INAUDIBLE ACOUSTICS: TECHNIQUES AND APPLICATIONS

BY

NIRUPAM ROY

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Romit Roy Choudhury, Chair
Assistant Professor Haitham Al-Hassanieh
Professor Klara Nahrstedt
Dr. Venkat Padmanabhan, Microsoft Research India
Professor Nitin Vaidya

Abstract

This dissertation is focused on developing a sub-area of acoustics that we call inaudible acoustics. We have developed two core capabilities, (1) *BackDoor* and (2) *Ripple*, and demonstrated their use in various mobile and IoT applications. In *BackDoor*, we synthesize ultrasound signals that are inaudible to humans yet naturally recordable by all microphones. Importantly, the microphone does not require any modification, enabling billions of microphone-enabled devices, including phones, laptops, voice assistants, and IoT devices, to leverage the capability. Example applications include acoustic data beacons, acoustic watermarking, and spy-microphone jamming. In *Ripple*, we develop modulation and sensing techniques for vibratory signals that traverse through solid surfaces, enabling a new form of secure proximal communication. Applications of the vibratory communication system include on-body communication through imperceptible physical vibrations and device-device secure data transfer through physical contacts. Our prototypes include an inaudible jammer that secures private conversations from electronic eavesdropping, acoustic beacons for location-based information sharing, and vibratory communication in a smart-ring sending password through a finger touch. Our research also uncovers new security threats to acoustic devices. While simple abuse of inaudible jammer can disable hearing aids and cell phones, our work shows that voice interfaces, such as Amazon Echo, Google Home, Siri, etc., can be compromised through carefully designed inaudible voice commands. The contributions of this dissertation can be summarized in three primitives: (1) exploiting inherent hardware nonlinearity for sensing out-of-band signals, (2) developing the vibratory communication system for secure touch-based data exchange, and (3) structured information reconstruction from noisy acoustic signals. In developing these primitives, we draw from principles in wireless networking, digital communications, signal processing, and embedded design and translate them to completely functional systems.

To Tarun-Sikha, my eternal sunshine.

Acknowledgments

I enjoyed every bit of my time at UIUC as a Ph.D. student. It was as much a journey inside through contemplation as it was a pursuit of knowledge through hypotheses and experiments. It made me understand that the purpose of education is not merely filling the mind with facts, but also perfecting its ability to see through illusions.

I owe my deepest gratitude to Prof. Romit Roy Choudhury, my advisor. He inspired my soul and enriched my mind with his philosophies. I delighted in discussing research with him and sharing his passion for work and commitment to perfection. I am forever indebted to him for all his sleepless nights in the lab, the battles he fought for me, and the effort he expended every day for the last five years to make me what I am today. Thank you, Romit, for being a great advisor and a wonderful person. You are my academic hero.

I am also grateful to my Ph.D. committee members and mentors – Prof. Haitham Hassanieh, Prof. Klara Nahrstedt, Dr. Venkat Padmanabhan, and Prof. Nitin Vaidya. Working with Haitham on the BackDoor project was a great learning experience because of his excellence in research and humility in life. I want to thank Klara for all her suggestions and kind support and Venkat for our discussions during the Ripple project. His vision of making societal impacts through technology will always inspire me. Nitin always amazed me by his attention to detail, which I have tried to emulate.

My research was also greatly improved by the faculty panel at the Coordinated Science Lab (CSL), UIUC. The "Feedback Friday" initiative gave me and many other students an opportunity to present our research to and receive constructive advice from a diverse panel of scholars. My special thanks to Prof. Rakesh Kumar, Prof. Jian Huang, and Prof. Nikita Borisov, who helped me organize my research and present it to a broad audience. I was fortunate, too, to work with Prof. Pramod Viswanath and my friend Shaileshh for the Qualcomm Innovation Fellowship program. I thank them for their help and patience.

No one's career begins with a Ph.D. program. My master's advisor, Prof. Srihari Nelakuditi, was, above all, a caring human being. I cherish the countless hours we spent discussing research. Thank you, Srihari. Your ability to remain happy under all circumstances was contagious. I am also thankful to my undergraduate research advisor, Prof. Abhik Mukherjee, who introduced me to the joy of innovation through our robotic navigation project and to my high school teachers at Midnapore Collegiate School, especially to Mr. Dipankar Sannigrahi, for believing in me.

Success, of course, depends not only on mentors but also on friends.

My labmates at SyNRG were supportive and fun. Papers co-authored with Mahanth, Sheng, and He enriched this dissertation in many ways. Ashutosh and Puneet always had time to discuss my research and give suggestions. Our adventures outside of work added joy to my life, especially the canoe trip where I found new respect for life jackets. My special thanks to Carol for her administrative support and for being such a loving person. I am a big fan of her baking, especially her carrot cakes. I am also thankful to my friends at CSL – Subho, Arjun, Yoga, Saurabh, Saboo, and Chuchu – for their cheerful company. We played cricket in the snow and celebrated each other's festivals.

I am among those fortunate people who still share drinks with their childhood friends. Washim, Debsundar, and Santanu have always been there for me to cheer me up and celebrate my every little success. I found a brother in Aritra, who stood by me during all my struggles and triumphs. I am also blessed to have friends like Sadik, Sumana, Sujoy, and Souvik.

Finally, a special note of gratitude goes to my family, without whom I would not be the man I have become.

First, I thank my awesome wife and my dearest friend, Sanorita. It has been 15 years since we first met and we have seen many ups and downs together. I have always found her by my side whenever I reached out. It is an immense pleasure by itself to know such a cheerful and optimistic person. Thank you, Sanorita and your parents, Mahendra and Shila, for your unwavering support.

Ever since I was a child, I have looked up to my older brother, Anupam. He shared his desserts with me and took punishments on my behalf. I could not have wished for a more supportive and loving brother. His pledge to promote experimental physics among underprivileged students is a constant source of motivation for me.

I spent a lot of my childhood time with my uncles, Dharmadas and Biswanath. I am grateful to Dharmadas for introducing me to drawing and poetry and Biswanath for introducing me to music. A part of me would have remained unopened without them.

Finally, my deepest gratitude and love to my mother, Sikha, and my father, Tarun. Whenever I try to write this part of the note, a host of memories and emotions rush to my mind and words fall short. I remember their faces during my first stage performance, when I learned to ride a bike, before leaving for my first road trip alone, and on the day of my graduation. Thousands of such tiny moments of their inspiration, support, sacrifices, and love have become the foundation of my life. Thank you, Ma, and thank you, Baba, for all that you are and for everything that you have done for me. This dissertation is dedicated to you.

Table of Contents

Chapter 1

Introduction

This dissertation is focused on developing a sub-area of acoustics that we call *inaudible acoustics*. We have developed two core capabilities and demonstrated their use in various mobile and IoT applications. Briefly, the two capabilities are as follows.

(1) Project *BackDoor*: We design ultrasound signals that are naturally inaudible to humans, yet completely recordable by all microphones without any hardware or software changes to them. This implies that any microphone will be able to decode transmitted information – data bits or voice commands – even though these transmissions are non-disturbing to human society. We show how such capabilities translate to new applications, both useful and malicious. Examples include inaudible data communication in the acoustic band, acoustic watermarking, spy-microphone jamming, and inaudible voice attacks on home devices like Amazon Echo and Google Home.

(2) Project *Ripple*: We develop a vibratory communication stack that transmits and receives vibration signals through solid surfaces, enabling a new form of secure proximal communication. The transmitters for such signals are widely practical since any mobile device's vibration motor can be carefully programmed to modulate data bits. As receivers, we show how accelerometer sensors, or even microphones, can be leveraged to sense and demodulate the data bits. Thus, tomorrow's smartwatches may have "vibrational radios" embedded in them, allowing two humans to exchange information by shaking hands.

We briefly sketch these two parts of the dissertation next, followed by how the work generalizes to beyond today's applications.

1.1 Techniques in Inaudible Acoustics

(1) Project *BackDoor*: Consider sounds, say at frequencies above 40 kHz, that are completely outside the human's audible range (20 kHz), as well as a microphone's recordable range (24 kHz). Our work shows that these high-frequency sounds can be designed to become recordable by unmodified microphones while remaining inaudible to humans. We call this system *BackDoor*. The core idea lies in exploiting fundamental nonlinearities in microphone hardware. Briefly, we design ultrasound signals and play it on a speaker such that, after passing through the microphone's nonlinear diaphragm and power-amplifier, the signal creates a signal component in the audible frequency range. The component can be regulated to carry data bits, thereby enabling an acoustic (but inaudible) communication channel to today's microphone-enabled devices, such as smartphones, smartwatches, laptop, hearing aids, and voice assistants (Amazon Echo, Google Home).

While nonlinearity presents opportunities, designing ultrasound for a nonlinear channel is challenging. Regular modulation schemes often result in distorted signals at the receiver as spurious frequencies overlap the target baseband signals and standard equalization schemes are not effective. Most importantly the ultrasonic transmitters – a speaker with a diaphragm – also exhibit nonlinearity similar to the microphones. After interacting with the speaker nonlinearity, the inaudible ultrasonic signals leak audible sounds. We overcome these challenges in *BackDoor* with a custom-made speaker system implementing a novel signal shaping technique. Our core idea is to use multiple speakers, and divide segments of the input spectrum across them such that leakage from each speaker is narrow band, and confined to low frequencies. These segments, however, align perfectly at the receiving microphone to reconstruct the target signal. This still does not fully solve the problem and produces a garbled, audible sound at the transmitter. To achieve true inaudibility, we solve a min-max optimization problem on the length of the input segments. The optimization picks the segment lengths in a way such that the aggregate leakage function is completely below the human auditory response curve (i.e., the minimum separation between the leakage and the human audibility curve is maximized). This ensures, by design, the *BackDoor* signal is inaudible.

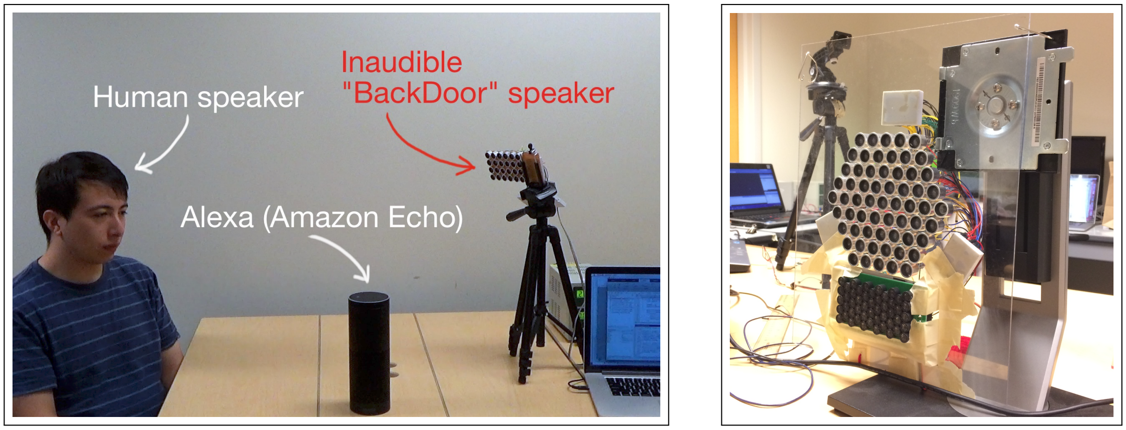
(2) Project *Ripple*: Smartphones and smartwatches contain a tiny actuator called vibration motor that generates the haptic alert when the device is on silent mode. Just like diaphragms of speakers oscillate to create sound waves in the air, a movable mass in the vibration motor generates vibration waves that traverse through solids. We explore this surface vibration as a modality for data communication. We have shown that it is possible to programmatically control the motion of this vibration motor to generate a band of vibration signals carrying data bits. The motion sensors in smart devices, like accelerometers, can sense the vibration signal and receive the data bits. Due to the limited bandwidth of accelerometers and inertia of the actuator, such vibratory communication system is fundamentally limited to around 10 bits per second of data rate. However, our vibratory communication system, called *Ripple*, overcomes these limitations to achieved 32 kbps of data rate.

The major enhancement in data rate was achieved by replacing the accelerometer with the microphone as the receiver. While microphones are designed to receive in-air vibrations of sound, we have shown that its diaphragm responds to physical vibrations as well. We develop an adaptive filter-based sensing technique that cancels noise from ambient sounds to recover subtle vibration signals from the microphone. We design an entire communication stack over these vibratory signals, including an OFDM-based physical layer and a link layer that detects collisions at the transmitter and performs proactive symbol retransmissions. We build fully functional prototypes of *Ripple* for on-body and device-device communication applications.

1.2 Applications of Inaudible Acoustics

(1) Inaudible Acoustic Communication: *BackDoor* essentially enables inaudible signals to be recordable by any regular microphone. We implement this technique in an acoustic beacon that transmits data bits at ultrasound frequencies above 40 kHz, but these signals can be received by any microphone-enabled device, like a smartwatch or a smartphone. Existing ultrasound-based communication systems [1] suffer from limited bandwidth, around 3 kHz, since they must remain above human hearing range (20 kHz) and below the microphone's cutoff frequency (24 kHz). Our inaudible beacon is free of these limitations. Using an ultrasound-based transmitter, it can utilize the entire microphone spectrum for communication. Thus, IoT devices could find an alternative channel for communication, reducing the growing load on Bluetooth (BLE). Museums and shopping malls could use acoustic beacons to broadcast information about nearby art pieces or products. Various ultrasound ranging schemes, that compute time of flight of signals, could benefit from the substantially higher bandwidth of *BackDoor* beacons.

