. More detailed results are shown in Figure 3.9a. The upper half shows the CDF of the SRVA Error. We can know that 50% of the recordings have SRVA Error lower than 0.84, and 80% of the recordings have SRVA Error lower than 0.98. The lower half shows the ratio of SRVA between scrambled recordings and original waveforms. The results show that all of the news waveforms having a recognition rate lower than 0.3. Here we want to mention that if a word appears multiple times in a speech, SRVA would result in a high value or

a low value compared to the actual word recognition rate. However, duplicated words have little impact because the duplicate rates of every segment, i.e., the ratio between the count of a specific word and the total count of words in the segment, are lower than 5%.

To evaluate the effectiveness of descrambling, an authorized device records the speech under the scrambling from Patronus. The authorized device then cancels the scramble using the received key. As shown in Figure 3.8, after descrambling, only 9 out of 55 (16.3%) segments having PESQ scores lower than 1.5. On average, descrambled recordings have 1.6x higher PESQ scores than their corresponding scrambled recordings. As for SRVA, we show the CDF of the SRVA Error in the upper half of Figure 3.9b. These results show that 50% of the descrambled recordings have SRVA Error lower than 0.43, which is 49% lower than scrambled recordings. Moreover, 80% of the descrambled recordings have SRVA Error lower than 0.64, which is 35% lower than scrambled recordings. As shown in the lower half of Figure 3.9b, ratios of SRVA between descrambled recordings and original waveforms are higher than 0.4 and lower than 0.8. They are at least 2x better than the scrambled recordings. The quality of the descrambled recordings is not as good as the original ones because there are residual components of the scramble after applying the NLMS adaptive filter. Moreover, background music and the volume of the original waveform also affects the quality of the descrambled recordings. For example, news C has a lower ratio after being descrambled by the authorized device compared to the other news clips because it has background music that could affect the performance of authorized devices. It also affects the SRVA of the record without scrambling, i.e., only 223 words are recognized from 295 in total. The reader of news E reads the news in a lower volume compared to others, so it has a lower ratio after being descrambled by the authorized device compared to the other news clips.

3.6.3 Effectiveness of Human Voice Scrambling and Descrambling

To verify whether Patronus works for real human speaking other than a sound player, we read the news and calculate SRVA. As shown in Figure 3.10a, Patronus can effectively scramble and descramble the human voice. Specifically, for the scrambled recordings, the median of SRVA Error

is 0.74, and 80% of scrambled recordings have SRVA Error lower than 0.83. For the descrambled recordings, the median of SRVA Error is 0.27, and 80% of the descrambled recordings have SRVA Error lower than 0.4. The descrambling effectiveness of the human speaker is better than that of recorded sounds because recorded sounds from VOA sometimes play background music.

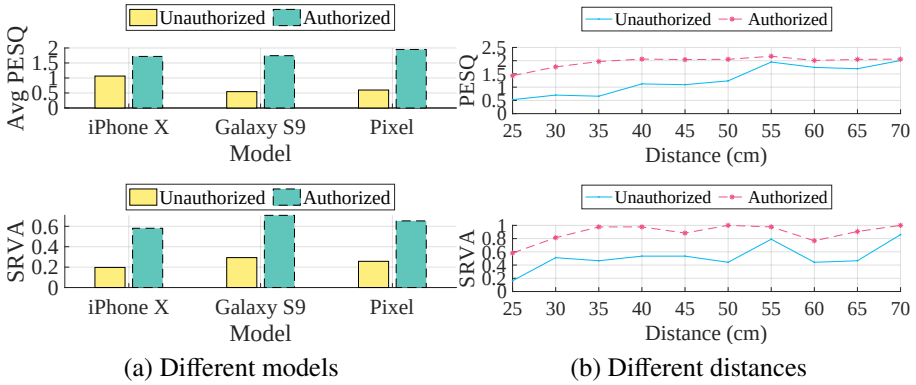
3.6.4 Effectiveness of Human Recognition to Scrambled Recordings and Descrambled Recordings

Because there might exist differences between machine learning-based speech recognition and human speech recognition, we invite 11 volunteers to write down words after listening to the 55 scrambled recordings and 55 descrambled ones. The results are shown in Figure 3.10b. People react differently to noise. Some people are very sensitive and the scrambled noise make them very uncomfortable. Note, the noise is generated by ultrasound speakers and only captured by the nonlinear effects of microphones, so it will not disturb the people in the original conversation. It will only be heard after getting recorded by unauthorized devices. Further, authorized devices will be able to filter out such noises eliminating the discomfort for those listeners. The recovered information from humans listening to descrambled recordings is still better than that of humans listening to scrambled ones. 50% of the scrambled recordings have SRVA Error lower than 0.63, and 80% of the scrambled recordings have SRVA Error lower than 0.86. As a comparison, 50% of the descrambled recordings have SRVA Error lower than 0.34, and 80% of the descrambled recordings have SRVA Error lower than 0.63.

3.6.5 Effectiveness on Different Mobile Models

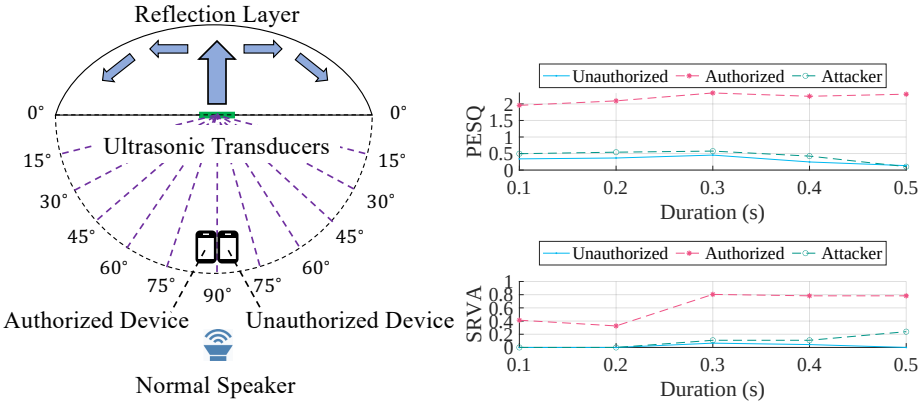
To verify whether Patronus works on different mobile models, we test it on three devices, an Apple iPhone X, a Samsung Galaxy S9, and a Google Pixel. We play all 55 segments using the normal speaker, and calculate average PESQs and SRVAs.

As shown in Figure 3.11a, less than 30% of words can be recognized by the STT service for all the unauthorized devices, and around 65% of words can be recognized for all the authorized



(a) Different models (b) Different distances

Figure 3.11: (a) Compare average PESQ and SRVA among different models, (b) compare PESQ and SRVA at different distances.



(a) Angle illustration (b) Different durations

Figure 3.12: (a) Illustration of the reflection layer experiment. (b) Compare PESQ and SRVA with different frequency switching times.

devices. When the mobile devices are unauthorized, the average PESQ of iPhone X is 1.06, and the average PESQ of the other two models are even lower, roughly 0.5. When the mobile devices are authorized, they all achieve an average PESQ around 1.85. This demonstrates that Patronus works well for all devices; namely, it prevents all models from making good unauthorized recordings and allows all models to make acceptable authorized recordings.

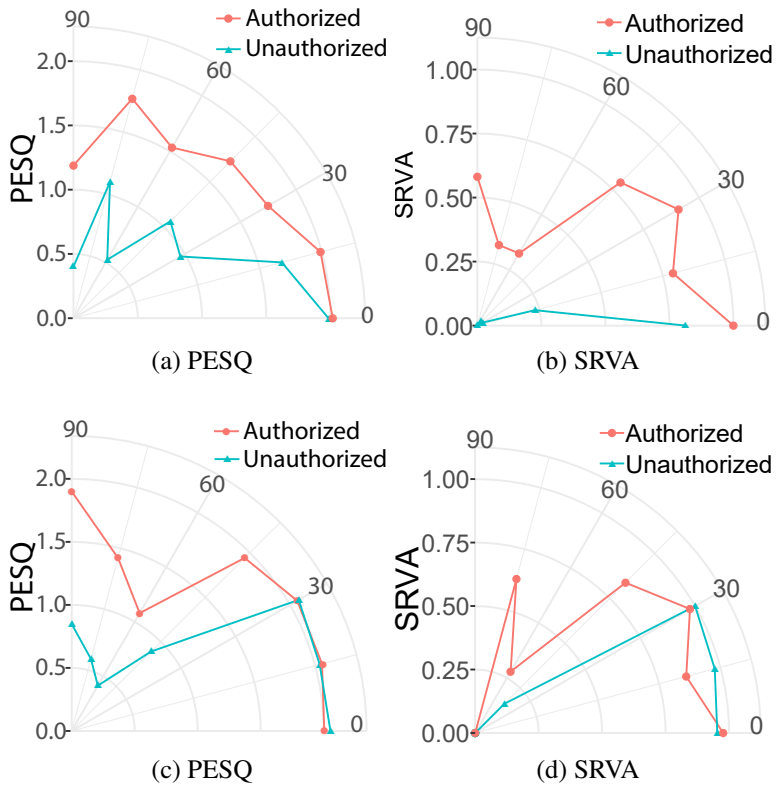
3.6.6 Impact of the Distance

We also characterize the impact of the distance between Patronus and the recording devices (both authorized and unauthorized). We put the Scramble Transmitter at the origin. A randomly-picked

speech segment (which has 43 words) is played by a normal speaker, which simulates the talker. The authorized device and an unauthorized device are recording at the same time. Their distance to the Scramble Transmitter varies from 25 cm to 70 cm. Results of SRVA and PESQ between two devices are shown in Figure 3.11b. Overall, as the distance increases, the ultrasound would attenuate more. Therefore, the strength of the scramble decreases as the distance from the scramble transmitter increases. As a result, when the device is far enough away, both the authorized and unauthorized device can both record a clear speech. On the other hand, when devices are close enough, unauthorized devices produce recordings that are severely scrambled whereas authorized devices can recover much clearer speech using the secret key. The working area can be extended by using high power ultrasonic speakers, which we will discuss later. Here we want to mention that although there is a bump in Figure 3.11b at 55 cm with the SRVA, PESQs of 55cm and 60cm are close. This means that humans cannot see much difference between these two recordings, something we confirmed in person by listening to these recordings with this objective in mind. Thus, the SRVA bump at 55cm might be due to an error-correction mechanism of the Google STT engine; of course, since this is proprietary technology, we do not know how or why this error-correction would produce such a performance bump for this recording.

3.6.7 Impact of the Reflection Layer

As we mentioned before, the ultrasound wave often propagates along a straight line. To enlarge the range of Patronus scrambling, we design a reflection layer. In this experiment, we apply the common speaker to play the chosen speech segment (43 words). As shown in Figure 3.12a, we point the ultrasonic speakers towards the reflection layer and change angles of both authorized and unauthorized devices to the ultrasonic speakers and measure Patronus' performance; in other experiments, the devices are always put at the 90° angle. We also measure the performance without using the reflection layer. We turn the ultrasonic speakers around so they face in the same direction as the normal speaker when we remove the reflection layer. The results when using the reflection layer are shown in Figure 3.13a and 3.13b, and the results without using the reflection layer are



(a) PESQ (b) SRVA

(c) PESQ (d) SRVA

Figure 3.13: (a) and (b): Compare PESQ and SRVA with the using of the reflection layer. (c) and (d): Compare PESQ and SRVA without the using of the reflection layer.

shown in Figure 3.13c and 3.13d. From the results, we see that with the reflection layer, Patronus can successfully scramble the unauthorized device when the angle is more than 15°, which is signficantly larger than the angle of more than 45° needed by Patronus without the reflection layer. Therefore, the reflection layer does significantly enlarge the scramble range of Patronus.

3.6.8 Impact of the Frequency Duration

We also measure the impact of the frequency duration. As we discussed in Section 3.4, we would like to make the duration of each frequency as short as possible. However, the shorter the frequency duration is, the harder it is for authorized devices to descramble. To verify this feature, we put an authorized and an unauthorized device at 40 cm to Patronus and play the chosen segment (43 words) using the normal speaker. Both devices record the speech under Patronus using 5 different frequency durations: 0.1 s, 0.2 s, 0.3 s, 0.4 s and 0.5 s. We calculate PESQs and SRVAs for each

DT (ms) \ MSO / RT (s)	1	2	3	4	5	6
1	51	96	159	209	265	328
2	73	145	218	291	373	454
5	161	322	487	634	798	954
10	290	582	851	1108	1389	1653
20	548	1094	1653	2165	2695	3298
30	822	1617	2348	3088	3830	4563

Table 3.1: Descramble time (DT) of different record times (RT) with different max scramble orders (MSO, the upper bound of k in Algorithm 1).

duration. Moreover, we implement the attack model from Section 3.4.2, which first calculates approximate scramble frequencies using STFT and then attempts to cancel the scramble using an NLMS adaptive filter. We calculate PESQs and SRVAs for each duration and all devices including the attack model.

As shown in Figure 3.12b, for all durations, SRVAs of the unauthorized device are lower than 0.1, and PESQs are lower than 0.5. The authorized device has higher SRVAs and PESQs than the unauthorized device. Specifically, when the duration comes to 0.3 s, the SRVA reaches roughly 0.8 and PESQ exceeds 2.0. This verifies our claim that authorized devices can successfully descramble when the frequency duration is long enough.

A shorter duration also makes it harder for attackers to crack the scrambled record, e.g., SRVAs for the attacker also increase as the duration increases. Although both SRVAs and PESQs are higher than those of the unauthorized device, they are still too low to extract useful information. The reason why the NLMS adaptive filter fails is that the attacker cannot identify the scramble frequencies with enough accuracy. NLMS adaptive filter solves the optimization problem defined by Equation (3.4), which estimates the weight vector h_2'. Since convolution does not change the frequency of the signal, the attacker cannot make up for any offset existing between the correct frequency and the result from STFT. According to the frequency resolution problem of STFT as discussed in Section 3.4.3.4, the simulated attacker in our experiment gets an average frequency offset around 3 Hz, which makes it hard to descramble the recording.

3.6.9 Descramble Time

Sometimes when we grant recording permission to a specific speaker, the speaker would like to perform real-time descrambling. Patronus can achieve this working with real-time smart devices such as Amazon Alexa. To prove this, we measure the descramble time for records with different durations on a laptop with an Intel Core i7-4870HQ 2.5 GHz CPU. Since different high-order scramble waves (second-order component, third-order component, ...) may exist in a record simultaneously, we measure descramble time as a function of different max scramble orders, i.e., the upper bound of k in Algorithm 1. As shown in Table 3.1, Patronus can descramble the record quickly. Specifically, when the record time is 1 s, Patronus can finish descrambling in 328 ms, even when the max scramble order is 6. This means that Patronus supports real-time descrambling.

3.7 Limitations and Future Works

Range: In our implementation, we use cheap and low power ultrasonic transducers to build the Scramble Transmitter. The result is a short working distance, i.e., less than 70 cm. To enlarge the working area to a wider range of angles, we designed a reflection layer and verified that it could enlarge the working area by using an iron wok in our prototype. We can also use a high power ultrasonic speaker to protect a larger area. Some commercial off-the-shelf devices can emit ultrasound which could be sensed in a larger area. For example, UPS+ [3] uses an ultrasonic speaker with a working area of 50m $\times$ 50m. However, it is expensive. We can reduce the cost by deploying one expensive speaker and multiple transducers like UPS+[3]. Here we provide users with three options to deploy Patronus according to their requirements such as working area and budget. The first option is to use cheap transducers and a reflection layer to protect a small area. The second is combining an expensive speaker and multiple transducers to protect a larger area. The third is using multiple expensive speakers to protect the largest area.

Volume: In our implementation, we assume the talker uses a normal volume, i.e., not too loud or too quiet. However, the performance of Patronus does vary as a function of the speaker volume. For example, if the talker speaks too loudly, the scramble cannot mess up the recording; in the opposite

extreme, a quiet talker cannot be recovered using descrambling. To adapt to different volumes, we can add a microphone to measure the talker's volume. With multiple deployed ultrasonic speakers or transducers, we can first detect the position of recording devices and then adjust the power of ultrasound emitted from the nearest speakers according to the talker's volume. There are two challenges that need to be solved. First, the microphone we use to measure the talker's volume can also be scrambled. Second, we need to localize recording devices before emitting scrambles. We leave these challenges as future work.

CHAPTER 4

BREATHPASS: ULTRASOUNIC AUTHENTICATION BY CHEST AND ABDOMEN MOVEMENT WHILE BREATHING

4.1 Introduction

With the advancement of modern smart devices, unlocking methods have shifted away from the "what you know" schema and toward the "who you are" schema. With the "what you know" method, a user is needed to pre-configure some information such as PINs and secret questions, and the device will then challenge the user to verify that she or he actually owns the device. Such a PIN is often complex to ensure security and makes it difficult for individuals to remember to some level. In addition, these passcodes or answers are vulnerable facing blindly replay-attack since the devices do not care who is entering the information. With "who you are" tactics, the user no longer needs to type in the complex PIN, thus simplifying and speeding up unlocking. These approaches are quite popular with users because of their non-invasive nature and ease of use; e.g., Apple employs Face-ID to unlock the iPhone and iPad via facial recognition [76]. Apart from facial recognition, fingerprint identification is a frequently used method for unlocking smart gadgets [77]. In addition, voiceprint recognition [78], iris recognition [79], heartbeat recognition [80], breathing voice recognition [81], gaze gesture [82], and tooth-edge recognition [83] also plays a key role in biometric recognition approaches.

These approaches, however, have drawbacks in two different aspects.

Vulnerable to Replay-attack: Some of them are still compromised by replay-attacks, e.g., many research efforts [84, 85, 86, 87, 88, 89, 90] focus on resolving the replay-attack among voiceprint-based, fingerprint-based, gaze-based, or face-based authentication. For example, we could spoof others' face and voice with masks and recording.

Lack of Mobility and Flexibility: Other approaches using iris, tooth-edge, heartbeat, and human breath are not sufficiently flexible on mobile devices, e.g., iris-based authentication requires the

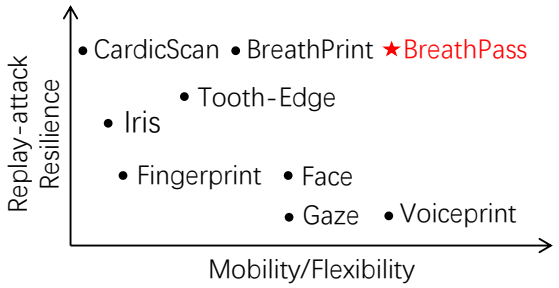


Figure 4.1: Comparasion of existing biometric authentication methods.

device to equip specific designed components such as inferred cameras, meanwhile, it needs users to look at a specific area to make sure that the inferred camera could capture a clear iris. Heartbeat-based authentication such as Cardiac Scan [80] requires the deployment of two radar sensors, which are not standard hardware and so have a high operating cost. BreathPrint [81] is a novel approach that doesn't need to equip specific designed components and can significantly defend against replay attacks, however, it cannot work in some scenarios including some people choose to wear a mask to protect themself from being infected by COVID-19 as the breathing voice that is needed by the system could be blocked by the face cover. The face cover also makes Smileauth [83] infeasible since it requires an image of the tooth-edge which is blocked by the face mask.

In this chapter, we propose BreathPass, a new non-invasive breath-based "who you are" authentication technique. BreathPass detects users' breath in a non-invasive manner, extracts features from their breath, and then verifies that the user is permitted. As shown in Figure 4.1, BreathPass is a novel approach that is hard to be compromised by replay attacks because breath pattern is hard to be spoofed and imitated. In addition, BreathPass is flexible enough since it only uses commercial off-the-shelf (COTS) components equipped on almost every devices, and can be used in a wider scenarios such as wearing different kinds of face covers and clothes, in different postures, and in different dynamic status such as walking or running. BreathPass faces the following challenges in order to implement it and achieve all of the aforementioned requirements:

1) As with BreathPrint, face covers may obstruct the voice of the user's breath. To overcome this challenge, BreathPass should avoid using a microphone to record users' breathing voices; instead, we employ an ultrasound-based chest wall and abdomen motion-sensing technology to characterize

users' breathing patterns. Specifically, BreathPass works by initially emitting ultrasonic waves through the speaker of a smart device, such as a smartphone. The ultrasonic waves then travel to the user's chest wall and abdomen, where they are reflected back to the smart device's microphone. The motion of the chest wall and abdomen, which characterizes human breath, alters the phase of the reflected signal, and such phase shifts are used to authenticate;

2) Unlike the speaker verification [91, 92] which normally converts the speech signal to a spectrogram in order to extract features, the motion of the chest wall and abdomen typically has an extremely low frequency of less than 1 Hz. As a result, features derived from spectrograms such as *Mel Frequency Cepstral Coefficients* (MFCC) or *Gammatone Frequency Cepstral Coefficients* (GFCC) [93] cannot be used to identify the breath. To address this issue, we implement the authentication mechanism using a one-dimensional Convolutional and Siamese Neural Network. Specifically, the neural network takes two raw chest wall and abdominal motion waveforms as the input. One of these two inputs is the template input collected from the enrollment stage, while the other is the matching input collected from the authentication stage. During training the neural network, it learns the breathing pattern and generates a vector of features, saying fingerprint, which can be used to calculate the distance between two inputs. Finally, BreathPass uses the distance between the two inputs to determine if they originate from the same person or not;

3) Unlike mechanical vibration, which typically has a stable frequency [94], breathing patterns between individuals do not share the same prior knowledge as mechanical vibration. Additionally, even when people are in the same posture, their breathing patterns may vary. In other words, small movements result in different breathing patterns. As a result, denoising the motion of the chest wall and abdomen requires developing a model that can suppress the moving-dependent noise while retaining the user-dependent difference. To address this issue, we introduce a technique called average fingerprinting. With such a technique, the template input is composed of multiple chest wall and abdomen motion signals that might come from different tiny postures. BreathPass generates multiple fingerprints from template signals using a neural network. Following that, the system computes the average of those fingerprints and then uses that average fingerprint to determine

the distance to the fingerprint obtained during the authentication stage. Finally, it calculates the authentication result using that distance.

Our contributions are listed as follows:

- We design a novel mechanism for sensing human breathing patterns and build a DNN to determine whether the breathing pattern provided by the user is authorized.
- By using the breath sensing mechanism and the DNN we built, we create BreathPass, which enables smart devices to perform authentication via the human breath. We also implement a proof-of-concept software to evaluate BreathPass's performance.
- On the basis of our implementation, we conduct extensive experiments. BreathPass achieves an 83% accuracy, a 73% true-positive rate (TPR), and a 5% false-positive rate (FPR) in general, according to the experiment results. The BreathPass system is stable when the user is wearing a variety of different face covers, clothing, and postures. We believe that in the future, it may be a candidate for a "who you are" unlocking mechanism, or it may serve as a complement to other untrustworthy mechanisms, such as eye recognition, in order to provide authentication services jointly.

The organization of the remainder of this chapter is as follows: In section 4.2, we illustrate the human breath preliminary and our system overview. In section 4.3, we introduce the design of BreathPass. In section 4.4, we introduce the implementation of BreathPass. In section 4.5, we present the results of our extensive experiments. In section 4.6, we summarize related works. In section 4.7, we discuss the limitation and the future work.

4.2 Preliminary

In this section, we illustrate the human breath process in detail and show the breath characteristics represented by chest/abdomen movement are diverse enough to be used for user authentication. In addition, we describe the overview of BreathPass.

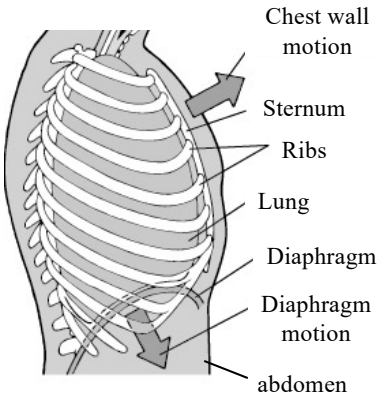


Figure 4.2: Illustration of chest/abdomen in the inhale step of a human breath process.

4.2.1 Human Breath Preliminary

As shown in Figure 4.2, multiple human body parts are involved in breathing. A respiration cycle is comprised of two steps: inhalation and exhalation. During the inhale step, air enters the body via the nose or mouth and travels to the lung. As a result of the chest wall expansion, the lung fills with air and expands; at the same time, the diaphragm at the bottom of the lung and at the top of the abdomen contracts. During this process, from the outside view, the chest wall expands, resulting in a larger chest cavity; at the same time, the abdomen expands as the diaphragm contracts. During the exhale step, on the contrary, the air will flow out the body from the nose or mouth. Because the human body is devoid of air, the chest wall contracts. Meanwhile, the diaphragm relaxes, resulting in a smaller chest cavity and abdomen.