(2) Inaudible Jammer for Acoustic Privacy: Hardware miniaturization of microphones has revolutionized mobile acoustic devices. As an unfortunate consequence, acoustic eavesdropping has become easy and unobtrusive. Apart from portable sound recorders, every smartphone is a potential sound recording device. Hidden miniaturized microphones can record conversations of a sensitive meeting. An apparently disinterested bystander can capture a conversation using the smartphone without even taking it out of the pocket. In a different scenario, a live performance can be recorded with a portable microphone and published without permission. Preventing acoustic eavesdropping is difficult as human ears and microphones operate at same frequency band. An attempt to jam microphones ends up playing loud sounds that disrupt the conversation itself. We build on the *BackDoor* technique to design an inaudible acoustic jammer that can block all microphones in a region by sending jamming signals. The prototype can silently jam spy microphones from recording, leading to a widespread application in acoustic privacy. Military and government officials can secure private and confidential meetings from electronic eavesdropping; cinemas and concerts can prevent unauthorized recording of movies and live performances.



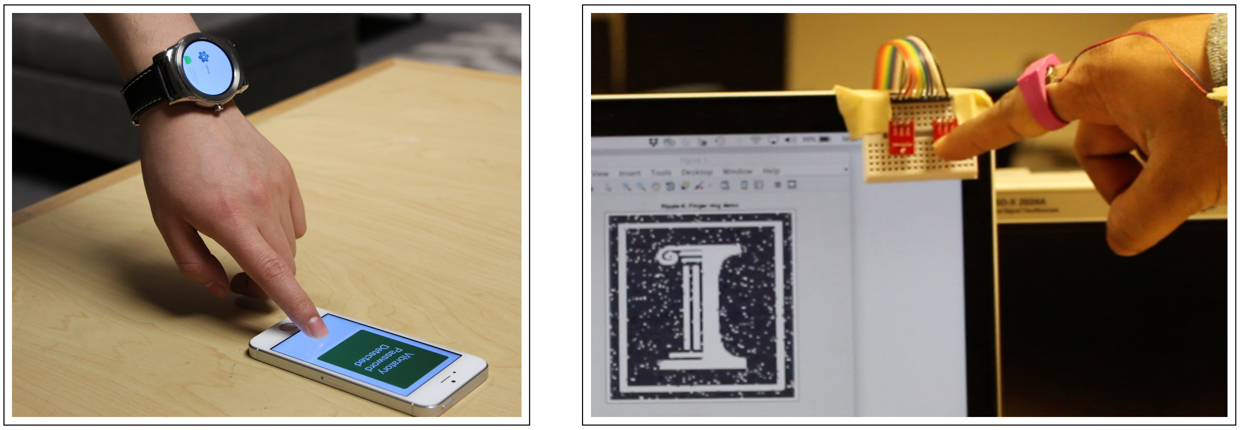
(a)

(b)

Figure 1.1: **Making microphones hear inaudible sounds:** We design ultrasound such that it becomes recordable after passing through the regular microphone hardware, while still remains inaudible to humans. This technique, called *BackDoor*, has applications in acoustic communication and jamming. However, *BackDoor* signals can be exploited to launch attacks on sound- and voice-activated devices. (a) A setup demonstrating inaudible voice command attack on Amazon Echo, an off-the-shelf voice assistant. (b) The ultrasound speaker system for long-range applications. This speaker system implements a novel signal shaping technique to avoid audible leakage. The video demonstration of the *BackDoor* system is available at: https://www.youtube.com/watch?v=_FrKySibcb8.

(3) Inaudible Voice Command Attack: Our research on inaudible acoustics also uncovers new kinds of threats to acoustic devices. Denial-of-service (DoS) attacks on sound devices are typically considered difficult as the jammer can be easily detected. However, *BackDoor* shows that inaudible jammers can disable hearing aids and cellphones without getting detected. For example, during a robbery, the perpetrators can prevent people from making 911 calls by silently jamming all phones' microphones. Inaudible voice commands launched from outside of the house can attack voice assistants, like Amazon Echo and Google Home, without raising an alarm to the people inside the house. To demonstrate the attack, we develop a working prototype of a long-range inaudible speaker system capable of sending inaudible voice signals to any voice-activated devices (Figure 1.1). This system requires an alternative transmitter design that can maintain inaudibility for this high power long range speaker system, that otherwise leaks a low-frequency noise.

(4) Protecting Voice Interfaces from Inaudible Attacks: Defending against this class of nonlinearity based inaudible attacks is not difficult if one were to assume hardware changes to the receiver (e.g., Amazon Echo or Google Home). An additional ultrasound microphone will suffice since it can detect the transmitted ultrasound signals in the air. However, with software changes alone, the problem becomes a question of forensics, i.e., can the received low-frequency signal, which is shifted due to the hardware nonlinearity, be discriminated from the same legitimate voice command transmitted with ultrasound? Our defense relies on the observation that voice signals exhibit a well-understood structure, composed of fundamental frequencies and harmonics. When this structure passes through nonlinearity, part of it remains preserved in the shifted and blended low-frequency signals. In contrast, legitimate human voice projects almost no energy in these low-frequency bands. We locate such indelible traces of nonlinearity in the received signal to identify an attack voice from legitimate commands.



(a) (b)

Figure 1.2: **Communicating through physical vibrations:** We develop a short-range communication modality using modulated vibrations in solids. This technique, called *Ripple*, enables secure data exchange between devices through physical contacts. Imperceptible vibration signals can carry data bits through the human body for touch-based communication with IoT devices. (a) An on-body vibratory communication prototype showing data transmission from a smartwatch. (b) Sending password from a smart-ring using vibrations that humans cannot feel. The video demonstration of the *Ripple* system is available at: https://synrg.csl.illinois.edu/ripple/.

(5) Communication through Touch: Unlike radio frequencies (RF), vibration does not broadcast the signal in the air. The vibratory communication, *Ripple*, requires physical contacts for the signal to traverse from the transmitter to the receiver, making it harder to eavesdrop. We leverage this property to develop secure applications like touch-based cryptographic key exchange and seamless authentication through touch. We build two prototypes on *Ripple*: (1) wearables (a finger ring and a smartwatch) that transmits vibratory passwords through the finger bone to enable touch-based authentication (Figure 1.2). The user can open a car door by simply touching the handle or activate a two-factor authentication for money transfer through a finger touch. (2) The second prototype is a surface communication between devices which are in physical contact or placed on the same table. Users can exchange contact details by touching each other's smartphones or start a one-to-many file transfer by placing them on the table.

1.3 Opportunities Beyond this Dissertation

This dissertation explores ultrasound and vibration for applications in inaudible acoustics. Principles developed in this work can be generalized for a larger application domain and can seed novel research directions. In this section, we outline a few of such future research opportunities.

Linearity is often a partial approximation of the real-world behavior, embraced for the purposes of simplicity. However, advancement in computing technologies empowers the next generation systems to explore and leverage the nonlinear behavior of signals. Out-of-band ultrasound sensing, described in Chapters 2 and 3, builds on the nonlinearities of acoustic hardware, however, the principles may offer benefits in other dimensions as well. For example, there is evidence [2] that air turns into a nonlinear medium for high-frequency sound. With proper signal design along with beam-forming techniques, ultrasonic sound can interact with air nonlinearity and create virtual signal sources. We can build on this technology to project sound to a specific location in space like a sound pocket in the air, leading to applications in device free virtual reality for gaming, navigation, and communication. Nonlinearity in liquid and other objects can also lead to new opportunities in acoustic imaging, material identification, and fingerprinting. Nonlinear mixing of signals generates baseband components, which can lead to simplified receiver architecture for communication, active sonar based motion tracking, and even new forms of acoustic localization. We are exploring some of these novel and powerful ideas around nonlinear sensing toward cross-disciplinary systems.

The imperceptible vibration-based signaling method can be further explored in diverse scenarios. Intra-body vibratory communication through bone conduction can lead to new applications in implantable medical devices. On the other hand, the primitive for speech recovery from vibration, described in Chapter 6, can be the key technique in a better speech analysis system. Vibrations induced in facial bones and muscles can serve as an alternative channel for speech. Although such vibrations carry only a partial information compared to the airborne sound channel, it is immune to the ambient noise. A careful combination of vibration and sound can lead to a robust speech recognition system for noisy environments and also an assistive system for people with speech disabilities.

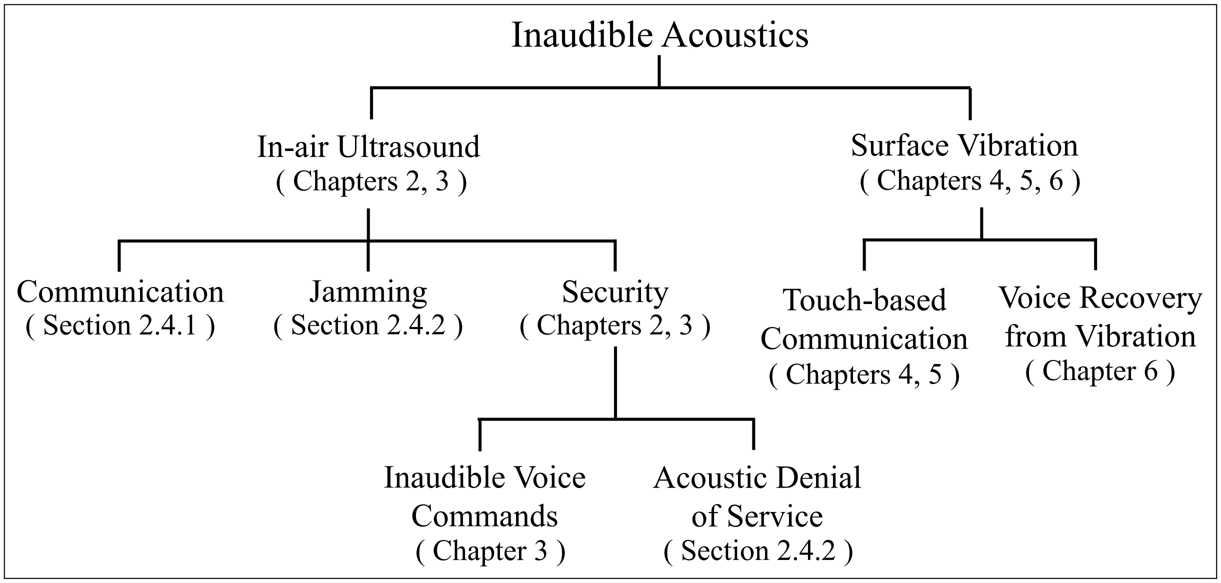


Figure 1.3: Organization of the topics in chapters.

1.4 Organization

The subsequent sections elaborate on these ideas of inaudible acoustics, starting with the basic concept of out-of-band signal sensing through sensor nonlinearity in Chapter 2. Chapter 3 extends this primitive to a long-range inaudible acoustic system. It also analyzes the security of voice interfaces against inaudible voice attack and presents a defense mechanism. Chapters 4 and 5 explain the vibratory communication system starting from basic vibration modulation techniques to a functional prototype of a touch-based data exchange system. Algorithms for voice signal reconstruction from vibration are presented in Chapter 6. Finally, we conclude in Chapter 7. Figure 1.3 shows the organization of the topics in the rest of this dissertation.

Chapter 2

Out-of-Band Inaudible Sound Sensing

2.1 Overview

This chapter shows the possibility of creating sounds that humans cannot hear but microphones can record. This is not because the sound is too soft or just at the periphery of human's frequency range. The sounds we create are actually 40 kHz and above, completely outside both human's and microphone's range of operation. However, given microphones possess inherent nonlinearities in their diaphragms and power amplifiers, it is possible to design sounds that exploit this property. To elaborate, we shape the frequency and phase of sound signals and play them through ultrasound speakers; when these sounds pass through the nonlinear hardware at the receiver, the high-frequency sounds are expected to create a low-frequency "shadow". The "shadow" is within the filtering range of the microphone and thereby gets recorded as normal sounds. Figure 2.1 illustrates the effect. Importantly, the microphone does not require any modification, enabling billions of phones, laptops, and IoT devices to leverage the capability. This chapter presents *BackDoor*, a system that develops the technical building blocks for harnessing this opportunity, leading to new applications in security and communications.

(1) Security: Given that microphones record these inaudible sounds, it should be possible to silently jam spy microphones from recording. Military and government officials can secure private and confidential meetings from electronic eavesdropping; cinemas and concerts can prevent unauthorized recording of movies and live performances. We also realized the possibility of security threats. Denial-of-service (DoS) attacks on sound devices are typically considered difficult as the jammer can be easily detected. However, *BackDoor* shows that inaudible jammers can disable hearing aids and cellphones without getting detected. For

This chapter revises the publication "BackDoor: Making Microphones Hear Inaudible Sounds," in MobiSys 2017 [3].