Certain clinical studies indicate that individuals' breathing patterns vary. Kaneko et al. [95], in particular, conducted an experiment in which three sensors were placed on the thorax and abdomen and a vector of three-dimensional movement was measured during breathing. The observed breathing movements were found to be related to the effects of age, sex, and posture. Raganarsdottir et al. [96] discovered that women have fewer abdominal movements than men do during deep breathing. Additionally, the previous study [81] cites some clinical studies [97, 98] to demonstrate that individual signatures of breath composition do exist.

Remark: We believe that by fully characterizing the movements of the chest wall and abdomen, distinct signatures for individual recognition can be extracted.

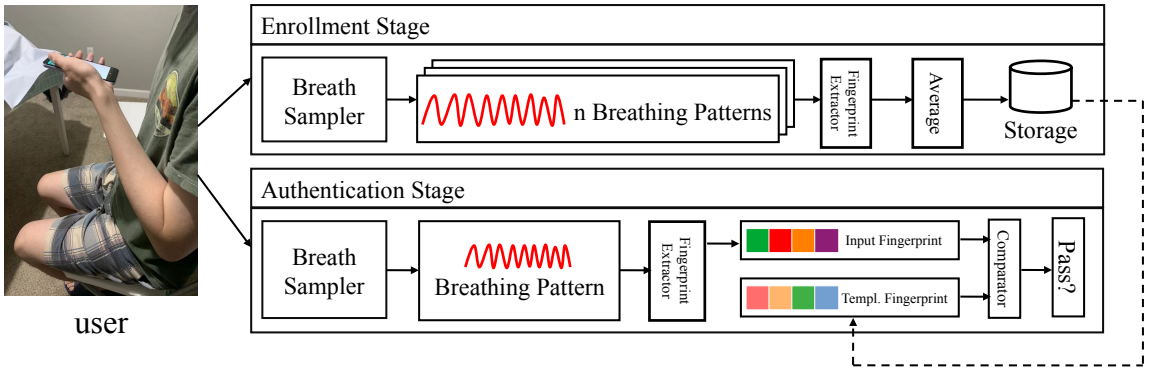


Figure 4.3: Overview of BreathPass system that consists of an enrollment stage and an authentication stage.

4.2.2 System Overview

As shown in Figure 4.3, BreathPass enables authentication in two stages: the enrollment stage and the authentication stage. The enrollment stage's objective is to sample the user's breath waveforms in a variety of different states (called *tiny postures*) using a breath sampler (Section 4.3.1) that emits a harmonic ultrasound signal. The different tiny gestures for BreathPass are just like the different edge parts of a fingerprint. To ensure robustness at all common circumstances, we sample breath waveform multiple times at the enrollment stage. Following that, the enrollment stage utilizes a DNN-based fingerprint extractor (Section 4.3.2) to extract **n** feature vectors, referred to as fingerprints, and then calculates the average of those fingerprints, referred to as the template fingerprint, and stores it in the local storage.

The authentication stage is similar to the enrollment stage, except that the breath sampler samples only one breath waveform. Following that, it uses the same fingerprint extractor as in the enrollment stage to extract the input fingerprint. The template fingerprint is then fetched from local storage and combined with the input fingerprint in the comparator (Section 4.3.3) to determine whether the sampled breath is from an authorized user.

4.3 Design

We begin this section by analyzing the properties of ultrasonic signals in order to provide guidelines for the chest/abdomen movement tracking design (§4.3.1). Then we go into detail about how we

extract the fingerprint of the breathing pattern (§4.3.2), how we determine whether an authentication fingerprint matches the enrolled one (§4.3.3), and finally about the BreathPass design workflow (§4.3.4).

4.3.1 Ultrasound-based Breath Sampler

Breathing Pattern Extraction: We use the speakers on smart devices, such as a smartphone, to play a stereo ultrasound signal. As shown in Figure 4.3, the speaker is perpendicular to and close to the chest wall. The left channel plays an ultrasound signal at an 18 kHz frequency, while the right channel plays one at a 22 kHz frequency. The ultrasound signals are reflected off the chest wall and abdomen and are picked up by the smartphone's microphone, which is also positioned near the chest wall. Formally, the signal emitted is denoted as follows:

$$s(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t), \tag{4.1}$$

where $f_1 = 18,000$ and $f_2 = 22,000$. After the microphone records the reflected signal $m(t)$, the breath sampler first employs a high pass filter to eliminate components below 16 kHz. Then, inspired by the previous efforts [99, 10], the breathing pattern can be regarded as the signal $x(t)$ modulated into $m(t)$ by the carrier of $s(t)$. Therefore, we have

$$m(t) = x(t)s(t). \tag{4.2}$$

To demodulate the breathing pattern $x(t)$, we need to multiply $m(t)$ by $s(t)$ and let the result pass through a low pass filter with an extremely low cutoff frequency, e.g., 200 Hz. From Equation (4.1) and (4.2), we have

$$\begin{aligned} m(t)s(t) =\ & x(t)s^2(t) = x(t)[\cos(2\pi f_1 t) + \cos(2\pi f_2 t)]^2 \\ =\ & x(t)[\cos^2(2\pi f_1 t) + 2\cos(2\pi f 1_t)\cos(2\pi f_2 t) \\ & + \cos^2(2\pi f_2 t)] \\ =\ & x(t)\{\frac{1}{2}[1 + \cos(2\pi 2 f_1 t)] + \cos(2\pi(f_1 + f_2)t) \\ & + \cos(2\pi(f_2 - f_1)t) + \frac{1}{2}[1 + \cos(2\pi 2 f_2 t)]\}. \end{aligned} \tag{4.3}$$

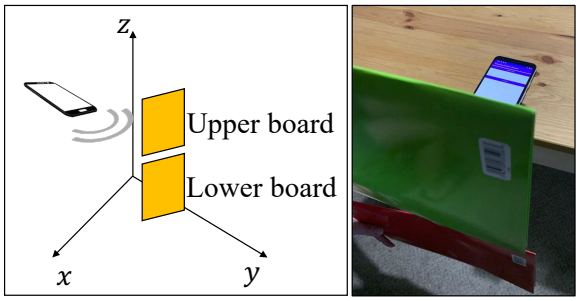


Figure 4.4: A controlled experiment verifying our ultrasound frequency selection. The board movement to mimic the chest wall and abdomen motion during breathing.

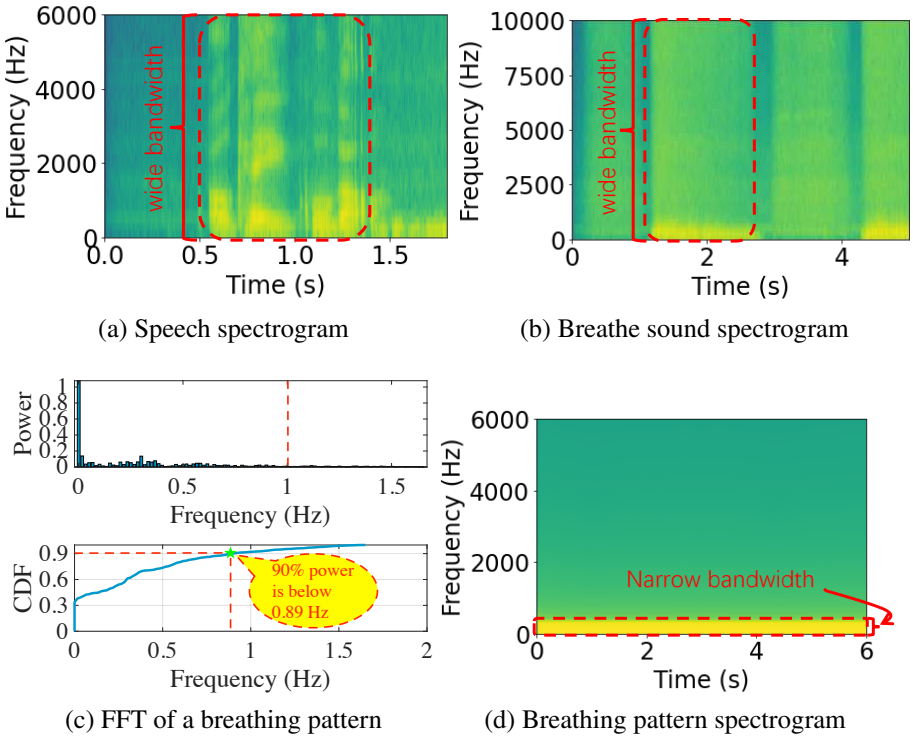
After a low pass filter with a 200 Hz cutoff frequency, the components $\cos(2\pi 2 f_1 t)$, $\cos(2\pi 2 f_2 t)$, $\cos(2\pi(f_1 + f_2)t)$, and $\cos(2\pi(f_1 - f_2)t)$ all disappear. Therefore, we have

$$m(t)s(t) \Longrightarrow x(t)(\frac{1}{2} + \frac{1}{2}) = x(t). \tag{4.4}$$

We use the extracted $x(t)$ as the breathing pattern to perform authentication.

Ultrasound Frequency Selection: We want to emphasize why we use a pair of ultrasound signals with two frequencies rather than a single frequency wave. Because the higher frequency signal is more easily attenuated with distance than the lower frequency signal. If we place an ultrasound speaker perpendicular to the chest wall and play a pair of ultrasound signals, the lower frequency signal is more likely to reach the abdomen, whereas the higher frequency signal can only reach the chest.

To verify this, we conduct the experiment depicted in Figure 4.4. We place two boards perpendicular to the y-axis and place a smartphone where its speaker towards the upper board. The distance between the upper board and the speaker is 10 cm. The speaker plays the ultrasound signal with the frequency of 18 kHz and 22 kHz, alternatively. When the speaker plays a particular frequency ultrasound, we move the upper and the lower board back and forth along the y-axis at a time to simulate only the chest wall moving, or only the abdomen moving, and extract the motion waveform similar to Equation (4.4). We calculate the power under each scenario, and denote P_{18u} as the power of the sensed motion waveform of the upper board by using the 18 kHz ultrasound, and denote P_{18l} as the power of the sensed motion waveform of the lower board by using the 18 kHz ultrasound. Similarly, we denote P_{22u} as the power of the sensed motion waveform of the



(a) Speech spectrogram

(b) Breathe sound spectrogram

(c) FFT of a breathing pattern

(d) Breathing pattern spectrogram

Figure 4.5: (a) Spectrogram of a speech "OK, Google!". (b) Spectrogram of a breathing sound. (c) FFT and CDF of a breathing pattern. (d) Spectrogram of a breathing pattern.

upper board by using the 22 kHz ultrasound and denote P_{22l} as the power of the sensed motion waveform of the lower board by using the 22 kHz ultrasound.

To determine the sensitivity, we must compare $Q_l = \frac{P_{18l}}{P_{22l}}$ with $Q_u = \frac{P_{18u}}{P_{22u}}$. Q_l and Q_u indicate how much power the 18 kHz ultrasound can sense when the 22 kHz ultrasound can sense 1 unit power of the lower and upper board movement, respectively. In our experiment, $\frac{Q_l}{Q_u} \approx 1.61 > 1$, indicating that 18 kHz is more sensitive to lower board movement than 22 kHz. Similarly, 22 kHz is more sensitive to upper board movement than 18 kHz. As a result of the fact that ultrasound at 22 kHz is more sensitive to chest wall motion than ultrasound at 18 kHz, we use a pair of ultrasound signals to fully characterize the motion of the chest wall and abdomen.

4.3.2 Fingerprint Extractor Design

Design Issues: After sampling the breathing pattern, both the enrollment stage and the authentication stage all send their samples to the fingerprint extractor. In order to get a feasible fingerprint that

can be used to perform authentication, the design of fingerprint extractor should take the following challenges into consideration:

1) *Denoise.* Previous works [94, 10, 100] proposed multiple approaches to denoise the vibration waveform or the breathing waveform for machine damage or human disease detection. For example, mmVib [94] reports the machine error when an abnormal vibration is detected. The system collects the vibration waveform with noises and leverages a model to denoise the signal. After that, the system will measure the distance between the sampled signal and the normal status signal. If the distance is within a threshold, then the system classifies the machine works normally. Otherwise, the system reports the machine in an abnormal status. To build such a denoise model, the system usually collects vibration waveform with noises when the machine works normally and use a series of transformation and processes, e.g., matching arc on the I-Q plane [94], to match the noise waveform with the standard vibration waveform as precise as possible. After matching, it fixes the processes and parameters to denoise future signals. If the machine works in abnormal status, the signal after being processed by the same model with the same parameters is far from the standard one. Such an idea was also adopted by SpiroSonic [10] and BreathListener [100] to detect if human breathes normally. The common point of these works is to find the identical pattern when the machine or the lung works normally. In BreathPass, however, the goal is to characterize the difference in the breathing pattern among different people instead of finding the typical pattern from different peoples' breaths. Therefore, it is hard for us to build a denoise model by extracting the common pattern.

2) *Stability.* The chest wall and the abdomen motion are not as stable as a machine. A different breathing pattern may be extracted even the user stays in the same posture, but after a tiny movement; e.g., when a user leans back on a chair from the straight waist, a different breathing pattern will be extracted. Therefore, the design of the fingerprint extractor must take such a stability issue into consideration.

3) *Feature selection.* The most similar task to BreathPass is speaker verification. The speaker verification task first uses a microphone to record the user who is saying a predefined sentence

or any other sentence. The system then verifies if the recorded voice comes from the authorized users. Existing speaker verification solutions typically first transform the voice signal into the spectrogram, then extract spectrogram-based features such as *Mel Frequency Cepstral Coefficients* (MFCC) or *Gammatone Frequency Cepstral Coefficients* (GFCC) [93]. After that, such solutions leverages the *Gaussian Mixture Model* (GMM) or build a Deep Neural Network (DNN)-based model to verify the speaker.

BreathPrint [81] adopts a similar idea. Different from the speaker verification task is that it uses a microphone to record the user's breathing voice, then extracts GFCC features and leverages the GMM model to verify if the breathing voice comes from the authorized user.

To extract spectrogram-based features, the system needs to get a signal with a reasonable wide bandwidth; e.g., speaker verification and BreathPrint [81] are both capable of leveraging spectrogram-based features since the spectrogram of a speech "OK, Google!" could reach up to 6 kHz as shown in Figure 4.5a, and the spectrogram of a breath sound can reach up to 10 kHz as shown in Figure 4.5b. Therefore, there is a sufficient area to embed information in the spectrogram so that these systems are able to use photo verification-liked models to perform authentication.

In BreathPass, however, we cannot use spectrogram-based features since the bandwidth of a breathing pattern is extremely narrow. An adult typically finishes a breathing cycle in 2 to 3 seconds. We plot the result of the Fast Fourier Transform (FFT) of a breathing pattern that we sampled with the method described in Section 4.3.1 as shown in Figure 4.5c. We can find that the majority of the power is under 1 Hz (90% of power is below 0.89 Hz). Therefore, the bandwidth of a breathing pattern in the spectrogram is too narrow as shown in Figure 4.5d to provide sufficient information that can be used to perform authentication.

Our Solution: To address the first challenge that we cannot build a denoise model based on observation, we, therefore, build a DNN-based model to learn how to denoise the signal and extract the fingerprint itself. To cope with the third challenge, we use the raw breathing pattern waveform as the input instead of extracting spectrogram-based features.

As shown in Figure 4.6, the fingerprint extractor is consists of a series of convolutional layers

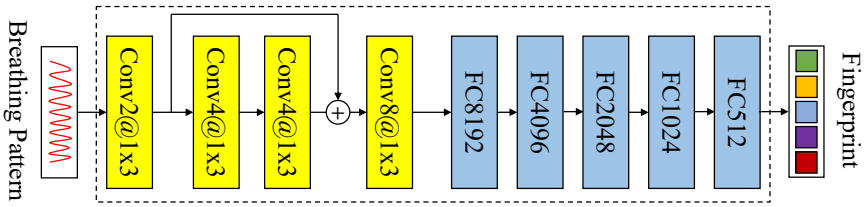


Figure 4.6: The structure of our DNN model for fingerprint extractor.

followed by some fully connected layers. Each layer uses the ReLU function to activate, and there is a max-pooling layer with a length of 4 after the first and the last convolutional layers. Each fully connected layer except the last one adopts dropout with the parameter 0.2 to avoid overfitting. We also adopt the idea of ResNet [101] of adding a skip link between convolutional layers to avoid gradient vanishing. The fingerprint extractor takes a breathing pattern waveform sampled by the method described in Section 4.3.1 as the input. After the last fully connected layer of the extractor, it outputs a vector of 512 floating-point numbers as the fingerprint.

The second challenge about stability requires us to remove moving-dependent noises while reserving the user-dependent difference among different users' breathing patterns. To cope with this issue, we introduce the average fingerprint technique. Specifically, instead of using the fingerprint that comes from a single breathing pattern waveform as the result of the enrollment stage, we sample multiple breathing patterns in the enrollment stage and get multiple fingerprints correspondingly. We calculate the average of these fingerprints as the result of the enrollment stage.

The idea behind the average fingerprint is that if we focus on the same user, moving-dependent noises are unstable while the user-dependent difference is stable, therefore, if we take the average of multiple fingerprints, unstable moving-dependent noises will be smoothed while the stable user-dependent difference is reserved. We show the effectiveness of the average fingerprint in Section 4.5.10.

4.3.3 Comparator Design

After getting fingerprints from both the enrollment stage and the authentication stage, we need to build a comparator to measure the distance between two fingerprints.

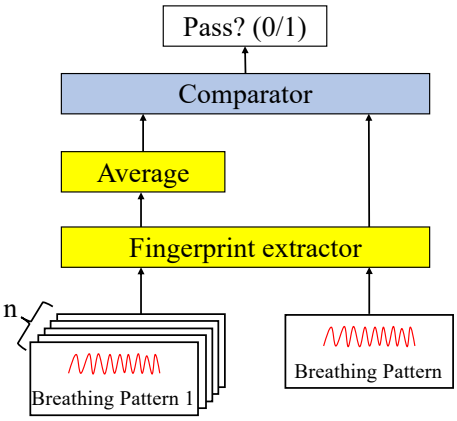


Figure 4.7: The end-to-end system design combining the fingerprint extractor with the comparator.

In general, there are two approaches to building such a comparator. The first one is applying triplet loss [102] while training the feature extractor. With this approach, we need to provide a triplet (A, P, N) as the training data, where A and P are the breathing patterns from the same person, while N is from another person. The goal of triplet loss is to find a DNN parameter that outputs fingerprints (A', P', N') satisfies $d(A', P') + \alpha \leq d(A', N')$, where $d(\cdot)$ is the euclidean distance or the cosine similarity.

In our practice, however, it is hard to train the fingerprint extractor with the triplet loss. Therefore we adopt another idea [103]. After getting the fingerprints, instead of building a comparator with the triplet loss, we build the comparator by applying logistic regression. Specifically, we have the target function

$$f(x, y) = \sigma(w^T \|x - y\|_2 + b), \tag{4.5}$$

where $\sigma(\cdot)$ is the sigmoid function, w is the vector of parameters of the comparator, b is the bias, and x and y are the fingerprints from the enrollment stage and the authentication stage, respectively. During training the comparator, the target output $f(x, y)$ is set to 1 if the x and y are from the same user, otherwise, the target output is set to 0.

4.3.4 Combine the Fingerprint Extractor with the Comparator

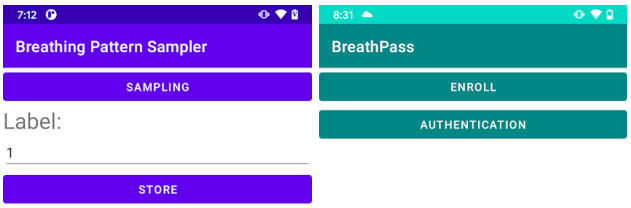
As shown in Figure 4.7, we put these components together. The fingerprint extractor and comparator are combined into a single neural network. During the training process, we randomly choose n breathing patterns from the same volunteer in the training dataset and choose another breathing pattern from a random volunteer in the training dataset. If these two volunteers are the same one, then the final result, i.e., `Pass?`, is set to 1; otherwise, it is set to 0.

As for the deployment of these components, the left lower side of the figure, i.e., n breathing patterns, comes from the enrollment stage. We store the average fingerprint in advance as shown in Figure 4.3. During the authentication stage, the system samples a breathing pattern as shown on the right lower side of the Figure 4.7. If the output of the comparator is greater than 0.5, we denote the final result, i.e., `Pass?`, as 1, indicating that authentication was successful; otherwise, we denote the final result as 0, indicating that authentication failed. If authentication fails, BreathPass prompts the user to sample his breathing pattern and attempt authentication again; if authentication continues to fail, BreathPass will prevent the user from sampling his breathing pattern until the user enters the correct PIN number.

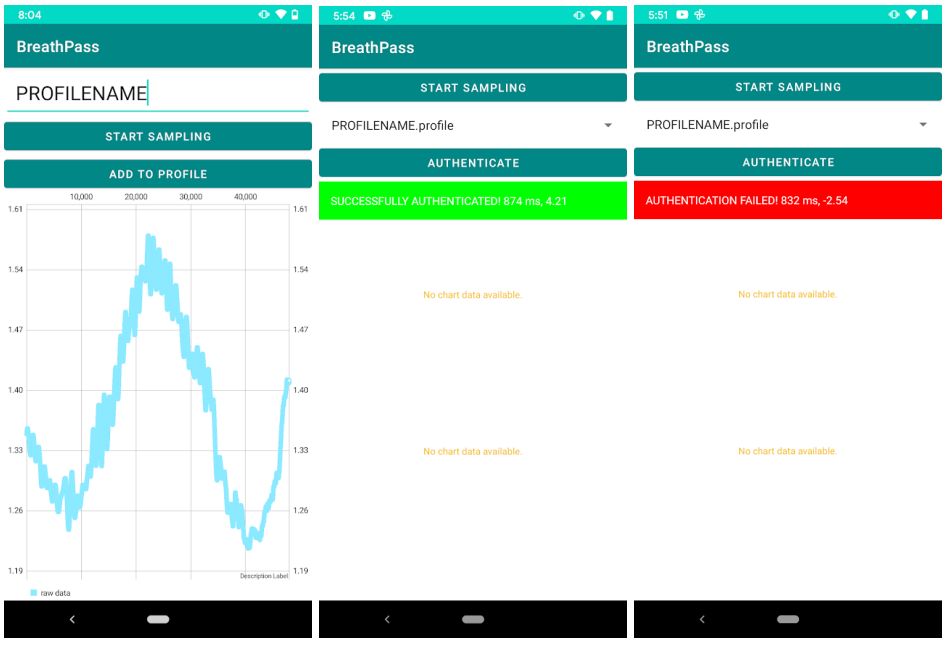
4.4 Implementation

4.4.1 Breathing Pattern Sampler and Data Collection

We develop the breathing pattern sampler on Android smartphones, as shown in Figure 4.8a. We use the native Android library *AAudio* [104] to generate, emit, and record the ultrasound waves. With the approval of our IRB under expedited review, we use our sampler to collect data and extract the breathing patterns of 20 volunteers. Each volunteer is continuously sampled for 60 seconds and five times (i.e., 300s in total). The 20 volunteers cover people of different gender and age that may frequently use smart devices. We use a Google Pixel 3a smartphone running Android 11 to perform sampling. The sampling rate is set as 48 kHz. During sampling, we place the smartphone on a desk and let the speaker towards a volunteer's chest. The distance between the smartphone and



(a) Breathing pattern sampler (b) BreathPass



(c) Enrollment (d) Successfully authenticated (e) Failure to authenticate

Figure 4.8: The UI of BreathPass implementation on a smartphone. (a) the breathing pattern sampler for general data collection; (b)-(e) the pages of our proof-of-concept application.

the volunteer is between 5 and 10 cm. An interval separates two consecutive samplings to allow the volunteer to adjust their tiny posture. Once the microphone samples the reflected ultrasound signals, we leverage *Apache Commons Math* package [105] to build a high pass filter that eliminates all components below 16 kHz, leaving only the ultrasound signals and then extracts the breathing pattern using the design described in Section 4.3.1. The extracted breathing patterns are indexed by number, as shown in Figure 4.8a. Specifically, 1 to 5 represent the first volunteer, 6 to 10 represent the second volunteer, and so on.

4.4.2 Training the Feature Extractor and Comparator

After getting the dataset from 20 volunteers, we build the feature extractor and comparator using PyTorch on a desktop equipped with an NVIDIA GeForce RTX 3090 GPU, as discussed in Section 4.3.2 and Section 4.3.3. We randomly select 10 volunteers' data for the training set and the remaining volunteers' data for the test set. During each iteration of the training and testing, we first randomly select a volunteer, then we randomly choose a 60s long breathing pattern from the indexed volunteer dataset, and finally, we randomly crop a segment of the 60s long breathing pattern ranging from 1s to 5s. This process is repeated 10 times to create the template inputs. Then we get another segment of breathing pattern but alternatively choose the same volunteer and a different volunteer as the authentication input. If we have chosen the same volunteer, the target of the DNN output is set to 1; otherwise, we set it to 0. We also add some fake breathing patterns which are collected by the breathing pattern sampler with the smartphone speaker towards the wall or towards nothing to enhance the classification accuracy. We always set the target of the DNN output to 0 if any of these fake patterns are chosen.