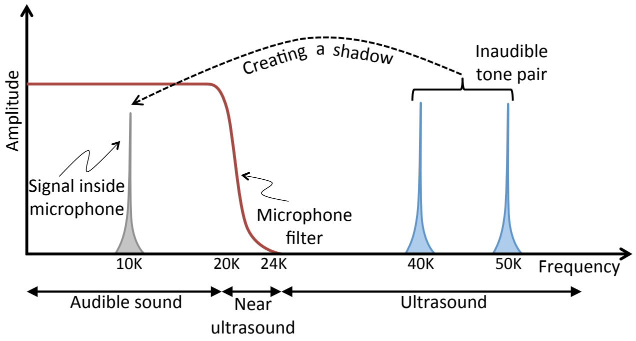


Figure 2.1: The main idea of frequency translation underlying *BackDoor*.

example, during a robbery, the perpetrators can prevent people from making 911 calls by silently jamming the microphones in all phones.

(2) Communications: Ultrasound systems today aim to achieve inaudible data transmissions to the microphone [1]. However, they suffer from limited bandwidth, around 3 kHz, since they must remain above the human hearing range (20 kHz) and below the microphone's cutoff frequency (24 kHz). Moreover, FCC imposes strict power restrictions on these bands since they are partly audible to infants and pets [4]. *BackDoor* is free of these limitations. Using an ultrasound-based transmitter, it can utilize the entire microphone spectrum for communication. Thus, IoT devices could find an alternative channel for communication, reducing the growing load on Bluetooth (BLE). Museums and shopping malls could use acoustic beacons to broadcast information about nearby art pieces or products. Various ultrasound ranging schemes, that compute *time of flight* of signals, could benefit from the substantially higher bandwidth in *BackDoor*.

This chapter focuses on developing the technical primitives that enable these applications. In the simplest case, *BackDoor* plays two tones at say 40 kHz and 50 kHz. When these tones arrive together at the microphone, they are received and amplified as expected, but also multiplied due to fundamental nonlinearities in the system. Multiplication of frequencies f_1 and f_2 result in frequency components at $(f_1 - f_2)$ and $(f_1 + f_2)$. Given that $(f_1 - f_2)$ is 10 kHz in this case, well within the microphone's range, the signal passes unaltered through the low-pass filter (LPF). Human ears, on the other hand, do not exhibit such nonlinearities and completely filter out the 40 kHz and 50 kHz sounds.

While the above is a trivial case of sending a tone, *BackDoor* intends to load data on transmitted carrier signals and demodulate the "shadow" after receiving through the microphone. This entails challenges. First, the nonlinearities we intend to exploit are not unique to the microphone; they are also present in speakers that transmit the sounds. As a result, the speaker also produces a “shadow” within the audible range, making its output audible to humans. We address this by using multiple speakers and isolating the signals in frequency across the speakers. We show, both analytically and empirically, that none of these isolated sounds create a “shadow” as they pass through the speaker’s diaphragm and amplifier. However, once these sounds arrive and combine nonlinearly inside the microphone, the “shadow” emerges within the audible range.

Second, for communication applications, standard modulation and coding schemes cannot be used directly. Section 2.4.1 shows how appropriate frequency-modulation, combined with inverse filtering, resonance alignment, and ringing mitigation are needed to boost achievable data rates. Finally, for security applications, jamming requires transmitting noisy signals that cover the entire audible frequency range. With audible jammers, this requires speakers to operate at very high volumes. Section 2.4.2 describes how *BackDoor* is designed to achieve equally effective jamming, but in complete silence. We leverage the *adaptive gain control* (AGC) in microphones, in conjunction with selective frequency distortion, to improve jamming at modest power levels.

The final *BackDoor* prototype is built on customized ultrasound speakers and evaluated for both communication and security applications across different types of mobile devices. Our results reveal the following:

- The 100 different sounds played to seven individuals confirmed that *BackDoor* was completely inaudible.

- *BackDoor* attained data rates of 4 kbps at a distance of 1 meter, and 2 kbps at 1.5 meters – this is 2× higher in throughput and 5× higher in distance than systems that use the near-ultrasound band.

- *BackDoor* is able to jam and prevent the recording of any conversation within a radius of 3.5 meters (and potentially a room-level coverage with higher power [5]). When 2000 English words were played back to seven humans and a speech recognition software [6], fewer than 15% of the words were decoded correctly. Audible jammers, aiming at comparable performance, would need to play white noise at a loudness of 97 dBSpl, considered seriously harmful to human ears [7].

In sum, this chapter elaborates on the following contributions:

- *Exploits nonlinearities in off-the-shelf microphones to enable a “backdoor” from high to low frequencies.* This backdoor permits playback of high-frequency sounds that are inaudible to humans and yet recordable through microphones.

- *Builds enabling primitives for applications in acoustic communication and privacy.* The acoustic radio outperforms today's near-ultrasound systems, while jamming raises the bar against eavesdropping.

The subsequent sections expand on these contributions. We begin with an acoustic primer, followed by intuitions, system design, and evaluation.

2.2 Acoustic Systems Primer

2.2.1 Common Microphone Systems

Any sound recording system requires two main modules – a transducer and an analog-to-digital converter (ADC). The transducer contains a "diaphragm" that vibrates due to sound pressure, producing a proportional change in voltage. The ADC measures this voltage variation (at a fixed sampling frequency) and stores the samples in memory. These samples represent the recorded sound in the digital domain.

A practical microphone needs two more components between the diaphragm and the ADC, namely a *pre-amplifier* and a *low-pass filter*. Figure 2.2 shows the pipeline. The pre-amplifier's task is to amplify the output of the transducer by a gain of around 10× so that the ADC can measure the signal effectively using its predefined quantization levels. Without this amplification, the signal is too weak (around tens of millivolts).

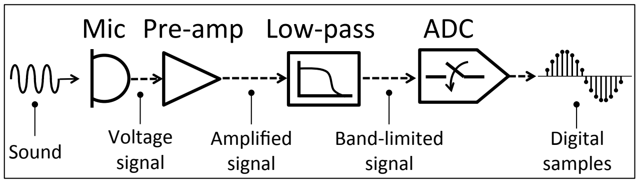


Figure 2.2: The sound recording signal flow.

As per Nyquist's law, if the ADC's sampling frequency is f_s Hz, the sound must be band limited to $\frac{f_s}{2}$ Hz to avoid aliasing and distortions. Since natural sound can spread over a wide band of frequencies, it needs to be low pass filtered (i.e., frequencies greater than $\frac{f_s}{2}$ removed) before the A/D conversion. Since ADCs in today's microphones operate at 48 kHz, the low-pass filters (LPFs) are designed to cut off signals at 24 kHz. Figure 2.3 shows the effect of the low-pass (or anti-aliasing) filter on the recorded sound spectrum.

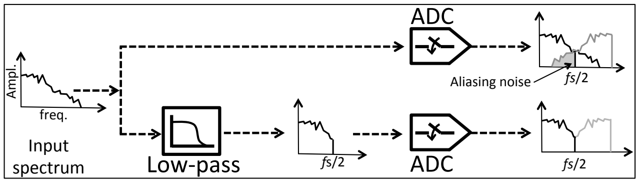


Figure 2.3: The digital spectrum with and without the (anti-aliasing) low-pass filter.

Sound Playback through Speakers: Sound playback is simply the reverse of recording. Given a digital signal as input, the digital-to-analog converter (DAC) produces the corresponding analog signal and feeds it to the speaker. The speaker's diaphragm oscillates to the applied voltage producing varying sound pressures in the medium, which is then audible to humans.

2.2.2 Linear and Nonlinear Behavior

Modules inside a microphone are mostly linear systems, meaning that the output signals are linear combinations of the input. If the input sound is S, then the output can be represented by

$$S_{out} = A_1 S$$

Here A_1 is a complex gain that can change the phase and/or amplitude of the input frequencies, but does not generate spurious new frequencies. This behavior makes it possible to record an exact (but higher-power) replica of the input sound and playback without distortion.

In practice, however, microphone hardware maintain strong linearity only in the audible frequency range; outside this range, the response exhibits nonlinearity. The diaphragm also exhibits similar behavior. Thus, for $f > 25$ kHz, the net recorded sound S_{out} may be expressed in terms of the input sound S as follows:

$$S_{out}\bigg|_{f>25} = \sum_{i=1}^{\infty} A_i S^i = A_1 S + A_2 S^2 + A_3 S^3 + ...$$

While in theory the nonlinear output is an infinite power series, the third- and higher-order terms are extremely weak and can be ignored. *BackDoor* finds opportunities to exploit the second-order term, which can be manipulated by designing the input signal S.

2.3 Core Intuition and Validation

As mentioned earlier, our core idea is to operate the microphone at high (inaudible) frequencies, thereby invoking the nonlinear behavior in the diaphragm and pre-amplifier. This is counterintuitive because most researchers and engineers strive to avoid nonlinearity. In our case, however, we intend to create an inlet into the audible frequency range and nonlinearity is essentially the "backdoor". We sketch the basic technique next, followed by some measurements to validate assumptions.

To operate the microphone in its nonlinear range, we use an off-the-shelf ultrasound speaker and play a sound S, composed of two inaudible tones $S_1 = 40$ and $S_2 = 50$ kHz. Mathematically, $S = Sin(2\pi 40t) + Sin(2\pi 50t)$. After passing through the diaphragm and pre-amplifier of the microphone, the output S_{out} can be modeled as:

$$S_{out} = A_1(S_1 + S_2) + A_2(S_1 + S_2)^2$$

$$= A_1\big\{Sin(\omega_1 t) + Sin(\omega_2 t)\big\} \; + A_2\big\{Sin^2(\omega_1 t) +$$

$$Sin^2(\omega_2 t) + 2Sin(\omega_1 t)Sin(\omega_2 t)\big\}$$

where $\omega_1 = 2\pi 40$ and $\omega_2 = 2\pi 50$.

Now, the first-order terms produce frequencies ω_1 and ω_2, which lie outside the microphone's cutoff. The second-order terms, however, is a multiplication of signals, resulting in various frequency components, namely, $2\omega_1$, $2\omega_2$, $(\omega_1 - \omega_2)$, and $(\omega_1 + \omega_2)$. Mathematically,

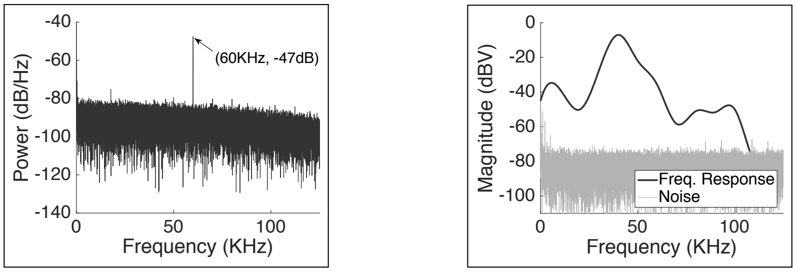
$$A_2(S_1 + S_2)^2 = 1 - \frac{1}{2}Cos(2\omega_1 t) - \frac{1}{2}Cos(2\omega_2 t) + Cos((\omega_1 - \omega_2)t) - Cos((\omega_1 + \omega_2)t)$$

With the microphone's cutoff at 24 kHz, all of the above frequencies in S_{out} get filtered out by the LPF, except $Cos((\omega_1 - \omega_2)t)$, which is essentially a 10 kHz tone. The ADC is oblivious of how this 10 kHz signal was generated and records it like any other sound signal. We call this the "shadow" signal. The net effect is that a completely inaudible frequency has been recorded by unmodified off-the-shelf microphones.

2.3.1 Measurements and Validation

For the above idea to work with unmodified off-the-shelf microphones, two assumptions need validation. (1) The diaphragm of the microphone should exhibit some sensitivity at the high-end frequencies (> 30 kHz). If the diaphragm does not vibrate at such frequencies, there is no opportunity for nonlinear mixing of signals. (2) The second-order coefficient A_2 needs to be adequately high to achieve a meaningful signal-to-noise ratio (SNR) for the shadow signal, while the third- and fourth-order coefficients (A_3, A_4) should be negligibly weak. We verify these next.

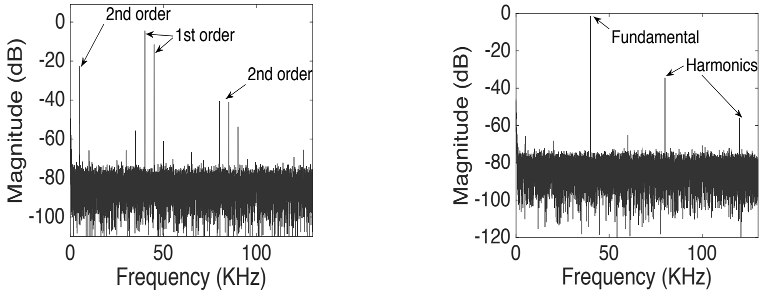
(1) Sensitivity to High Frequencies: Figure 2.4 reports the results when a 60 kHz sound was played through an ultrasonic speaker and recorded with a programmable microphone circuit. To verify the presence of a response at this high frequency, we "hacked" the circuit using an FPGA kit, and tapped into the signal before it entered the low-pass filter (LPF). Figure 2.4(a) shows the clear detection of the 60 kHz tone, confirming that the diaphragm indeed vibrates to ultrasounds. We also measured the channel frequency response at the output of the pre-amplifier (before the LPF); Figure 2.4(b) illustrates the results. The take away message is that the analog components indeed operate at a much wider bandwidth; it is the digital domain that restricts the operating range.