4.4.3 Proof-of-concept Application

To explore the efficiency of the system on different smart devices in practice (see Section 4.5.11), we develop a proof-of-concept Android application as shown in Figure 4.8b, 4.8c, 4.8d, and 4.8e. We port our PyTorch model built in Section 4.4.2 with TorchScript [106] and load it into our

application. As shown in Figure 4.8b, our application provides the enrollment stage and the authentication stage. The enrollment stage (Figure 4.8c) integrates the breathing pattern sampler discussed in Section 4.4.1, but we only sample 1 second long each time. Here we choose the 1-second segment because experiment results indicate that this segment length provides the best performance. After sampling the breathing patterns, by clicking *ADD TO PROFILE* button, our app launches the DNN-based fingerprints extractor to obtain the fingerprint, and stores a copy into a directory named `PROFILENAME.profile`. The app will display a segment of the breathing pattern to ensure that it is correctly extracted.

As shown in Figure 4.8d and 4.8e, during the authentication stage, the user first samples his breathing pattern, then chooses which template profile is going to be authenticated. After hitting the *AUTHENTICATE* button, the app will fetch the corresponding fingerprints from the directory `PROFILENAME.profile` and calculate their average. Then the app launches the fingerprint extractor model to generate the fingerprint of sampled breathing patterns. Finally, it launches the comparator to perform an authentication process and displays whether the authentication is successful.

4.5 Evaluation

4.5.1 Overview

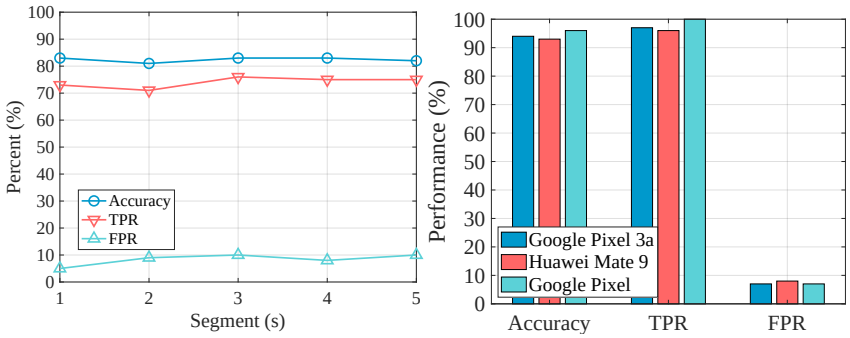
To evaluate BreathPass, we use the data collected in section 4.4 to train and test BreathPass. In general, we use the following metrics to evaluate the performance of BreathPass:

Accuracy: We use accuracy to determine whether BreathPass can correctly identify the authorized user whose fingerprints are stored during the enrollment stage. The accuracy is calculated as

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = y_i)}{N}, \tag{4.6}$$

where N is number of test cases, I is the indicator function, $\hat{y}_i$ is the output given by BreathPass, and y_i is the correct label. In general, the greater the accuracy, the better.

True positive rates (TPR) and false positive rates (FPR): Besides the accuracy, we also focus on two metrics, i.e., true positive rates (TPR) and false positive rates (FPR). We calculate the TPR



(a) (b)

Figure 4.9: (a) General performance of BreathPass (b) Performance of different mobile models.

by using the equation

$$\text{TPR} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = 1 \text{ and } y_i = 1)}{\sum_{i=1}^{N} I(y_i = 1)}, \tag{4.7}$$

and the FPR is calculated by

$$\text{FPR} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = 1 \text{ and } y_i = 0)}{\sum_{i=1}^{N} I(y_i = 0)}, \tag{4.8}$$

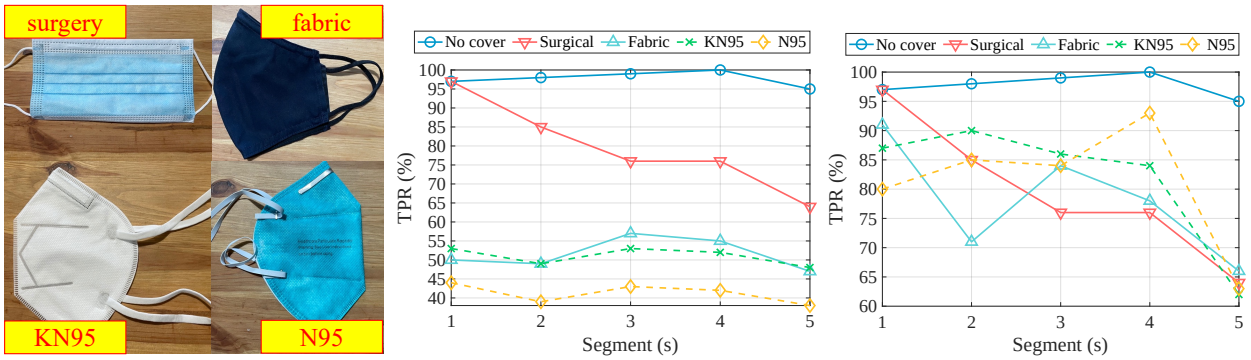
where N is the number of test cases, I is the indicator function, $\hat{y}_i$ is the output given by BreathPass, and y_i is the correct label.

When an enrolled user attempts to unlock the device, the TPR determines the likelihood that the system will successfully authenticate. When an unauthorized user attempts to unlock the device, the FPR determines the possibility that the system will pass the authentication by mistake. The higher the TPR, the better, while the lower the FPR, the better.

Apart from the TPR and FPR, two additional metrics are used to characterize the authentication system's performance: true negative rates (TNR) and false negative rates (FNR), which indicate the likelihood of an unauthorized user being successfully blocked by the system and the likelihood of an authorized user failing the authentication, respectively. However, we are unconcerned with these two values because an attacker cannot do anything if the device cannot be unlocked.

With these three metrics, we would like to answer the following questions:

- Is it possible that the breathing pattern we sampled could be used to perform authentication and can BreathPass become a candidate of "who you are" scheme under the COVID-19



(a) Masks (b) TPR w/o grouping (c) TPR w/ grouping

Figure 4.10: Performance of BreathPass with different kinds of clothes.

scenario?

- Is BreathPass performing well to the same user but wearing different kinds of face covers, clothes, with different postures, with dynamic status, under different environments, and on different mobile models?
- Can BreathPass defend against replay attacks?
- What's the impact if we use the single template breathing pattern to generate the fingerprint rather than using the average fingerprint?
- Can BreathPass finish the authentication within a reasonable time limit?

4.5.2 General Evaluation

Setup: To determine whether the extracted breathing pattern can be used for authentication, we train and test the fingerprint extractor and comparator using the dataset collected in Section 4.4. We have formed the training dataset by randomly choosing 10 volunteers from the whole dataset. In this experiment, we use the remaining 10 volunteers as well as the fake breathing patterns to perform testing. We perform 1000 iterations of testing. During each iteration, we randomly choose 1s to 5s breathing pattern segments as described in Section 4.4.1 to form the test datasets with mini-batches of 32, resulting in a total of $32 \times 1000 = 32000$ test cases. The output of the sigmoid

function in Equation (4.5) is in the range of [0, 1]. If the output is greater than or equal to 0.5, the result is considered passed; otherwise, the result is considered failed.

Results: As shown in Figure 4.9a, BreathPass achieves over 80% accuracies, over 70% TPRs, and less than 10% FPRs for any segment length of the input breathing pattern. BreathPass achieves an accuracy of 83%, a TPR of 73%, and an FPR of 5% when the input breathing pattern is segmented for 1 second, which is the best segment length. As a result, we assert that the breathing pattern we sampled can be used for authentication and that when combined with the TPR and the FPR, BreathPass can serve as a candidate for a "who you are" scheme, either alone or in conjunction with eye recognition in the COVID-19 scenario.

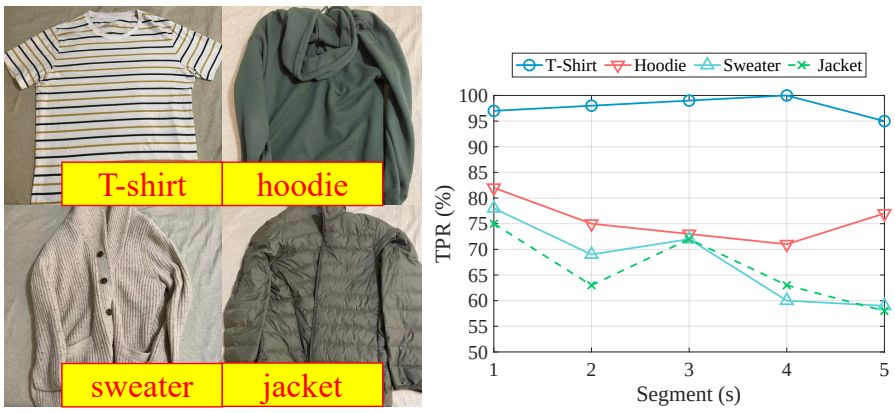
4.5.3 Effectiveness on Different Mobile Models

Setup: To verify whether BreathPass is able to work on different mobile models, we launch BreathPass on three different mobile phones, i.e., Google Pixel 3a, Huawei Mate 9, and Google Pixel. In this experiment, we use these three mobile models to collect a volunteer's breathing pattern respectively. Then we form the positive test cases by selecting pairs of 1s breathing patterns from the collected breathing pattern. After that, we make pairs of the breathing patterns from a given mobile model and the test datasets as discussed in Section 4.5.2 as the negative cases.

Results: As shown in Figure 4.9b, there is little difference between accuracies, TPRs, and FPRs among different mobile models. Therefore, BreathPass can work on different mobile models.

4.5.4 Influence of Different Kinds of Face Covers

Wearing a face cover may obstruct airflow into the user's nose or mouth, thereby altering the user's breathing pattern. To characterize the effect of various types of face covers on users, we prepared four types of commonly used face covers, i.e., surgical, fabric, KN95, and N95, as shown in Figure 4.10a. There are almost no blocks when wearing the surgical mask, while the remaining makes breathing harder than wearing the surgical mask or not wearing a face cover. We would like to characterize the performance across different kinds of face covers. In this experiment we only



(a) Clothes (b) TPR

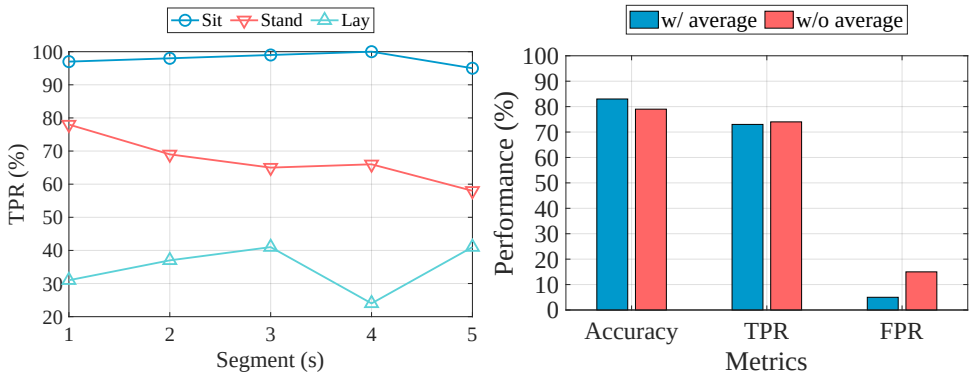
Figure 4.11: Performance of BreathPass with different kinds of clothes.

care about TPRs, which means the possibility of successfully authenticated while wearing different face covers.

Setup without grouping: In this experiment, we invite a volunteer to wear each of the four types of face covers separately and evaluate the BreathPass's performance. We ask the volunteer to enroll his breathing pattern with no face cover, and perform authentication with wearing different kinds of face covers.

Results without grouping: As shown in figure 4.10b, TPRs decrease while wearing the masks which blocks the airflow, but almost all of them are over 40%, which means that the extracted breathing pattern is still feasible while wearing different kinds of face covers.