(a) (b)

Figure 2.4: (a) Microphone signals (measured before the LPF) confirm the diaphragm and pre-amplifier's sensitivity to ultrasound frequencies. (b) Full frequency response at the output of the amplifier.

(2) Magnitude of Nonlinear Coefficients: Figure 2.5(a) shows the entire spectrum after the nonlinear mixing has occurred, but before the LPF. Except for the shadow at ($\omega_1 - \omega_2$), we observe that all other frequency spikes are above the LPF's 24 kHz cutoff frequency. Similarly, the nonlinear effect on a single frequency – shown in Figure 2.5(b) – only produces integer multiples of the original frequency, i.e., ω, 2ω, 3ω, and so on. These two types of nonlinear distortions are called *intermodulation* and *harmonic* distortions, respectively. Importantly, the shadow signal is still conspicuous above the noise floor, while the third-order distortion is marginally above noise. This confirms the core opportunity to leverage the shadow.



(a) (b)

Figure 2.5: (a) The intermodulation distortion of signal. (b) The harmonic distortion of signal.

2.3.2 Hardware Generalizability

Before concluding this section, we report measurements to confirm that nonlinearities are present in different kinds of hardware (not just a specific make or model). To this end, we played high-frequency sounds and recorded them across a variety of devices, including smartphones (iPhone 5S, Samsung Galaxy S6), smartwatch (Samsung Gear2), video camera (Canon PowerShot ELPH 300HS), hearing aids (Kirkland Signature 5.0), laptop (MacBook Pro), etc. Figure 2.6 summarizes the SNR for the shadow signals for each of these devices. The SNR is uniformly conspicuous across all the devices, suggesting potential for widespread applicability.

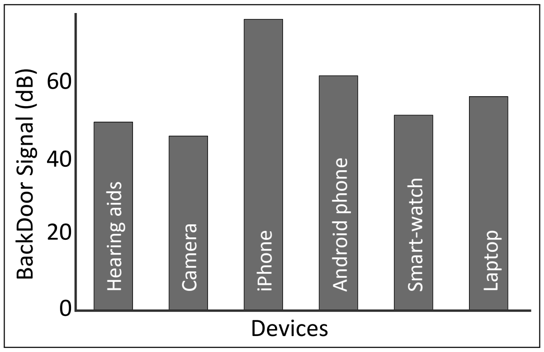


Figure 2.6: Consistent shadow at 5 kHz (in response to 45 and 50 kHz ultrasound tones) confirms nonlinearity across various microphone platforms.

2.4 System Design

This section details the two technical modules in *BackDoor*: communication and jamming.

2.4.1 Communication

Thus far, the shadow signal is a trivial tone carrying one bit of information (presence of absence). While this was useful for explanation, our actual goal is to modulate the high-frequency signals at the speaker and demodulate the shadow at the microphone to achieve meaningful data rates. We discuss the challenges and opportunities in developing this communication system.

(1) Failure of Amplitude Modulation (AM):
Our first idea was to modulate a single ultrasound tone, a data carrier, with a message signal $m(t)$. Assuming amplitude modulation [8, 9], this results in $m(t)Sin(\omega_c t)$, where ω_c is a high frequency, ultrasound carrier. Now, if $m(t) = a.Sin(\omega_m t)$, then the speaker should produce this signal:

$$S_{AM} = aSin(\omega_m t)Sin(\omega_c t)$$

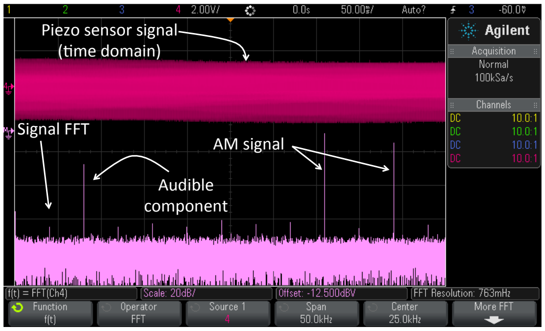
Now, when this signal arrives at the microphone and passes through the nonlinearities, the squared components of the amplifier's output will be:

$$\begin{aligned} S^2_{out,AM} &= A_2\big\{aSin(\omega_m t).Sin(\omega_c t)\big\}^2 \\ &= -A_2\frac{a^2}{4}\big\{Cos(\omega_c t - \omega_m t) - Cos(\omega_c t + \omega_m t)\big\}^2 \\ &= -A_2\frac{a^2}{4}Cos(2\omega_m t) + (terms\ with\ frequencies\ above\ \omega_c\ and\ DC) \end{aligned}$$

The result is a signal that contains a $Cos(2\omega_m t)$ component. So long as ω_m, the frequency of the data signal, is less than 10 kHz, the corresponding shadow at $2\omega_m = 20$ kHz is within the LPF cutoff. Thus, the received sound data can be band-pass filtered in software, and the data signal correctly demodulated.

Importantly, the above phenomenon is reminiscent of *coherent demodulation* in conventional radios, where the receiver would have multiplied the modulated signal, which can be represented as $(aSin(\omega_m t)Sin(\omega_c t))$, with the frequency and phase-synchronized carrier signal $Sin(\omega_c t)$. The result would be the $m(t)$ signal in baseband, i.e., the carrier frequency ω_c eliminated. Our case is somewhat similar – the carrier also gets eliminated, and the message signal appears at $2\omega_m$ (instead of ω_m). This is hardly a problem since the signal can be extracted via band-pass filtering. Thus, the net benefit is that the microphone's nonlinearity naturally demodulates the signal and translates to within the LPF cutoff, requiring no changes to the microphone. Put differently, *nonlinearity may be a natural form of self-demodulation and frequency translation*, the root of our opportunity.

Unfortunately, the ultrasound transmitter – a speaker with a diaphragm – also exhibits nonlinearity. The above property of self-demodulation triggers in the transmitter side as well, resulting in $m(t)$ becoming audible. Figure 2.7 shows the output of the speaker as visualized by the oscilloscope; a distinct audible component appears due to amplitude modulation. In fact, any modulation that generates *waveforms with non-constant envelopes* [10] is likely to suffer this problem. This is unacceptable and brings forth the first design question: *how to cope with transmitter-side nonlinearity?*



f(t) = FFT(Ch4) Scale: 20dB/ Offset: -12.500dBV FFT Resolution: 763mHz

Figure 2.7: The AM signal produces an audible frequency due to self-demodulation, shown in this oscilloscope screenshot.

(2) Bypassing Transmitter Nonlinearity:

The design goal at this point is to modulate the carrier signal with data without affecting the envelope of the transmitted signal. This raises the possibility of *angle modulation* (i.e, modulating the phase or frequency but leaving amplitude untouched). However, we recognized that phase modulation (PM) is also unsuitable in this application because of unpredictable noise from phone movements. In particular, the smaller wavelength of ultrasonic signals are easily affected by phase noise and involves complicated receiver-side schemes during demodulation. Therefore, we choose the other alternative of angle modulation: *frequency modulation* (FM). Of course, FM modulation is not without tradeoffs; we discuss them and address the design questions step by step.

(3) FM without Frequency Translation:

FM modulated signals, unlike AM, do not get naturally demodulated or frequency-translated when pass through nonlinear transmitter. Assuming $Cos(\omega_m t)$ as the message signal, we have the input to the speaker as:

$$S_{fm} = Sin(\omega_c t + \beta Sin(\omega_m t))$$

Note that the phase of the FM carrier signal should be the integral of the message signal, hence it is $Sin(\omega_m t)$. Now when S_{fm} gets squared due to nonlinearity, the result is of the form $(1 + Cos(2\omega_c t + otherTerms))$ i.e., a DC component and another component at $2\omega_c$. Hence, along with the original ω_c frequency the transmitter output contains frequency at $2\omega_c$, both above the audible cutoff. Thus nothing gets recorded by the microphone. The advantage, however, is that the output of the speaker is no longer audible. Moreover, as typically the speaker has a low response at high frequencies near $2\omega_c$, the output signal is dominated by the data signal at ω_c as in original S_{fm}.

(4) Second Carrier for Frequency Translation:

To get the message signal recorded, we need to frequency-shift the signal at ω_c to the microphone's audible range, without affecting the signal transmitted from the speaker. To achieve this, *BackDoor* introduces a second ultra-sound signal transmitted from a second speaker collocated with the first speaker. Let us assume this second signal is called the *secondary carrier*, ω_s. Since ω_s does not mix with ω_c at the transmitter, the signal that arrives at the microphone diaphragm is simply of the form:

$$S_{fm}^{Rx} = \Bigg(A_1 Sin(\omega_c t + \beta Sin\omega_m t) + A_1 Sin(\omega_s t)\Bigg)$$

Note that the first term from the FM modulated ω_c signal, and the second term from the ω_s secondary carrier. Now, upon arriving on the receiver, the microphone's nonlinearity essentially squares this whole signal as $(S_{fm}^{Rx})^2$. Expanding this mathematically results in a set of frequencies centered at $(\omega_c - \omega_s)$, and the others at $(\omega_c + \omega_s)$, $2\omega_c$, and $2\omega_s$. If we design ω_c and ω_s to have a difference less than the LPF cutoff, the microphone can record the signal.

Choosing ω_c and ω_s :

As we considered the requirements of the system, the choice of ω_c and ω_s became clear. First, note that the FM-modulated signal has a bandwidth of, say $2W$, ranging from $(\omega_c - W)$ to $(\omega_c + W)$. Thus, assuming that the microphone's LPF cutoff is 20 kHz, we should translate the center frequency to 10 kHz; this maximizes W that can be recorded by the microphone. Immediately, we know that $(\omega_c - \omega_s) = 10$ kHz.

Second, the microphone's diaphragm exhibits resonance at certain frequencies; ω_c and ω_s should leverage this to improve the strength of the recorded signal. Figure 2.8 plots the normalized power of the *translated signal* for different values of ω_c and ω_s. Given $(\omega_c - \omega_s) =$ 10 kHz, the resonance effects demonstrate the maximum response when ω_c is 40 kHz, and ω_s is 50 kHz.

(5) Coping with the "Ringing" Effect:

The piezo-electric material in the speaker, that actually vibrates to create the sound, behaves

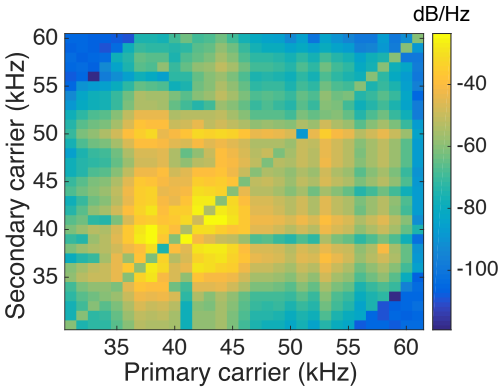
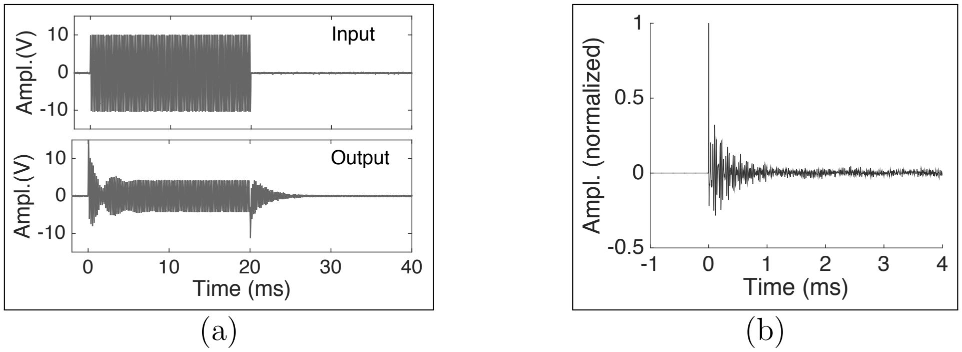


Figure 2.8: Resonance for various $\omega_c - \omega_s$ values.

as an oscillatory inductive-capacitive circuit. This loosely means that the actual vibration is a weighted sum of input sound samples (from the recent past), and hence, the piezoelectric material has a heavy-tailed impulse response (shown in Figure 2.9). Mathematically, the output of the speaker can be computed as a convolution between this impulse response and the input signal. Unfortunately, the nonlinearity of the speaker impacts this convolution process as well, and generates low frequency components similar to the natural demodulation effect discussed earlier. The result is a "ringing effect", i.e., *the transmitted sound becomes slightly audible even with FM modulation.*



(a)

(b)

Figure 2.9: (a) The prolonged oscillation in an ultrasonic transmitter following a 40 kHz sine burst input. (b) The impulse response of the ultrasonic transmitter.