Setup with grouping: To further improve the TPRs while wearing the face covers that blocks the airflow, we can split the face covers into two groups, i.e., no airflow blocked (no face covers and surgical) and airflow blocked (fabric, KN95 and N95). We ask the volunteer to enroll with one of them in a group and perform authentication with wearing another one in that group. Specifically, we firstly use breathing patterns collected without a face cover to generate the template fingerprint and use breathing patterns collected with the surgical mask as the input of the authentication stage. Then we use the KN95 dataset to generate the template fingerprint and use breathing patterns from the fabric, and the N95 dataset, respectively, as the input of the authentication stage. Finally, we generate the template fingerprint using breathing patterns from the N95 dataset and use it to evaluate



(a) Posture TPR (b) Performance w/ or w/o average

Figure 4.12: (a) TPR of BreathPass with different postures. (b) Performance with or without average fingerprint technique.

performance when wearing the KN95 mask. This is reasonable, as the user could enroll in both groups separately and choose one manually or automatically before performing the authentication. We use datasets of the volunteers that form the test datasets in Section 4.4.2 to test negative cases. The volunteer in this experiment is not one of the volunteers in Section 4.4.2.

Results with grouping: As shown in figure 4.10c, compare to the TPRs without face cover, all TPRs with a face cover are decreased, but most of the TPRs are higher than 70%, and in particular, for 1s, the TPRs are all higher than 80%. Therefore, BreathPass is feasible across different face covers.

4.5.5 Influence of Different Clothes

Setup: Since BreathPass extracts breathing patterns from the motion of the chest wall and the abdomen, the breathing pattern collected might be influenced by different clothes because different wearings might have different effects of blocking ultrasound signals. In this experiment, we choose the most common used four kinds of clothes, i.e., T-shirt, hoodie, sweater, and jacket, as shown in Figure 4.11a, and invite a volunteer to sample his breathing patterns while wearing these clothes, correspondingly. We then use the dataset collected with wearing the T-shirt to generate the template fingerprint, and use the breathing pattern from datasets with wearing all four clothes correspondingly as the input of the authentication stage. The volunteer in this experiment is not

one of the volunteers in Section 4.4.2.

Results: As shown in figure 4.11b, the TPRs are almost higher than 65%, and in particular, for 1s, the TPRs are all over 75%. Therefore, BreathPass is feasible across different kinds of clothes.

4.5.6 Influence of Different Postures

Setup: As discussed in Section 4.3.1, different postures result in different breathing patterns. Therefore, to characterize the influence of different postures, we invite a volunteer to provide breathing patterns by the method discussed in Section 4.4.1 with the three most common postures, i.e., sitting, standing, and laying down. We use breathing patterns extracted from sitting posture to generate the template fingerprint, and use breathing patterns extracted from all these three postures respectively as inputs of the authentication stage. The volunteer in this experiment is not one of the volunteers in Section 4.4.2.

Results: As shown in figure 4.12a, the sitting and standing posture have higher TPRs than laying down. The TPRs for sitting and standing are almost higher than 60%, and in particular, for 1s, the TPRs are all over 70%. Therefore, BreathPass is feasible across different postures.

4.5.7 Influence of Dynamic Status

Setup: Some dynamic status such as walking or after running may result in different breathing pattern, to verify if BreathPass could still successfully authenticate the user under these dynamic status. We ask a volunteer to enroll his breathing pattern while sitting in a quiet room at rest, and perform authentication while sitting in a quiet room at rest (marked baseline), during walking, and after running 500m, respectively. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases. The volunteer in this experiment is not one of the volunteers in Section 4.4.2.

Results: As shown in table 4.1, walking has almost no effect to authentication. Authentication after running has a bigger effect as it significantly changes the breathing pattern, however, it still

Class	Baseline	Walking	Running	Outside	TV
TPR	97%	94%	78%	78%	80%

Table 4.1: TPRs of different dynamic status and environments.

achieves 78% of the TPR, which means that BreathPass is feasible when the user is under dynamic status.

4.5.8 Influence of Different Environments

Setup: To verify if BreathPass could still successfully authenticate under different environments. We ask a volunteer to enroll his breathing pattern while sitting in a quiet room at rest, and perform authentication while sitting in a quiet room at rest (marked baseline), outside while raining (lower noise), and near a TV set playing a concert with a high volume (higher noise), respectively. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases. The volunteer in this experiment is not one of the volunteers in Section 4.4.2.

Results: As shown in table 4.1, compare to baseline, authentication outside while raining and near the TV set decrease the TPR. It probably because the raining falling down between the speaker and the chest wall affects the transmition and reflection of the ultrasound signals, and suppression effects of the microphone [8] affects the recording quality when background noise is huge. The TPRs however, are around 80%, which means that BreathPass is feasible under different environments.

4.5.9 Defend Replay Attacks

Setup: It is unlikely to replay a breathing pattern by recoding other's breath like voice recording does. In addition, we cannot spoof our authentication system with an easy way just like making face/fingerprint/iris masks. The best way we can issue a replay attack is to make an attacker to observe and imitate users' breath pattern. To verify whether BreathPass could defend against replay attacks, we ask four groups of volunteers (each contains two volunteers) to sit in the same room at the same seat. In each group, two volunteers are similar in terms of gender, age, height and weight.

In each group, a volunteer acts as an attacker to imitate the other's breath by controlling the pace of breathing and get the same times of breathing cycle in a minute. Then we use the non-attacker volunteer's breathing pattern in a group to enroll the BreathPass and use the breathing pattern from the attacker in that group to perform authentication. The segment of breathing pattern is 1 s in the experiments. We only care about FPRs here to see if such a replay attack can break in the system.

Results: We get four FPRs, 3.6%, 3.8%, 9%, and 31% for each of the groups. We can see the FPRs of the first three groups are lower than 10% which is impossible to be used to attack the authentication process. For the forth group, the FPR becomes higher to 31%. The reason is that the volunteers are more like "twins" than other groups. Their body shape, habit, and exercise-level are similar as well. Therefore, the imitated breathing pattern may be similar with the original one. Even though the FPR is 31% which is far less than our overall TPR 73%. Saying that, we can sacrifice the computation efficiency (e.g., continuous two successful authentication) to improve the security if the user wants.

4.5.10 Effectiveness of the Average Fingerprint

Setup: During our experiment, we found that even all volunteers in Section 4.4.1 are sitting while sampling their breathing patterns, the DNN-based model also cannot get a good performance. This is because even a tiny move within the same posture could result in different breathing patterns that affect the overall performance. As discussed in Section 4.3.2, we introduce an average fingerprint technique to eliminate moving-dependent noises while reserving user-dependent differences. In this experiment, we build another model of the same DNN architecture as discussed in Section 4.3 but without the average fingerprint technique. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases. We compare the performance between models with and without the average fingerprint technique to show the effectiveness of the average fingerprint technique. Specifically, we use the same training dataset to train the same model without the average fingerprint technique. After the model converges, we use the same test dataset as Section 4.5.2 to test the performance.

Results: As shown Figure 4.12b, we can find that the accuracy without the average technique is lower than the model with the average technique. We can further find that although they have close TPRs, the FPR without the average technique is much higher than the model with the average technique, which is unacceptable. The reason why the model without the average technique has a high FPR is because the model cannot eliminate moving-dependent noises, thus taking them as the feature to construct the fingerprint. Therefore, it is necessary to apply the average fingerprint technique so that the model could successfully eliminate moving-dependent noises while reserving user-dependent differences.

4.5.11 Efficiency on Mobile Phones

Setup: To make BreathPass practical, the DNN-model needs to finish the inference on a mobile device within a reasonable time limit after a user samples his breathing pattern. To test the efficiency of BreathPass, we port our model on a Google Pixel 3a Android mobile phone as discussed in Section 4.4.3. The application shows the time used by the DNN-model along with the authentication results. We perform 10 times of authentication. The configuration is the same as the previous experiments, and we use 1s segment length of breathing patterns as the inputs. Specifically, we enroll 10 breathing signals that each of them is 1s long, and extract 10 fingerprints, respectively, and store them on the smartphone. During the authentication stage, after the user samples his 1s breathing pattern, we first calculate the average of 10 fingerprints, then take the result of the average and sampled breathing pattern as the input to the model. The model extracts the fingerprint of the sampled breathing pattern and runs the comparator to give the result of the authentication.

Results: We calculate the average running time, and the result is 855.7 ms, which shows that BreathPass can be used practically.

4.6 Related Works

Ultrasound Sensing: Ultrasound sensing has been a popular area of research in recent years. Ultrasound sensing is based on distance measurement and positioning. After getting the objects' trajectory, systems may employ classifiers or construct a model to detect the objects' actions. Numerous techniques have been proposed to localize or track objects, e.g., the technique based on time-of-arrival (ToA) or time-difference-of-arrival (TDoA) [107, 108, 109, 110, 111], Doppler frequency shift (DFS) [112, 113, 114], or phase-based technique [99, 115, 116]. In particular, Liu et al. [107] achieves ultrasound positioning accuracy of tens of centimeters using the time-of-arrival (ToA) technique. It modulates some signals emitted from several anchor nodes with known positions. When a microphone receives these signals, it first calculates the time difference between emitting and receiving, then calculates the distance between the microphone to each anchor. Finally, it uses trilateration to determine the microphone's position. LLAP [99] extracts the phase difference generated by the object's moving and achieves a 1-D tracking accuracy of 3.5 mm and a 2-D tracking accuracy of 4.6 mm. Vernier [115], on the other hand, leverages the vernier principle to improve tracking accuracy and achieves a 3D tracking error of less than 4 mm. Although all of these techniques claim to be inaudible to humans, pets and infants can hear ultrasound with a frequency close to that of audible sounds. Therefore, UPS+ [3] leverages the nonlinearity effects of commercial off-the-shelf (COTS) microphones that have been extensively studied by Backdoor [8] and proposed a new ultrasound positioning system. This system employs ultrasound at a frequency that is inaudible to pets and infants, making the ultrasound positioning system more environmentally friendly.

Additionally, ultrasound sensing systems can be used to complete a variety of sophisticated tasks. For example, existing research efforts [117, 118] employ ultrasound signals to detect sleep apnea. Specifically, these works emit modulated ultrasound, i.e., FMCW chirp or pseudo-white noise signal, and then use a classification algorithm to determine whether an apnea symptom exists. Moreover, SpiroSonic [10] uses reflected ultrasonic signals to detect whether the user's pulmonary function is normal. BreathListener [100] also uses reflected ultrasonic signals to

quantify the driver's breathing status, thereby determining whether or not the driver is driving safely. AcuTe [119] measures ambient temperature via ultrasonic sensing by utilizing the linear relationship between temperature and sound speed.

Wireless Authentication: Recent research efforts have primarily concentrated on device-to-device [120, 121, 122, 123] and human-to-device authentication [124, 125, 126, 81, 80, 127]. For example, in the case of device-to-device authentication, GeneWave [120] derives the initial acoustic channel response and creates a coding scheme for key agreement and exchange. DeMiCPU [123] finds that different CPUs generate distinct magnetic induction signal-based fingerprints, then designed a DeMiCPU sensor to read the fingerprint and classify it using ExtraTrees, thereby performing authentication.

As for the human-to-device authentication, Cardiac Scan [80] puts two radar sensors in front of and behind the user, respectively. During authentication, these two radar sensors first capture the user's cardiac motion, then extracts Fiducial-based invariant identity descriptors, and use them to match with the owner template captured in advance to perform authentication. It requires the target device to be equipped with radar sensors, which increases the cost of operation and precludes widespread use. Wang et al. [126] measure the heartbeat using the built-in accelerometer of smartphones, then extract features using wavelet transform and apply the SVM model to determine whether the captured heartbeat was from an authorized user. WiHF [127] uses a WiFi signal to deduce the user's identified gesture and then performs user identification using DNN. This method requires a WiFi connection and operates exclusively on the server side, making it unsuitable for client-side outdoor scenarios. BreathPrint [81] conducted a comprehensive and careful survey on existing clinical studies [128, 129, 130, 131, 132, 133] and summarizes that different people have distinct breathe sounds, then it designed a system that requires the user to initiate breathing gestures (e.g., sniff, normal, deep breath) towards a microphone. Then, it extracts GFCC features and uses a GMM model to determine whether or not the recorded breathing sounds originated from the authorized user. While this is a non-invasive solution, it requires users to place a microphone near their nose, which adds to the cost of operation. Meanwhile, during the COVID-19 pandemic,

people typically wear a face cover, which prevents the microphone from recording the sound of breathing.

4.7 Discussion and Future Work

Although BreathPass can become a candidate of "who you are" unlocking mechanism, it is still challenging to make it practical because of the following aspects, and we leave them as the future work:

Sampling position: In our experiment, all of the volunteers are strictly limited to the position where they put the mobile phone, i.e., 5 to 10 cm before and perpendicular to their chest wall. In practice, however, users could hold their smartphone at any distance and any orientation, which significantly affects the sampled breathing pattern. Therefore, to make BreathPass practical, we need to make the user hold their smartphone freely.

Lower FPR: The authentication system should provide an extremely low FPR in order to keep security. Although BreathPass achieves 5% FPR in general, it is still far from the authentication systems that can be used in practice. Getting a lower FPR while keeping a reasonable TPR before using BreathPass in practice is needed.

CHAPTER 5

CONCLUSION

This dissertation shows various unconventional ways to exploit the side effect of devices.

In Chapter 2, we present RainbowLight, a high-precision 3D visible light based localization system. Compared with existing approaches, RainbowLight does not require special hardware design and pre-collected light features. RainbowLight works on COTS mobile phones without strict user holding requirement. It works well for different types of lamps as well as light off scenario. Those features significantly reduce the deployment, maintenance and using overhead. The evaluation results show that RainbowLight achieves an average localization error of 3.3 cm in 2D and 9.6 cm in 3D. We believe RainbowLight can be applied to today's buildings with a very small overhead to enable many visible light based applications.

Acoustic privacy protection has always been an important topic. In Chapter 3, we study the nonlinear effects on commercial off-the-shelf microphones. Based on our study, we propose Patronus, which leverages the nonlinear effects to disrupt unauthorized devices from recording the speech while simultaneously allowing authorized devices to record clear speech audio. We implement and evaluate Patronus in a wide variety of representative scenarios. Results show that Patronus effectively blocks unauthorized devices from making secret recordings while allowing authorized devices to successfully make clear recordings.

In Chapter 4, we propose BreathPass, a novel biometric authentication method that is more resilience to replay-attack and has a high flexibility to mobile devices. It samples breathing patterns from users and extracts fingerprints from them to achieve authentication. We believe that BreathPass can become a candidate of "who you are" unlocking mechanism, or become complementary to another untrustable mechanism such as eye recognition to provide authentication service together, and with wearing different kinds of face covers, clothes, with different postures, dynamic status, and under different environments.

We believe with careful study and exploitation of side effects of devices, and carefully utilize

ultrasound signals, smart devices can sufficiently take advantage of their power and resource and achieve more powerful functionalities.

There are still many directions along with this dissertation and generatres multiple future works which are remained to be addressed. Regarding visible light positioning, existing works mostly focus on active positioning, i.e., the target either needs to be equipped with a camera, or needs to attach an anchor on it. How to achieve a passive visible light positioning with a low deploying and using cost is still a problem. Regarding ultrasounic privacy protection, because of the capacity of NLMS filter, the metrics of PESQ and SRVA are limited and still have a capacity to improve. Many research efforts employed artificial intelligence methods such as autoencoder, or generative adversarial network, to improve the performance of denoising, which could also be used for descrambling. In addition, many security architectures on the chipset such as Intel SGX, ARM TrustZone, or RISC-V KeyStone provide highly effective methods to protect memory physically. It opens a new door to enhance the security level of IoT devices and applications.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Chi Zhang and Xinyu Zhang. Litell: robust indoor localization using unmodified light fixtures. In *Proceedings of ACM MobiCom*, 2016.

[2] Shilin Zhu, Chi Zhang, and Xinyu Zhang. Automating visual privacy protection using a smart led. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 329–342, 2017.

[3] Qiongzheng Lin, Zhenlin An, and Lei Yang. Rebooting ultrasonic positioning systems for ultrasound-incapable smart devices. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[4] Zhou Zhou, Mohsen Kavehrad, and Peng Deng. Indoor positioning algorithm using light-emitting diode visible light communications. *Optical Engineering*, 51(8):085009, 2012.

[5] Bo Xie, Guang Tan, and Tian He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proceedings of ACM SenSys*, 2015.

[6] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of ACM MobiCom*, 2014.

[7] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117, 2017.

[8] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 2–14, 2017.

[9] Yitao He, Junyu Bian, Xinyu Tong, Zihui Qian, Wei Zhu, Xiaohua Tian, and Xinbing Wang. Canceling inaudible voice commands against voice control systems. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2019.

[10] Xingzhe Song, Boyuan Yang, Ge Yang, Ruirong Chen, Erick Forno, Wei Chen, and Wei Gao. Spirosonic: monitoring human lung function via acoustic sensing on commodity smartphones. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[11] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of ACM MobiCom*, 2014.

[12] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via rf body reflections. In *Proceedings of USENIX NSDI*, 2015.

[13] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 3d tracking via body radio reflections. In *Proceedings of USENIX NSDI*, 2014.

[14] Mei Wang, Zhehui Zhang, Xiaohua Tian, and Xinbing Wang. Temporal correlation of the rss improves accuracy of fingerprinting localization. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2016.

[15] Kun Qian, Chenshu Wu, Zheng Yang, Zimu Zhou, Xu Wang, and Yunhao Liu. Enabling phased array signal processing for mobile wifi devices. *IEEE Transactions on Mobile Computing*, 17(8):1820–1833, 2017.

[16] Chunhui Duan, Lei Yang, Qiongzheng Lin, and Yunhao Liu. Tagspin: High accuracy spatial calibration of rfid antennas via spinning tags. *IEEE Transactions on Mobile Computing*, 17(10):2438–2451, 2018.

[17] Yu-Lin Wei, Chang-Jung Huang, Hsin-Mu Tsai, and Kate Ching-Ju Lin. Celli: Indoor positioning using polarized sweeping light beams. In *Proceedings of ACM MobiSys*, 2017.

[18] Bo Xie, Kongyang Chen, Guang Tan, Mingming Lu, Yunhuai Liu, Jie Wu, and Tian He. Lips: A light intensity–based positioning system for indoor environments. *ACM Transactions on Sensor Networks*, 12(4):28, 2016.

[19] Radu Stoleru, Tian He, John A. Stankovic, and David Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *Proceedings of ACM SenSys*, 2005.

[20] Song Liu and Tian He. Smartlight: Light-weight 3d indoor localization using a single led lamp. In *Proceedings of ACM SenSys*, 2017.

[21] Nishkam Ravi and Liviu Iftode. Fiatlux: Fingerprinting rooms using light intensity. In *Proceedings of Pervasive*, 2007.

[22] Shilin Zhu and Xinyu Zhang. Enabling high-precision visible light localization in today's buildings. In *Proceedings of ACM MobiSys*, 2017.

[23] Qiang Xu, Rong Zheng, and Steve Hranilovic. Idyll: indoor localization using inertial and light sensors on smartphones. In *Proceedings of ACM Ubicomp*, 2015.

[24] Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of ACM MobiSys*, 2015.

[25] Masaki Yoshino, Shinichiro Haruyama, and Masao Nakagawa. High-accuracy positioning system using visible led lights and image sensor. In *Proceedings of IEEE RWS*, 2008.

[26] S-H Yang, E-M Jeong, D-R Kim, H-S Kim, Y-H Son, and S-K Han. Indoor three-dimensional location estimation based on led visible light communication. *Electronics Letters*, 49(1):54–56, 2013.

[27] Ruipeng Gao, Yang Tian, Fan Ye, Guojie Luo, Kaigui Bian, Yizhou Wang, Tao Wang, and Xiaoming Li. Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment. *IEEE Transactions on Mobile Computing*, 15(2):460–474, 2016.

[28] Chi Zhang and Xinyu Zhang. Pulsar: Towards ubiquitous visible light localization. In *Proceedings of ACM MobiCom*, 2017.

[29] Zhao Tian, Kevin Wright, and Xia Zhou. The darklight rises: Visible light communication in the dark. In *Proceedings of ACM MobiCom*, pages 2–15, 2016.

[30] Edward Collett. *Field guide to polarization*, volume 15. SPIE press Bellingham, 2005.

[31] WikiPedia. Birefringence. https://en.wikipedia.org/wiki/Birefringence.

[32] Wikipedia. Snell's law. https://en.wikipedia.org/wiki/Snell\%27s_law.

[33] SHEN Wei-min. Interference pattern of convergent light for a uniaxial crystal with optical axis parallel to surface. *College Physics*, 6:001, 2005.

[34] Dennis H Goldstein. *Polarized light*. CRC press, 2017.

[35] Wikipedia. Color wheel. https://en.wikipedia.org/wiki/Color_wheel\#Color_wheels_and_paint_color_mixing.

[36] Zhao Tian, Yu-Lin Wei, Wei-Nin Chang, Xi Xiong, Changxi Zheng, Hsin-Mu Tsai, Kate Ching-Ju Lin, and Xia Zhou. Augmenting indoor inertial tracking with polarized light. In *Proceedings of ACM MobiSys*, pages 362–375, 2018.

[37] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, Feng Zhao, Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, et al. Travi-navi: Self-deployable indoor navigation system. *IEEE/ACM Transactions on Networking (TON)*, 25(5):2655–2669, 2017.

[38] Jinsong Han, Chen Qian, Xing Wang, Dan Ma, Jizhong Zhao, Wei Xi, Zhiping Jiang, and Zhi Wang. Twins: Device-free object tracking using passive tags. *IEEE/ACM Transactions on Networking (TON)*, 24(3):1605–1617, 2016.

[39] Jizhong Zhao, Wei Xi, Yuan He, Yunhao Liu, Xiang-Yang Li, Lufeng Mo, and Zheng Yang. Localization of wireless sensor networks in the wild: Pursuit of ranging quality. *IEEE/ACM Transactions on Networking (ToN)*, 21(1):311–323, 2013.

[40] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, Fugui He, and Tianzhang Xing. Enabling contactless detection of moving humans with dynamic speeds using csi. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(2):52, 2018.

[41] Zuwei Yin, Chenshu Wu, Zheng Yang, and Yunhao Liu. Peer-to-peer indoor navigation using smartphones. *IEEE Journal on Selected Areas in Communications*, 35(5):1141–1153, 2017.

[42] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2. 0: Passive human tracking with a single wi-fi link. *Proceedings of ACM MobiSys*, 2018.

[43] Pat Pannuto, Benjamin Kempke, Li-Xuan Chuo, David Blaauw, and Prabal Dutta. Harmonium: Ultra wideband pulse generation with bandstitched recovery for fast, accurate, and robust indoor localization. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):11, 2018.

[44] Chunyi Peng, Guobin Shen, and Yongguang Zhang. Beepbeep: A high-accuracy acoustic-based system for ranging and localization using cots devices. *ACM Transactions on Embedded Computing Systems*, 11(1):4:1–4:29, 2012.

[45] K. Liu, X. Liu, L. Xie, and X. Li. Towards accurate acoustic localization on a smartphone. In *Proceedings of IEEE INFOCOM*, 2013.

[46] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained smartphone localization via acoustic anchors. *IEEE Transactions on Mobile Computing*, 15(5):1144–1156, 2016.

[47] Pengfei Zhou, Yuanqing Zheng, and Mo Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of ACM MobiSys*, 2014.

[48] Yin Chen, Jie Liu, Dimitrios Lymberopoulos, and Bodhi and Priyantha. Fm-based indoor localization. In *Proceedings of ACM MobiSys*, 2012.

[49] Yonghang Jiang, Zhenjiang Li, and Jianping Wang. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. *IEEE Transactions on Mobile Computing*, 2018.

[50] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, and Guojie Luo. Smartphone-based real time vehicle tracking in indoor parking structures. *IEEE Transactions on Mobile Computing*, 16(7):2023–2036, 2017.

[51] The Guardian. Apple apologises for allowing workers to listen to siri recordings. https://www.theguardian.com/technology/2019/aug/29/apple-apologises-listen-siri-recordings. (Accessed on Feb. 28, 2020).

[52] CNBC. Amazon echo recorded conversation, sent to random person: report. https://www.cnbc.com/2018/05/24/amazon-echo-recorded-conversation-sent-to-random-person-report.html. (Accessed on Feb. 28, 2020).

[53] The Guardian. Ukraine prime minister offers resignation after leaked recording. https://www.theguardian.com/world/2020/jan/17/ukraine-prime-minister-oleksiy-goncharuk-offers-resignation-after-leaked-recording. (Accessed on Feb. 28, 2020).

[54] Yu-Chih Tung and Kang G. Shin. Exploiting sound masking for audio privacy in smartphones. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, page 257–268, 2019.

[55] Anti-eavesdropping and recording blocker device, China Patent 201320228440, Oct. 2013.

[56] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible voice commands: The long-range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 547–560, 2018.

[57] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[58] Xinyan Zhou, Xiaoyu Ji, Chen Yan, Jiangyi Deng, and Wenyuan Xu. Nauth: Secure face-to-face device authentication via nonlinearity. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2080–2088. IEEE, 2019.