To explain the self-demodulation effect, we assume a simplified impulse response "h":

$$h = \sum_{i=0}^{\infty} k_i \delta(t-i) \approx k_0 \delta(t) + k_1 \delta(t-1)$$

When an angle modulated (FM/PM) signal "S" is convolved with "h", the output "S_{out}" is:

$$\begin{aligned} S_{out} &= S * h \\ &= sin(\omega_c t + \beta sin(\omega_m t)) * (k_0 \delta(t) + k_1 \delta(t-1)) \\ &= \quad k_0 sin(\omega_c t + \beta sin(\omega_m t)) \\ &\quad + \quad k_1 sin(\omega_c (t-1) + \beta sin(\omega_m (t-1))) \end{aligned}$$

While S_{out} contains only high-frequency components (since convolution is linear), the non-linear counterpart S_{out}^2 mixes the frequencies in a way that has lower-frequency components (or shadows):

$$\begin{aligned} S_{out}^2 &= k_0 k_1 cos(\omega_c + 2\beta sin(\frac{\omega_m}{2}) sin(\omega_m t - \frac{\omega_m}{2})) \\ &\quad + (terms\ with\ frequencies\ over\ 2\omega_c\ and\ DC) \end{aligned}$$

Figure 2.10 shows the spectrum of S_{out} and S_{out}^2, with and without the convolution. Observe the low frequency "shadow" that appear due to the second-order term for the convolved signal – this shadow causes the ringing and is noticeable to humans.

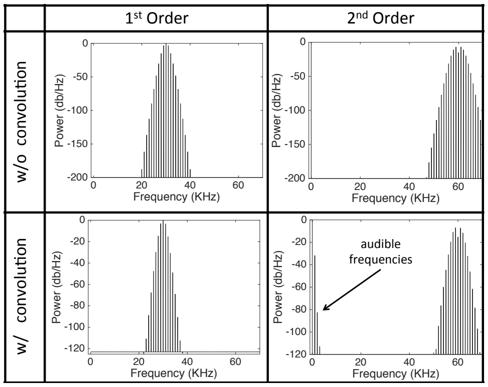


Figure 2.10: The spectrogram of S_{out} and S_{out}^2, with and without the convolution. The shadow signal appears due to second-order nonlinear effects on the convolved signal.

In most speakers, this "shadow" signal is weak; some expensive speakers even design their piezo-electric materials to be linear in a wider operating region precluding this possibility. However, we intend to be functional across all speaker platforms (even the cheapest ones) and aim to be completely free of any ringing whatsoever. Hence, we adopt an inverse filtering approach to remove ringing.

(6) Inverse Filtering to Eliminate Ringing:

Our core idea draws inspiration from *pre-coding* in wireless communication, i.e., we modify the input signal S_{fm} so that it remains the same after convolution. In other words, if the modified signal $S_{mod} = h^{-1} * S_{fm}$, then the impact of convolution on S_{mod} results in $h * h^{-1} * S_{fm}$, which is S_{fm} itself. With S_{fm} as the output of the speaker, we do not experience ringing. Of course, we need to compute h^{-1}, i.e., learn the coefficients of the impulse response. For this, we monitor the current passing through the ultrasonic transmitter at different frequencies and calculate the $(k_0, k_1, k_2, ...)$. Fortunately, unlike wireless channels, the response of the transmitter does not vary over time and hence the coefficients of the inverse filter can be pre-calculated. Figure 2.11(a) shows the frequency response of one of our ultrasound speakers, while Figure 2.11(b) shows how our inverse filtering scheme curbs the ringing effect.

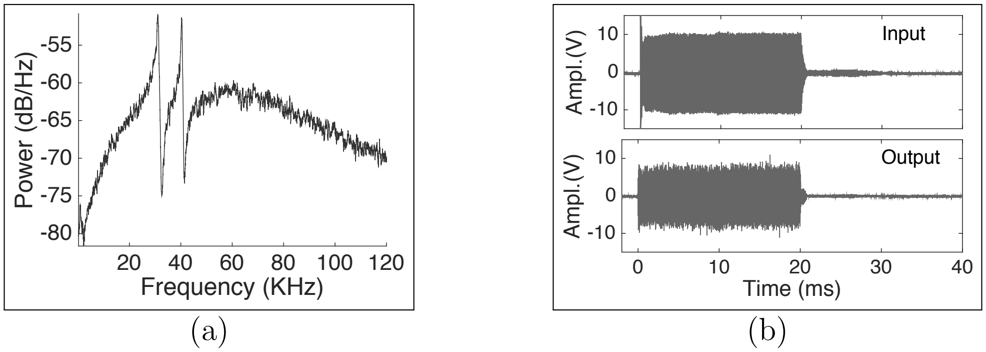


Figure 2.11: (a) Frequency response of the ultrasonic speaker. (b) Inverse filtering method almost eliminates ringing effect compared to Figure 2.9.

(7) Receiver Design:

This completes the transmitter design and the receiver is now an unmodified microphone (from off-the-shelf phones, cameras, laptops, etc.). Of course, to extract the data bits, we need to receive the output signal from the microphone and decode them in software. For example, in smartphones, we have used the native recording app, and operated on the stored signal output. The decoding steps are as follows.

We begin by band-pass filtering the signal as per the modulating bandwidth. Then, we need to convert this signal to its baseband version and calculate the instantaneous frequency to recover the modulating signal $m(t)$. This signal contains the negative-side frequencies that overlap with the spectrum-of-interest during the baseband conversion. To remove the negative frequencies, we Hilbert transform the signal, producing a complex signal [11]. Now, for baseband conversion, we multiply this complex signal with another complex signal $e^{-j2\pi(\omega_s - \omega_c)t}$. Here $(\omega_s - \omega_c)$ is 10 kHz, i.e., the shifted carrier frequency. This operation brings the modulated spectrum to baseband, centered around DC. The differentiation of its phase gives the instantaneous frequency [12], which is then simply mapped to data bits. Section 2.5 will present performance evaluation, but before that, we present the techniques for inaudible voice jamming.

2.4.2 Jamming

Imagine military applications in which a private conversation needs to be held in an untrusted environment, potentially bugged with spy microphones. We envision turning on one/few *BackDoor* devices in that room. The device will broadcast appropriately designed ultrasound signals that will not interfere with human conversation, but will jam microphones in the vicinity. This section targets two jamming techniques toward this goal: (1) passive gain suppression, and (2) active frequency distortion. Together, the techniques mitigate electronic eavesdropping.

(1) Passive Gain Suppression:
Our core idea is to leverage the *automatic gain control* (AGC) circuit [13, 14, 15] in the microphone to suppress voice conversations. By transmitting a narrowband ultrasound frequency at high amplitude, we expect to force the microphone to alter its dynamic range, thereby weakening the SNR of the voice signal. We elaborate next, beginning with a brief primer on AGC.

AGC Primer: Our acoustic environment has large variations in volume levels ranging from soft whispers to loud bangs. While human ears seamlessly handle this dynamic range, it poses one of the major difficulties in microphones. Specifically, when a microphone is configured at a fixed gain level, it fails to record a soft signal below the minimum quantization limit, while a loud sound above the upper range is clipped, causing severe distortions. To cope, microphones use an Automatic Gain Control (AGC) (as a part of its amplifier circuit) that adjusts the signal amplitude to fit well within the ADC's lower and upper bounds. As a result, the signal covers the entire range of the ADC, offering the best possible signal resolution.

Figure 2.12 demonstrates the AGC operation in a common MEMS microphone (ADMP401) connected to the line-in port of a Linux laptop running the ALSA sound driver. We simultaneously play 5 kHz and 10 kHz tones through two different (but collocated) speakers and display the power spectrum of the received sound. Figure 2.12(a) reports both the signals at around -20 dB. However, when we increase the power of the 10 kHz signal to reach its AGC threshold (while keeping the 5 kHz signal unaltered), Figure 2.12(b) shows how the microphone reduces the overall gain to accommodate the loud 10 kHz signal. This results in a 25 dB reduction of the unaltered 5 kHz signal.

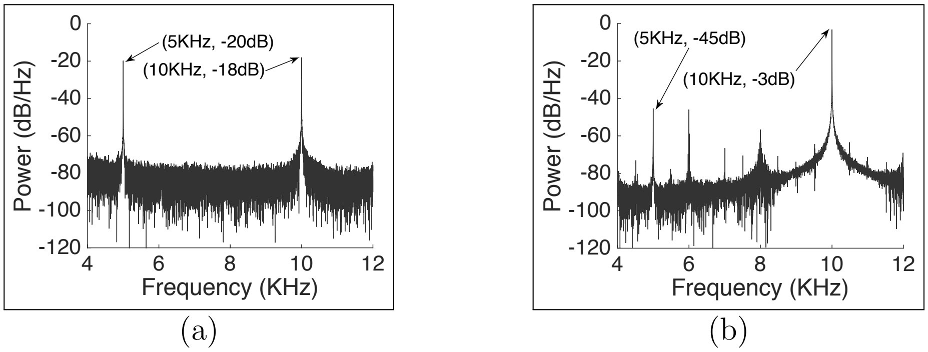


Figure 2.12: Automatic gain control: (a) The 5 kHz tone is at −20 dB when the amplitude of the 10 kHz frequency is at comparable power. (b) The 5 kHz tone reduces to −45 dB when the amplitude of the 10 kHz tone is made to exceed the AGC threshold. Some spurious frequencies also appear due to nonlinearities.

Voice Suppression via AGC: In line with the above idea, when our ultrasound signal at ω_c passes through the AGC (i.e., before this frequency is removed by the low-pass filter), it alters the AGC gain configuration and significantly suppresses the voice signals in the audible frequency. Figure 2.13 shows the reduction in the received sound power in a Samsung Galaxy S-6 smartphone when ultrasound tones are played at different frequencies from a piezoelectric speaker. Evident from the plot, the maximum reduction is due to the signal at 40 kHz – this is because 40 kHz is the resonance frequency of the piezoelectric transducer, and thereby delivers the highest power. In that sense, using the resonance frequency offers double gains, one toward increasing the SNR of our communication signal, and the other for jamming.

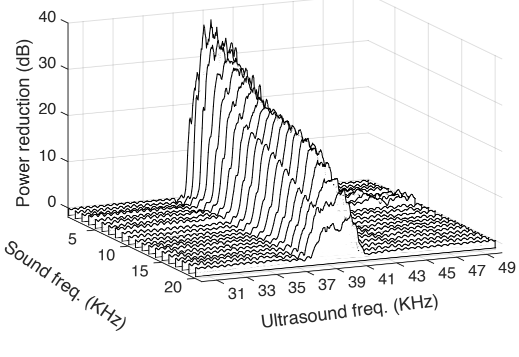


Figure 2.13: The reduction in sound power due to the AGC. The reduction is maximum for the 40 kHz tone due to the speaker's resonance at this frequency.

This reduction in signal amplitude results in low resolution when sampled with discrete quantization levels at the ADC. In fact, an adequately loud ultrasonic tone can completely prevent the microphone from recording any meaningful voice signal by reducing its amplitude below the minimum quantization level. However, as the electrical noise level is usually higher than the minimum quantization level of the ADC, it is sufficient to reduce the signal power below that noise floor.

Figure 2.14 shows the reduction in the signal power of a recorded voice segment for three different power levels of the 40 kHz tone. In practice, an absolute amplitude reduction is difficult unless the speaker uses high power. Importantly, high power speakers are possible with *BackDoor* since the jamming signal is inaudible. *On the other hand, regular white noise audio jammers must operate below strict power levels to not interfere with human conversation/tolerance. This is a key advantage of jamming with BackDoor.* Nonetheless, we still attempt to lower the power requirement by injecting additional frequency distortions at the eavesdropper's microphone.

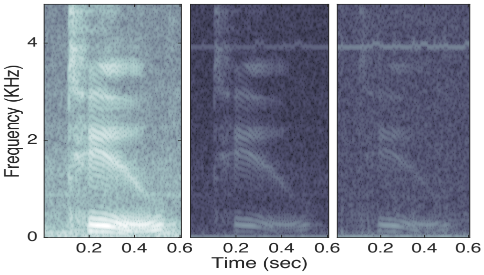
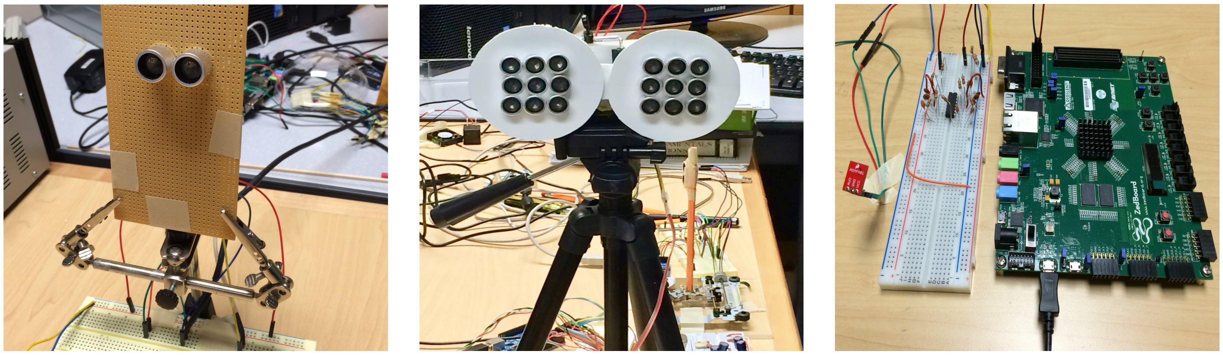


Figure 2.14: The reduction in signal power of recorded voice segment for three power levels (darker is lower power).

(2) Injecting Frequency Distortion:
A traditional jamming technique is to add strong white noise to reduce the SNR of the target signal. We first implement a similar technique, but with inaudible band-limited Gaussian noise. Specifically, we modulate the ω_c carrier with white noise, bandpass filtered to allow frequencies between 40 kHz to 52 kHz only. The 52 kHz ω_s carrier shifts this noise to $[0, 12]$ kHz, which is sufficient to affect the voice signal.

To improve, we then shape the white noise signal to boost power in frequencies that are known to be important for voice. Note that these distortions are designed in the ultrasound bands (to maintain inaudibility), and hence they are played through the ultrasound speakers. Section 2.5 will report results on word legibility, as a function of the separation between the jammer and the spy microphone.



(a) (b) (c)

Figure 2.15: *BackDoor* experimental setup: (a) Two ultrasonic speakers mounted on a circuit board for data communication. (b) A 2-watt speaker array system for jamming applications. (c) The FPGA-based setup for probing into individual components of the microphone.

2.5 Evaluation

BackDoor was evaluated on three main metrics: (1) human audibility, (2) throughput, packet error rates (PER) and bit error rates (BER) for data communication, and (3) the efficacy of jamming. We summarize the key results here, followed by details.

• Tables 2.1 and 2.2 report human perception of audibility for *BackDoor* for various frequencies, modulations, and SNR levels. Except for amplitude modulation (AM), all the human volunteers reported complete silence.

• Figures 2.17 and 2.18 report the variation of throughput against increasing distance, different phone orientations, and impact of acoustic interference. The results show throughput of 4 kbps at 1 meter away which is 2× to 4× higher than today's mobile ultrasound communication systems.

• Figure 2.19 compares the jamming radius for *BackDoor* and audible white noise-based jammers. To achieve the same jamming effect (say, $<$ 15% words legible by humans), we find that the audible jammer requires a loudness of 97 dBSpl which is similar to a jackhammer and can cause severe damage to humans [7]. *BackDoor*, on the other hand, remains completely silent. Conversely, when the white noise sound level is made tolerable, the legibility of the words was 76%.

We elaborate on these results below, starting with details on our implementation platform.

2.5.1 Implementation

(1) Transmitter Speakers: Figure 2.15(a) and (b) show two different transmitter prototypes we have developed, the first one for communication and the other for jamming. The communication transmitter consists of two ultrasonic piezoelectric speakers [16]; each transmits a separate frequency as described in Section 2.4. A programmable waveform generator (Keysight 33500b series) drives the speakers with frequency modulated signals. The signals are amplified using an NE5535AP op-amp based non-inverting amplifier, permitting signals up to 150 kHz. The jamming transmitter in Figure 2.15(b) is composed of two speaker arrays, each array with nine piezoelectric speakers connected in parallel to generate a 2 watt jamming signal. The signals driving these arrays are first amplified using an LM380 op-amp based power amplifier separately powered from a constant DC-voltage source. Figure 2.16 shows the circuit diagram of the speaker array.

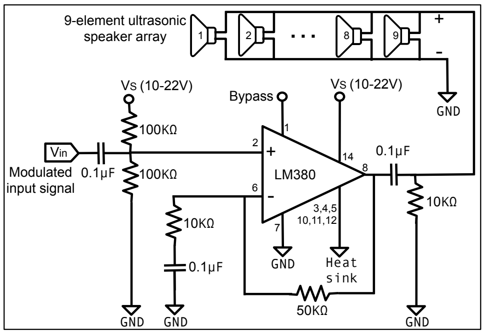


Figure 2.16: The circuit diagram of the jamming transmitter.

(2) Receiver Microphones: We experiment with two types of receivers. The first is an off-the-shelf Samsung Galaxy S6 smartphone (released in Aug, 2015) running Android OS 5.1.1. Signals are recorded through a custom Android app using the standard APIs. The second receiver is shown in Figure 2.15(c) – a more involved setup that was mainly used for micro-benchmarks reported earlier in Sections 2.3 and 2.4. This allowed us to tap into different components of the microphone pipeline, and analyze signals in isolation. The system runs on a high bandwidth data acquisition ZedBoard, a Xilinx Zynq-7000 SoC based FPGA platform [17], that offers a high-rate internal ADC (up to 1 Msample/sec). A MEMS microphone (ADMP 401) is externally connected to this ADC, offering undistorted insights into higher-frequency bands of the spectrum.

Table 2.1: Perceived loudness of *BackDoor* in comparison to audible sounds for unmodulated signals.

Reference Mic.	2 kHz Tone		5 kHz Tone	
SNR (dB)	*BackDoor*	Audible	*BackDoor*	Audible
25	0	0.75	0	3.33
30	0	1.5	0	4.08
35	0	2	0	4.91
40	0	2.67	0	5.42
45	0	3.17	0	6.17

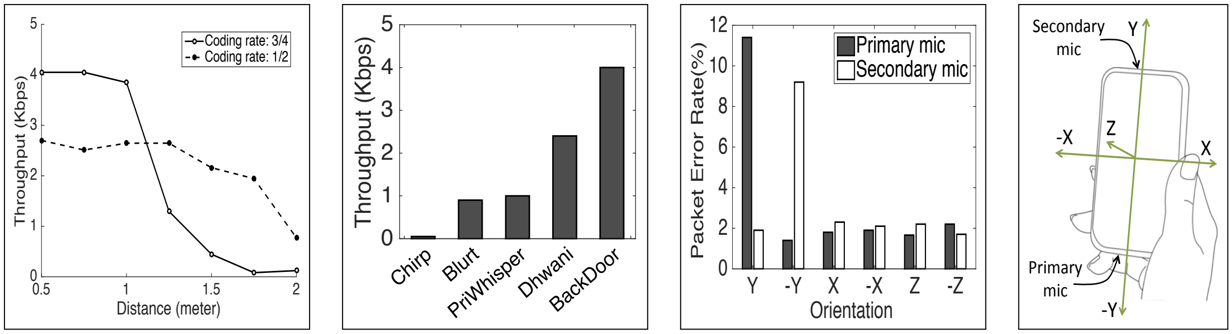
Table 2.2: Perceived loudness of *BackDoor* in comparison to audible sounds for modulated signals.

Reference Mic.	FM		AM		White Noise	
SNR (dB)	*BackDoor*	Audible	*BackDoor*	Audible	*BackDoor*	Audible
25	0	1.2	0	0.46	0	0.1
30	0	2.3	0.1	1.36	0	0.26
35	0	3.5	0.1	1.85	0	0.5
40	0	4.2	0.16	2.4	0	0.8
45	0	4.8	0.68	3.06	0	1.24

2.5.2 Human Audibility Results

We played *BackDoor* signals to a group of seven users (ages between 27 and 38) seated around a table 1 to 3 meters away from the speakers. Each user reported the perceived loudness of the sound on a scale of 0-10, with 0 being perceived silence. As a baseline, we also played audible sounds and asked the users to report the loudness levels. A reference microphone is placed at 1m from the speaker to record and compute the SNR (Signal to Noise Ratio) of all the tested sounds. We varied the SNR and equalized them at the microphone for fair comparison between audible and inaudible (*BackDoor*) sounds.

Four types of signals were played: (1) Single Tone Unmodulated Signals: In the simplest form, we transmitted multiple pairs of ultrasonic tones ($\langle 40, 42\rangle$ and $\langle 40, 45\rangle$) that generate a single audible frequency tone in the microphone. As baseline, we separately played a 2 kHz and 5 kHz audible tone. (2) Frequency Modulated Signals: We modulated the frequency of a 40 kHz *primary carrier* with a 3 kHz signal. We also transmitted a 45 kHz *secondary carrier* on the second speaker, producing 3 kHz FM signal centered at 5 kHz in the microphone. As baseline, we played the equivalent audible FM signal on the same speakers. (3) Amplitude Modulated Signals: Similar to FM signals, we created these AM signals by modulating the amplitude of 40 kHz signal with a 3 kHz tone. (4) White Noise Signals: Finally, we generated white Gaussian noise with zero mean and variance proportional to the transmitted power,



(a) (b) (c) (d)

Figure 2.17: *BackDoor* Communication Results: (a) Throughput vs. distance. (b) Throughput comparison against related P2P communication schemes. (c) Packet error rate vs. orientation. (d) Phone orientations.

at a bandwidth of 8 kHz, band-limited to $[40, 48]$ kHz. We also transmit a 40 kHz tone on the second speaker to frequency shift the white noise to the audible range of the speaker. As baseline, we create audible white noise with the same properties band-limited to $[0, 8]$ kHz and played it on the speakers.

Audibility vs. SNR: Tables 2.1 and 2.2 summarize the average of perceived loudness that users reported for both *BackDoor* and audible signals as a function of the SNR measured at the reference microphone. For all types of signals except amplitude modulation (AM), *BackDoor* is completely inaudible to all the users. AM signals are audible due to speaker nonlinearity, as described earlier. However, the perceived loudness of *BackDoor* is significantly lower than that of audible signals. Thus, so long we avoid AM, *BackDoor* signals remain inaudible to humans but produce audible signals inside microphones with the same SNR as loud audible signals.

2.5.3 Communication Results

The *BackDoor* transmitter is the two-speaker system while the receiver is the Samsung smartphone. The recorded acoustic signal is extracted and processed in MATLAB; we compute bit error rate (BER), packet error rate (PER) and throughput under varying parameters. Overall, 40 hours of acoustic transmission was performed to generate the results.

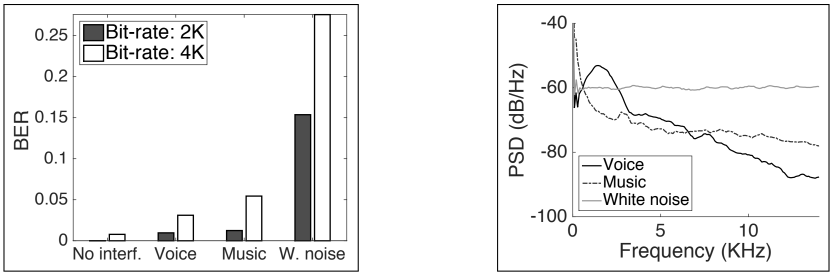
(1) Throughput:

Figure 2.17(a) reports *BackDoor*'s net end-to-end throughput for increasing separation between the transmitter and the receiver. *BackDoor* can achieve a throughput of 4 kbps at 1 m, 2 kbps at 1.5 m and 1 kbps at 2 m. Figure 2.17(b) compares *BackDoor*'s performance in terms of throughput and range with state-of-the-art mobile acoustic communication systems (in both commercial products [18, 19] and research [1, 20]). The figure shows that *BackDoor* achieve 2× to 80× higher throughput. This because these systems are constrained to a very narrow communication band whereas *BackDoor* is able to utilize the entire audible bandwidth.

(2) Impact of Phone Orientation:
Figure 2.17(c) shows the *packet error rate* (PER) when data is decoded by the primary and secondary microphones in the phone, placed in six different orientations (shown in Figure 2.17(d)). The aim here is to understand how real-world use of the phone impacts data delivery. To this end, the phone was held at a distance of 1 m away from the transmitter, and the orientation changed after each transmission session. The plot shows that except Y and $-Y$, the other orientations are comparable. This is because the $Y/-Y$ orientation aligns the two receivers and transmitters in almost a straight line, resulting in maximal SNR difference. Hand blockage of the further-away microphone makes the SNR gap pronounced. It should be possible to compare the SNR at the microphones and select the better microphone for minimized PER (regardless of the orientation).

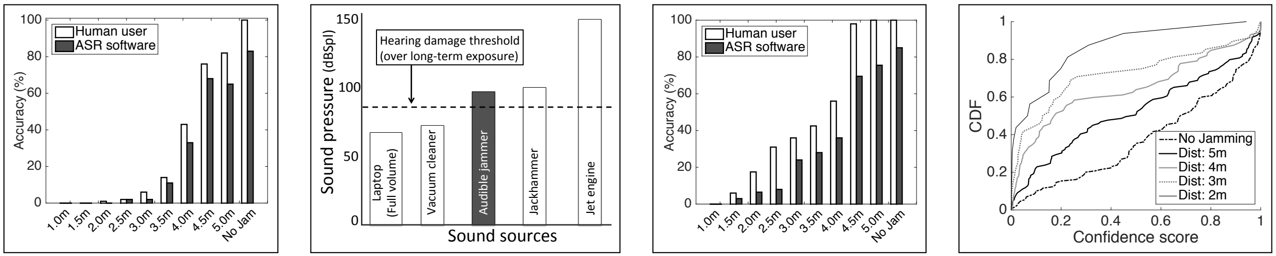
(3) Impact of Interference:
Figure 2.18(a) reports the bit error rate (BER) variation against three different audible interference sources. To elaborate, we played audible interference signals – a presidential speech, an orchestral music, and white noise – from a nearby speaker, while the data transmission was in progress. The intensity of the interference at the microphone was at 70 dBSpl, equaling the level of volume one hears on average in face-to-face conversations. This is certainly much louder than average ambient noise, and hence, this serves as a strict test for *BackDoor*'s resilience to interference. Also, the smartphone receiver was placed 1m away from the speaker, and transmissions were at 2 kbps and 4 kbps.