[59] Qiben Yan, Kehai Liu, Qin Zhou, Hanqing Guo, and Ning Zhang. Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided wave. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[60] Aleksandr Rovner. The principle of ultrasound. https://www.echopedia.org/wiki/The_principle_of_ultrasound, 2015.

[61] Ali H Sayed. *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[62] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.

[63] Anran Wang, Chunyi Peng, Ouyang Zhang, Guobin Shen, and Bing Zeng. Inframe: Multiflexing full-frame visible communication channel for humans and devices. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2014.

[64] Anran Wang, Zhuoran Li, Chunyi Peng, Guobin Shen, Gan Fang, and Bing Zeng. Inframe++ achieve simultaneous screen-human viewing and hidden screen-camera communication. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 181–195, 2015.

[65] Viet Nguyen, Yaqin Tang, Ashwin Ashok, Marco Gruteser, Kristin Dana, Wenjun Hu, Eric Wengrowski, and Narayan Mandayam. High-rate flicker-free screen-camera communication with spatially adaptive embedding. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[66] Kai Zhang, Yi Zhao, Chenshu Wu, Chaofan Yang, Kehong Huang, Chunyi Peng, Yunhao Liu, and Zheng Yang. Chromacode: A fully imperceptible screen-camera communication system. *IEEE Transactions on Mobile Computing*, 2019.

[67] Qian Wang, Kui Ren, Man Zhou, Tao Lei, Dimitrios Koutsonikolas, and Lu Su. Messages behind the sound: real-time hidden acoustic signal capture with smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 29–41, 2016.

[68] Man Zhou, Qian Wang, Kui Ren, Dimitrios Koutsonikolas, Lu Su, and Yanjiao Chen. Dolphin: Real-time hidden acoustic signal capture with smartphones. *IEEE Transactions on Mobile Computing*, 18(3):560–573, 2018.

[69] Lan Zhang, Cheng Bo, Jiahui Hou, Xiang-Yang Li, Yu Wang, Kebin Liu, and Yunhao Liu. Kaleido: You can watch it but cannot record it. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 372–385, 2015.

[70] Ingo R Titze and Daniel W Martin. Principles of voice production, 1998.

[71] Ronald J Baken and Robert F Orlikoff. *Clinical measurement of speech and voice*. Cengage Learning, 2000.

[72] Sheng Shen, Nirupam Roy, Junfeng Guan, Haitham Hassanieh, and Romit Roy Choudhury. Mute: bringing iot to noise cancellation. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 282–296, 2018.

[73] ITUT Rec. P. 800.1, mean opinion score (mos) terminology. *International Telecommunication Union, Geneva*, 2006.

[74] Mika Wilson. Pesq - what is it and how could it transform your customer experience? https://www.spearline.com/blog/post/pesq---what-is-it-and-how-could-it-transform-your-customer-experience-/, 2018. (Accessed on Oct. 2, 2020).

[75] Kamil Wojcicki. Pesq matlab wrapper. https://www.mathworks.com/matlabcentral/fileexchange/33820-pesq-matlab-wrapper. (Accessed on Mar. 6, 2020).

[76] About Face ID advanced technology - Apple Support. https://support.apple.com/en-us/HT208108. (Accessed on Nov. 04, 2021).

[77] In-screen fingerprint sensors coming to 100 million phones by 2019? - cnet. https://www.cnet.com/tech/mobile/in-screen-fingerprint-sensors-coming-to-100-million-phones-by-2019-report/. (Accessed on Nov. 04, 2021).

[78] Zia Saquib, Nirmala Salam, Rekha Nair, and Nipun Pandey. Voiceprint recognition systems for remote authentication-a survey. *International Journal of Hybrid Information Technology*, 4(2):79–97, 2011.

[79] John Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1167–1175, 2007.

[80] Feng Lin, Chen Song, Yan Zhuang, Wenyao Xu, Changzhi Li, and Kui Ren. Cardiac scan: A non-contact and continuous heart-based user authentication system. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 315–328, 2017.

[81] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. Breathprint: Breathing acoustics-based user authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 278–291, 2017.

[82] Yinghui Li, Zhichao Cao, and Jiliang Wang. Gazture: Design and implementation of a gaze based gesture control system on tablets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–17, 2017.

[83] Hongbo Jiang, Hangcheng Cao, Daibo Liu, Jie Xiong, and Zhichao Cao. Smileauth: Using dental edge biometrics for user authentication on smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–24, 2020.

[84] Yongchao Ye, Lingjie Lao, Diqun Yan, and Lang Lin. Detection of replay attack based on normalized constant q cepstral feature. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 407–411. IEEE, 2019.

[85] S Saranya, Suvidha Rupesh Kumar, and B Bharathi. Deep learning approach: detection of replay attack in asv systems. In *International Conference on Soft Computing and Signal Processing*, pages 291–298. Springer, 2019.

[86] Bin Hao, Xiali Hei, Yazhou Tu, Xiaojiang Du, and Jie Wu. Voiceprint-based access control for wireless insulin pump systems. In *2018 IEEE 15th international conference on mobile ad hoc and sensor systems (MASS)*, pages 245–253. IEEE, 2018.

[87] Miroslav Goljan, Jessica Fridrich, and Mo Chen. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security*, 6(1):227–236, 2010.

[88] Roberto Caldelli, Irene Amerini, and Andrea Novi. An analysis on attacker actions in fingerprint-copy attack in source camera identification. In *2011 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2011.

[89] Robert W Frischholz and Alexander Werner. Avoiding replay-attacks in a face recognition system using head-pose estimation. In *2003 IEEE International SOI Conference. Proceedings (Cat. No. 03CH37443)*, pages 234–235. IEEE, 2003.

[90] Gang Pan, Zhaohui Wu, and Lin Sun. Liveness detection for face recognition. *Recent advances in face recognition*, pages 109–124, 2008.

[91] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016.

[92] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[93] Xiaojia Zhao, Yang Shao, and DeLiang Wang. Casa-based robust speaker identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1608–1616, 2012.

[94] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

[95] Hideo Kaneko and Jun Horie. Breathing movements of the chest and abdominal wall in healthy subjects. *Respiratory care*, 57(9):1442–1451, 2012.

[96] Maria Ragnarsdóttir and Ella Kolbrun Kristinsdóttir. Breathing movements and breathing patterns among healthy men and women 20–69 years of age. *Respiration*, 73(1):48–54, 2006.

[97] Pablo Martinez-Lozano Sinues, Malcolm Kohler, and Renato Zenobi. Human breath analysis may support the existence of individual metabolic phenotypes. *PloS one*, 8(4):e59909, 2013.

[98] JERE Mead and STEPHEN H Loring. Analysis of volume displacement and length changes of the diaphragm during breathing. *Journal of Applied Physiology*, 53(3):750–755, 1982.

[99] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 82–94, 2016.

[100] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 54–66, 2019.

[101] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[102] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[103] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[104] Android. Aaudio library. https://developer.android.com/ndk/guides/audio/aaudio/aaudio. (Accessed on Nov. 02, 2021).

[105] Apache. Commons math: The apache commons mathematics library. https://commons.apache.org/proper/commons-math/. (Accessed on Nov. 02, 2021).

[106] PyTorch. Torchscript. https://pytorch.org/docs/stable/jit.html. (Accessed on Nov. 02, 2021).

[107] Kaikai Liu, Xinxin Liu, Lulu Xie, and Xiaolin Li. Towards accurate acoustic localization on a smartphone. In *2013 Proceedings IEEE INFOCOM*, pages 495–499. IEEE, 2013.

[108] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Acoustic ranging and communication via microphone channel. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 291–296. IEEE, 2012.

[109] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Guoguo: Enabling fine-grained smartphone localization via acoustic anchors. *IEEE transactions on mobile computing*, 15(5):1144–1156, 2015.

[110] Chunyi Peng, Guobin Shen, and Yongguang Zhang. Beepbeep: A high-accuracy acoustic-based system for ranging and localization using cots devices. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(1):1–29, 2012.

[111] Hyosu Kim, Anish Byanjankar, Yunxin Liu, Yuanchao Shu, and Insik Shin. Ubitap: Leveraging acoustic dispersion for ubiquitous touch interface on solid surfaces. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 211–223, 2018.

[112] Joseph Paradiso, Craig Abler, Kai-yuh Hsiao, and Matthew Reynolds. The magic carpet: physical sensing for immersive environments. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems*, pages 277–278. 1997.

[113] Kaustubh Kalgaonkar and Bhiksha Raj. One-handed gesture recognition using ultrasonic doppler sonar. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1889–1892. IEEE, 2009.

[114] Stephen P Tarzia, Robert P Dick, Peter A Dinda, and Gokhan Memik. Sonar-based measurement of user presence and attention. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 89–92, 2009.

[115] Yunhao Liu, Jiliang Wang, Yunting Zhang, Linsong Cheng, Weiyi Wang, Zhao Wang, Weimin Xu, and Zhenjiang Li. Vernier: Accurate and fast acoustic motion tracking using mobile devices. *IEEE Transactions on Mobile Computing*, 2019.

[116] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 591–605, 2018.

[117] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proceedings of ACM MobiSys*, pages 45–57, 2015.

[118] Anran Wang, Jacob E Sunshine, and Shyamnath Gollakota. Contactless infant monitoring using white noise. In *Proceedings of ACM MobiCom*, pages 1–16, 2019.

[119] Chao Cai, Zhe Chen, Henglin Pu, Liyuan Ye, Menglan Hu, and Jun Luo. Acute: acoustic thermometer empowered by a single smartphone. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 28–41, 2020.

[120] Pengjin Xie, Jingchao Feng, Zhichao Cao, and Jiliang Wang. Genewave: Fast authentication and key agreement on commodity mobile devices. *IEEE/ACM Transactions on Networking*, 26(4):1688–1700, 2018.

[121] Hongbo Liu, Yang Wang, Jie Yang, and Yingying Chen. Fast and practical secret key extraction by exploiting channel response. In *2013 Proceedings IEEE INFOCOM*, pages 3048–3056. IEEE, 2013.

[122] Jinsong Han, Chen Qian, Panlong Yang, Dan Ma, Zhiping Jiang, Wei Xi, and Jizhong Zhao. Geneprint: Generic and accurate physical-layer identification for uhf rfid tags. *IEEE/ACM Transactions on Networking*, 24(2):846–858, 2015.

[123] Yushi Cheng, Xiaoyu Ji, Juchuan Zhang, Wenyuan Xu, and Yi-Chao Chen. Demicpu: Device fingerprinting with magnetic signals radiated by cpu. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1149–1170, 2019.

[124] Kevin R Farrell, Richard J Mammone, and Khaled T Assaleh. Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on speech and audio processing*, 2(1):194–205, 1994.

[125] Michael Schmidt and Herbert Gish. Speaker identification via support vector classifiers. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 105–108. IEEE, 1996.

[126] Lei Wang, Kang Huang, Ke Sun, Wei Wang, Chen Tian, Lei Xie, and Qing Gu. Unlock with your heart: Heartbeat-based authentication on commercial mobile phones. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2(3):1–22, 2018.

[127] Chenning Li Li, Manni Liu, and Zhichao Cao. Wihf: Gesture and user recognition with wifi. *IEEE Transactions on Mobile Computing*, 2020.

[128] Volker Gross, Anke Dittmar, Thomas Penzel, Frank Schuttler, and Peter Von Wichert. The relationship between normal lung sounds, age, and gender. *American journal of respiratory and critical care medicine*, 162(3):905–909, 2000.

[129] Hans Pasterkamp, Steve S Kraman, and George R Wodicka. Respiratory sounds: advances beyond the stethoscope. *American journal of respiratory and critical care medicine*, 156(3):974–987, 1997.

[130] Hans Pasterkamp, Richard E Powell, and Ignacio Sanchez. Lung sound spectra at standardized air flow in normal infants, children, and adults. *American journal of respiratory and critical care medicine*, 154(2):424–430, 1996.

[131] Hans Pasterkamp, Jürgen Schäfer, and George R Wodicka. Posture-dependent change of tracheal sounds at standardized flows in patients with obstructive sleep apnea. *Chest*, 110(6):1493–1498, 1996.

[132] Ignacio Sanchez and Hans Pasterkamp. Tracheal sound spectra depend on body height. *American Review of Respiratory Disease*, 148:1083–1083, 1993.

[133] JT Sharp, JP Henry, SK Sweany, WR Meadows, RJ Pietras, et al. The total work of breathing in normal and obese men. *The Journal of clinical investigation*, 43(4):728–739, 1964.