Evident from the graph, voice and music has minimal impact on the communication error. On the other hand, white noise can severely degrade performance. Figure 2.18(b) plots the power spectral density of each interference – the decay beyond 4 kHz for voice and music explains the performance plots. Put differently, since *BackDoor* operates around 10 kHz frequency, voice and music signals do not affect the band as much as white noise, that remains flat over the entire spectrum.



(a) (b)

Figure 2.18: (a) BER vs. interference. (b) Spectral density of interfering signals.



(a) (b) (c) (d)

Figure 2.19: Jamming results: (a) *BackDoor* jams a radius of 3.5m at 2W power. (b) White noise power needed to match *BackDoor* is intolerable. (c) Jamming radius when *BackDoor* uses inaudible white noise, showing importance of selectively jamming voice-centric harmonics. (d) Confidence of speech recognizer.

2.5.4 Jamming Results

Consider the case where Bob is saying a secret to Alice and Eve has planted a microphone in the vicinity, attempting to record Bob's voice. In suspicion, Bob places a *BackDoor* jammer in front of him on the table. We intend to report the efficacy of jamming in such a situation. Specifically, we extract the jammed signal from Eve's microphone and play it to an automatic speech recognizer (ASR), as well as to a group of seven human users. We define *Legibility* as the percentage of words correctly recognized by each. We plot L_{asr} and L_{human} for increasing jamming radius, i.e., for increasing distance between Alice and Eve's microphone.

We still need to specify another parameter for this experiment – the loudness with which Bob is speaking. Acoustic literature suggests that at social conversations, say between two people standing at arm's length at a corridor, the average loudness is 65 dBSpl (dB of sound pressure level). We design our situation accordingly, i.e., when Bob speaks, his voice at Alice's location 1 m away is made to be 70 dBSpl (i.e., Bob is actually speaking louder than general social conversations).

In the actual experiment, we pretend that a smartphone is a spy microphone. Another smartphone's speaker is a proxy for Bob, and the words played are derived from Google's Trillion Word Corpus [21]; we pick the 2000 most-frequent words, prescribed as a good benchmark [22]. As mentioned earlier, the volume of this playback is set to 70 dBSpl at 1 m away. Now, the *BackDoor* prototype plays an inaudible jamming signal through its ultrasonic speakers to jam these speech signals.

Our baseline comparison is essentially against *audible* white noise-based jammers in today's markets. Assuming *BackDoor* jams up to a radius of R, we compute the loudness needed by white noise to jam the same radius. All in all, 14 hours of sound was recorded and a total of $25,000$ words were tested. The ASR software is the open-source *Sphinx4* library (pre-alpha version) published by CMU [6, 23]. We present the results next.

(1) Audible and Inaudible Jamming Radius:
Figure 2.19(a) plots L_{asr} and L_{human} for increasing the jamming radius. Even with a 1 W power, a radius of 3.5 m (around 11 feet) can be jammed around Bob. We compare against audible noise jammers presented in Figure 2.19(b). For jamming at the same radius of 3.5 m, the loudness necessary for the audible white noise is 97 dBSpl which is the same as a jackhammer and can cause damage to the human ear [7]. Conversely, we find that when the audible white noise is made tolerable (comparable to a white noise smartphone app playing at full volume), the legibility becomes 76%. Thus, *BackDoor* is a clear improvement over audible jammers. More importantly, increasing the power of *BackDoor* jammers can increase the radius proportionally. This can be easily achieved. In fact, current portable Bluetooth speakers already transmit 10× to 20× higher power than *BackDoor* [24, 25]. Audible jammers cannot increase their power to boost the range since they are already intolerable to humans.

(2) Impact of Selective Frequency Distortion:
Figure 2.19(c) shows results when the jamming signal is simply a white noise, without the deliberate distortions of voice-centric frequencies (fricatives, phonemes, and harmonics). Evidently, the performance is substantially weaker, indicating the importance of signal shaping and jamming. Finally, Figure 2.19(d) shows the confidence scores from ASR for all correctly recognized words. Results show quite low confidence on a large fraction of words, implying that voice fingerprinting and other voice-controlled systems would be easy to DoS-attack with a *BackDoor*-like system.

2.6 Points of Discussion

Needless to say, there is much room for further work and improvement. We discuss a few points here.

- **Jamming Range:** *BackDoor*'s restriction in the jamming range stems from the attenuation of ultrasound in air and the limited amplitude at which the ultrasound speakers can vibrate, producing low power signals. We have demonstrated a proof-of-concept with nine speakers that boosts the jamming power level – direct materials cost is around $4. It should be certainly possible to develop a bigger speaker array to significantly increase the power [5]. In some cases (e.g. movie theater) multiple short-range jammers can be used to sufficiently cover the space. The jammers could be wall powered where necessary, and yet, will remain inaudible.

- **Smarter Spy:** We have assumed a fairly simple attacker planting a single microphone in the vicinity. Multiple microphones, perhaps even with various beamforming capabilities, may be able to extract out the voice from the jamming signal. However, greater sophistication in jamming should be feasible too, such as variation in the jamming signal to prevent channel estimation; even some movements of the speakers. We leave this to future work.

- **Interference with Phone Calls:** Data communication with *BackDoor* can interfere with people talking on the phone nearby. To this end, data communication applications will inherently need to be proximate and at low power. One possibility is an acoustic NFC, but at greater ranges of 1 or 2 feet. Alternatively, the communication could be made as a spread spectrum so that the interference remains below the noise floor. Our ongoing work is investigating these unresolved issues.

2.7 Related Work

(1) Literature in Acoustic Nonlinearity: The literature in acoustic signal processing and communication is extremely rich. The notion of exploiting nonlinearity was originally studied in the 1957 by Westervelt's seminal theory [26, 27], which later triggered a series of research. The core vision was that nonlinearities of the air can naturally self-demodulate signals; when combined with directional propagation of ultrasound signals, it may be possible to deliver audible information over large distances using relatively low power [28, 29, 30]. Recently, there has been a revival of these efforts with AudioSpotlight [31], SoundLazer [32, 33], and other projects [34, 35, 36]. Our work, however, is opposite of these efforts – we are attempting to retain the inaudible nature of ultrasound while making it recordable inside electronic circuits.

(2) Medical Devices: Human bones have also been shown to exhibit nonlinearities that self-modulate signals, resulting in applications in bone conduction ultrasound hearing aids for severely deaf individuals [37, 38, 39, 40, 41]. Even bone conduction headphones are being considered that exploit similar nonlinearities [42].

(3) Assorted Topics Related to *BackDoor*: A set of recent works bear some degree of relevance to *BackDoor*. Dhwani [1] explores in-air sound signals as a short range, ad-hoc data transfer modality. Chirp [18] and Zoosh [43, 19] have rolled out commercial products using sound for a secure data exchange medium. GhostTalk [44] explores various attack scenarios on the consumer electronics using high power electromagnetic interference. Another thread of recent work has looked into watermarking audio-visual media. Dolphin [45] enables speaker-microphone communication by embedding data bits on the sound. It adapts the signal parameters in real-time to keep the embedded signal imperceptible to human ears while achieving the 500 bps data rate. Kaleido [46] proposes a video precoding based solution to prevent videotaping an on-screen show in a theater or on a website. It precodes distortions in the video such that it is invisible to humans but severely distorts videotaping (due to specific limitations of the camera). Finally, sound maskers have also been used for protecting private conversation, however, these techniques have been limited to audible frequencies [47, 48, 49, 50]. *BackDoor* differs from the above in the sense that it exploits discrepancies between humans and electronics, ultimately enabling a new capability to the best of our knowledge.

2.8 Chapter Summary

Device nonlinearity has been conventionally viewed as a peril. Concepts presented in this chapter break away from this point of view and discover various opportunities to harness nonlinearity. By carefully designing ultrasound signals, we demonstrate that such signals remain inaudible to humans but are record-able by unmodified off-the-shelf microphones. This translates to new applications including inaudible data communication, privacy, and acoustic watermarking. While our ongoing work is focused on deeper understanding of these capabilities and applications, our longer-term goal is focused on generalization to other platforms, such as wireless radios and inertial sensors.

Chapter 3

Inaudible Voice Commands: Attack and Defense

3.1 Overview

A number of recent research studies have focused on the topic of inaudible voice commands [3, 52, 53]. *Backdoor* [3], discussed in Chapter 2, showed how hardware nonlinearities in microphones can be exploited, such that *inaudible ultrasound signals* can become audible to any microphone. DolphinAttack [52] developed on *Backdoor* to demonstrate that no software is needed at the microphone, i.e., a voice enabled device like Amazon Echo can be made to respond to inaudible voice commands. A similar research work independently emerged in arXiv [53], with a video demonstration of such an attack [54]. These attacks are becoming increasingly relevant, particularly with the proliferation of voice enabled devices including Amazon Echo, Google Home, Apple Home Pod, Samsung refrigerators, etc.

While creative and exciting, these attacks are still deficient on an important parameter: *range*. DolphinAttack can launch from a distance of 5 ft to Amazon Echo [52] while the attack in [53] achieves 10 ft by becoming partially audible. In attempting to enhance range, we realized strong tradeoffs with inaudibility, i.e., the output of the speaker no longer remains silent. This implies that currently known attacks are viable in short ranges, such as Alice's friend visiting Alice's home and silently attacking her Amazon Echo [55, 52]. However, the general, and perhaps more alarming attack, is the one in which the attacker parks the car on the road and controls voice-enabled devices in the neighborhood, and even a person standing next to the attacker does not hear it. This chapter elaborates on the attempt to achieve such an attack radius, followed by defenses against them. We formulate the core problem next and outline our intuitions and techniques for solving them.

Briefly, nonlinearity is a hardware property that makes high-frequency signals arriving at

This chapter revises the publication "Inaudible Voice Commands: The Long-Range Attack and Defense," in NSDI 2018 [51].

a microphone, say s_{hi}, get shifted to lower frequencies s_{low} (see Figure 3.1). If s_{hi} is designed carefully, then s_{low} can be almost identical to s_{hi} but shifted to within the audibility cutoff of 20 kHz inside the microphone. As a result, even though humans do not hear s_{hi}, nonlinearity in microphones produces s_{low}, which then become legitimate voice commands to devices like Amazon Echo. This is the root opportunity that empowers today's attacks.

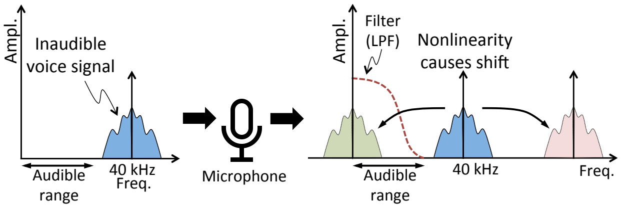


Figure 3.1: Hardware nonlinearity creates frequency shift. Voice commands transmitted over inaudible ultrasound frequencies get shifted into the lower audible bands after passing through the nonlinear microphone hardware.

Two important points need mention at this point. (1) Nonlinearity triggers at high frequencies and at high power – if s_{hi} is a soft signal, then the nonlinear effects do not surface. (2) Nonlinearity is fundamental to acoustic hardware and is equally present in speakers as in microphones. Thus, when s_{hi} is played through speakers, it will also undergo the frequency shift, producing an audible s_{low}. Dolphin and other attacks sidestep this problem by operating at low power, thereby forcing the output of the speaker to be almost inaudible. This inherently limits the range of the attack to 5 ft; any attempt to increase this range will result in audibility.

This chapter breaks away from the zero-sum game between range and audibility by an alternative transmitter design. Our core idea is to use multiple speakers, and stripe segments of the voice signal across them such that leakage from each speaker is narrowband, and confined to low frequencies. This still produces a garbled, audible sound. To achieve true inaudibility, we solve a min-max optimization problem on the length of the voice segments. The optimization picks the segment lengths in a way such that the aggregate leakage function is completely below the human auditory response curve (i.e., the minimum separation between the leakage and the human audibility curve is maximized). This ensures, by design, the attack is inaudible.

Defending against this class of nonlinearity attacks is not difficult if one were to assume hardware changes to the receiver (e.g., Amazon Echo or Google Home). An additional ultrasound microphone will suffice since it can detect the s_{hi} signals in air. However, with software changes alone, the problem becomes a question of forensics, i.e., can the shifted signal s_{low} be discriminated from the same legitimate voice command, s_{leg}? In other words, does nonlinearity leave an indelible trace on s_{low} that would otherwise not be present in s_{leg}?

Our defense relies on the observation that voice signals exhibit well-understood structure, composed of fundamental frequencies and harmonics. When this structure passes through nonlinearity, part of it remains preserved in the shifted and blended low-frequency signals. In contrast, the legitimate human voice projects almost no energy in these low-frequency bands. An attacker that injects distortion to hide the traces of voice, either pollutes the core voice command, or raises the energy floor in these bands. This forces the system into a zero-sum game, disallowing the attacker from erasing the traces of nonlinearity without raising suspicion.

Our measurements confirm the possibility to detect voice traces, i.e., even though nonlinearity superimposes many harmonics and noise signals on top of each other, and attenuates them significantly, cross-correlation still reveals the latent voice fingerprint. Of course, various intermediate steps of contour tracking, filtering, frequency-selective compensation, and phoneme correlation are necessary to extract out the evidence. Nonetheless, our final classifier is transparent and does not require any training at all, but succeeds for voice signals only, as opposed to the general class of inaudible microphone attacks (such as jamming [3]). We leave this broader problem to future work.

Our overall system, *BackDoor-II*, is built on multiple platforms. For the inaudible attack at long ranges, we have developed an ultrasound speaker array powered by our custom-made amplifier. The attacker types a command on the laptop, MATLAB converts the command to a voice signal, and the laptop sends this through our amplifier to the speaker. We demonstrate controlling Amazon Echo, iPhone Siri, and Samsung devices from a distance of 25 ft, limited by the power of our amplifier. For defense, we record signals from Android Samsung S6 phones, as well as from off-the-shelf microphone chips (popular in today's devices). We attack the system with various ultrasound commands, both from literature as well as our own. *BackDoor-II* demonstrates defense against all attacks with 97% precision and 98% recall. The performance remains robust across varying parameters, including multipath, power, attack location, and various signal manipulations.

In sum, our core contributions may be summarized as follows:

- A transmitter design that breaks away from the tradeoff between attack range and audibility. The core ideas pertain to carefully striping frequency bands across an array of speakers, such that individual speakers are silent but the microphone is activated.

- A defense that identifies human voice traces at very low frequencies (where such traces should not be present) and uses them to protect against attacks that attempt to erase or disturb these traces.

The subsequent sections elaborate on these ideas, beginning with some relevant background on nonlinearity, followed by threat model, attack design, and defense.

3.2 Background: Acoustic Nonlinearity

Microphones and speakers are in general designed to be linear systems, meaning that the output signals are linear combinations of the input. In the case of diaphragms and power amplifiers inside microphones and speakers, if the input sound signal is $s(t)$, then the output should ideally be:

$$s_{out}(t) = A_1 s(t)$$

where A_1 is the amplifier gain. In practice, however, acoustic components in microphones and speakers (like diaphragms, amplifiers, etc.) are linear only in the audible frequency range ($<$ 20 kHz). In ultrasound bands ($>$ 25 kHz), the responses exhibit nonlinearity [56, 57, 58, 59, 60]. Thus, for ultrasound signals, the output of the amplifier becomes:

$$\begin{aligned} s_{out}(t) = & \sum_{i=1}^{\infty} A_i s^i(t) = A_1 s(t) + A_2 s^2(t) + A_3 s^3(t) + ... \\ \approx & \; A_1 s(t) + A_2 s^2(t) \end{aligned} \tag{3.1}$$

Higher-order terms are typically extremely weak since $A_{4+} \ll A_3 \ll A_2$ and hence can be ignored.

Chapter 2 has shown ways to exploit this phenomenon, i.e., it is possible to play ultrasound signals that cannot be heard by humans but can be directly recorded by any microphone. Specifically, an ultrasound speaker can play two inaudible tones: $s_1(t) = \cos(2\pi f_1 t)$ at frequency $f_1 = 38$ kHz and $s_2 = \cos(2\pi f_2 t)$ at frequency $f_2 = 40$ kHz. Once the combined signal $s_{hi}(t) = s_1(t) + s_2(t)$ passes through the microphone's nonlinear hardware, the output becomes:

$$\begin{aligned} s_{out}(t) = & \; A_1 s_{hi}(t) + A_2 s_{hi}^2(t) \\ = & \; A_1(s_1(t) + s_2(t)) + A_2(s_1(t) + s_2(t))^2 \\ = & \; A_1 \cos(2\pi f_1 t) + A_1 \cos(2\pi f_2 t) \\ + & \; A_2 \cos^2(2\pi f_1 t) + A_2 \cos^2(2\pi f_2 t) \\ + & \; 2A_2 \cos(2\pi f_1 t) \cos(2\pi f_2 t) \end{aligned}$$

The above signal has frequency components at f_1, f_2, $2f_1$, $2f_2$, $f_2 + f_1$, and $f_2 - f_1$. This can be seen by expanding the equation:

$$\begin{aligned} s_{out}(t) = \;& A_1 \cos(2\pi f_1 t) + A_1 \cos(2\pi f_2 t) \\ + \;& A_2 + 0.5A_2 \cos(2\pi 2 f_1 t) + 0.5A_2 \cos(2\pi 2 f_2 t) \\ + \;& A_2 \cos(2\pi (f_1 + f_2) t) + A_2 \cos(2\pi (f_2 - f_1) t) \end{aligned}$$

Before digitizing and recording the signal, the microphone applies a low-pass filter to remove frequency components above the microphone's cutoff of 24 kHz. Observe that f_1, f_2, $2f_1$, $2f_2$, and $f_1 + f_2$ are all > 24 kHz. Hence, what remains (as an acceptable signal) is:

$$s_{low}(t) = A_2 + A_2 \cos(2\pi (f_2 - f_1) t) \tag{3.2}$$

This is essentially an $f_2 - f_1 = 2$ kHz tone which will be recorded by the microphone. However, this demonstrates the core opportunity, i.e., by sending a *completely inaudible signal*, we are able to generate an audible "copy" of it inside any unmodified off-the-shelf microphone.

3.3 Inaudible Voice Attack

We begin by explaining how the above nonlinearity can be exploited to send inaudible commands to *voice enabled devices* (VEDs) at a short range. We identify deficiencies in such an attack and then design the longer range, truly inaudible attack.

3.3.1 Short-Range Attack

Let $v(t)$ be a baseband voice signal that once decoded translates to the command: "Alexa, mute yourself". An attacker moves this baseband signal to a high frequency $f_{hi} = 40$ kHz (by modulating a carrier signal), and plays it through an ultrasound speaker. The attacker also plays a tone at $f_{hi} = 40$ kHz. The played signal is:

$$s_{hi}(t) = \cos(2\pi f_{hi} t) + v(t) \cos(2\pi f_{hi} t) \tag{3.3}$$

After this signal passes through the nonlinear hardware and low-pass filter of the microphone, the microphone will record:

$$s_{low}(t) = \frac{A_2}{2}\left(1 + v^2(t) + 2v(t)\right) \tag{3.4}$$

This shifted signal contains a strong component of $v(t)$ (due to more power in the speech components), and hence, gets decoded correctly by almost all microphones.

What happens to $v^2(t)$?
Figure 3.2 shows the power spectrum $V(f)$ corresponding to the voice command $v(t) =$"Alexa, mute yourself". Here the power spectrum corresponding to $v^2(t)$ which is equal to $V(f)*V(f)$ where $(*)$ is the convolution operation. Observe that the spectrum of the human voice is between $[50 - 8000]$ Hz and the relatively weak components of $v^2(t)$ line up underneath the voice frequencies after convolution. A component of $v^2(t)$ also falls at DC, however, degrades sharply. The overall weak presence of $v^2(t)$ leaves the $v(t)$ signal mostly unharmed, allowing VEDs to decode the command correctly.

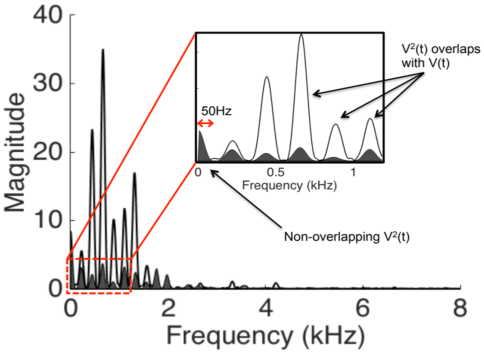


Figure 3.2: Spectrum of $V(f) * V(f)$ which is the nonlinear leakage after passing through the microphone.

However, to help $v(t)$ enter the microphone through the "nonlinear inlet", $s_{hi}(t)$ must be transmitted at sufficiently high power. Otherwise, $s_{low}(t)$ will be buried in noise (due to small A_2). *Unfortunately, increasing the transmit power at the speaker triggers nonlinearities at the speaker's own diaphragm and amplifier*, resulting in an audible $s_{low}(t)$ at the output of the speaker. Since $s_{low}(t)$ contains the voice command $v(t)$, the attack becomes audible. Past attacks sidestep this problem by operating at low power, thereby forcing the output of the speaker to be almost inaudible [61]. This inherently limits the radius of attack to a short range of 5 ft. Attempts to increase this range results in audibility, defeating the purpose of the attack.

Figure 3.3 confirms this with experiments in our building. Five volunteers visited marked locations and recorded their perceived loudness of the speaker's leakage. Clearly, speaker nonlinearity produces audibility, a key problem for long-range attacks.

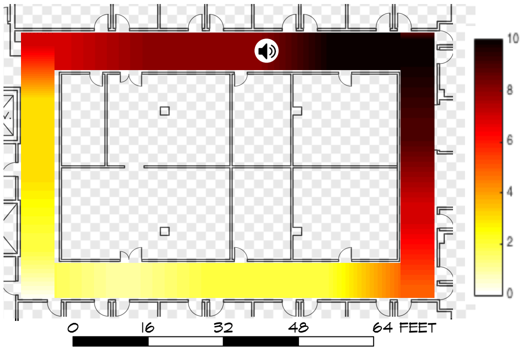


Figure 3.3: Heatmap showing locations at which $v(t)$ leakage from the speaker is audible.

3.3.2 Long-Range Attack

Before developing the long-range attack, we concisely present the assumptions and constraints on the attacker.

(1) Threat Model:
We assume that the attack scenario is the following.

- The attacker cannot enter the home to launch the attack, otherwise, the above short-range attack suffices.

- The attacker cannot leak any audible signals (even in a beamformed manner), otherwise such inaudible attacks are not needed in the first place.

- The attacker is resourceful in terms of hardware and energy (perhaps the attacking speaker can be carried in a car or placed at a balcony, pointed at VEDs in surrounding apartments or pedestrians).

- In case the receiver device (e.g., Google Home) is voice fingerprinted, we assume the attacker can synthesize the legitimate user's voice signal using known techniques [62, 63] to launch the attack.

- The attacker cannot estimate the precise *channel impulse response* (CIR) from its speaker to the voice enabled device (VED) that it intends to attack.

(2) Core Attack Method:
BackDoor-II develops a new speaker design that facilitates a considerably longer attack range, while eliminating the audible leakage at the speaker. Instead of using one ultrasound speaker, *BackDoor-II* uses multiple ultrasound speakers, physically separated in space. Then, *BackDoor-II* splices the spectrum of the voice command $V(f)$ into carefully selected segments and plays each segment on a different speaker, thereby limiting the leakage from each speaker.

(3) The Need for Multiple Speakers:
To better understand the motivation, let us first consider using two ultrasound speakers. Instead of playing $s_{hi}(t) = \cos(2\pi f_{hi}t) + v(t)\cos(2\pi f_{hi}t)$ on one speaker, we now play $s_1(t) = \cos(2\pi f_{hi}t)$ on the first speaker and $s_2(t) = v(t)\cos(2\pi f_{hi}t)$ on the second speaker where $f_{hi} = 40$ kHz. In this case, the two speakers will output:

$$\begin{aligned} s_{out1} = &\quad \cos(2\pi f_{hi}t) + \cos^2(2\pi f_{hi}t) \\ s_{out2} = &\quad v(t)\cos(2\pi f_{hi}t) + v^2(t)\cos^2(2\pi f_{hi}t) \end{aligned} \tag{3.5}$$

For simplicity, we ignore the terms A_1 and A_2 (since they do not affect our understanding of frequency components). Thus, when s_{out1} and s_{out2} emerge from the two speakers, human ears filter out all frequencies > 20 kHz. What remains audible is only:

$$\begin{aligned} s_{low1} = &\quad 1/2 \\ s_{low2} = &\quad v^2(t)/2 \end{aligned}$$

Observe that neither s_{low1} nor s_{low2} contains the voice signal $v(t)$, hence the actual attack command is no longer audible with two speakers. However, the microphone under attack will still receive the aggregate ultrasound signal from the two speakers, $s_{hi}(t) = s_1(t) + s_2(t)$, and its own nonlinearity will cause a "copy" of $v(t)$ to get shifted into the audible range (recall Equation 3.4). Thus, this two-speaker attack activates VEDs from greater distances, while the actual voice command remains inaudible to bystanders.

Although the voice signal $v(t)$ is inaudible, signal $v^2(t)$ still leaks and becomes audible (especially at higher power). This undermines the attack.

(4) Suppressing $v^2(t)$ Leakage:
To suppress the audibility of $v^2(t)$, *BackDoor-II* expands to N ultrasound speakers. It first partitions the audio spectrum $V(f)$ of the command signal $v(t)$, ranging from f_0 to f_N, into N frequency bins: $[f_0, f_1]$, $[f_1, f_2]$, $\cdots$, $[f_{N-1}, f_N]$ as shown in Figure 3.4. This can be achieved by computing an FFT of the signal $v(t)$ to obtain $V(f)$. $V(f)$ is then multiplied with a

rectangle function $rect(f_i, f_{i+1})$ which gives a filtered $V_{[f_i,f_{i+1}]}(f)$. An IFFT is then used to generate $v_{[f_i,f_{i+1}]}(t)$ which is multiplied by an ultrasound tone $\cos(2\pi f_{hi} t)$ and outputted on the i^{th} ultrasound speaker as shown in Figure 3.4.

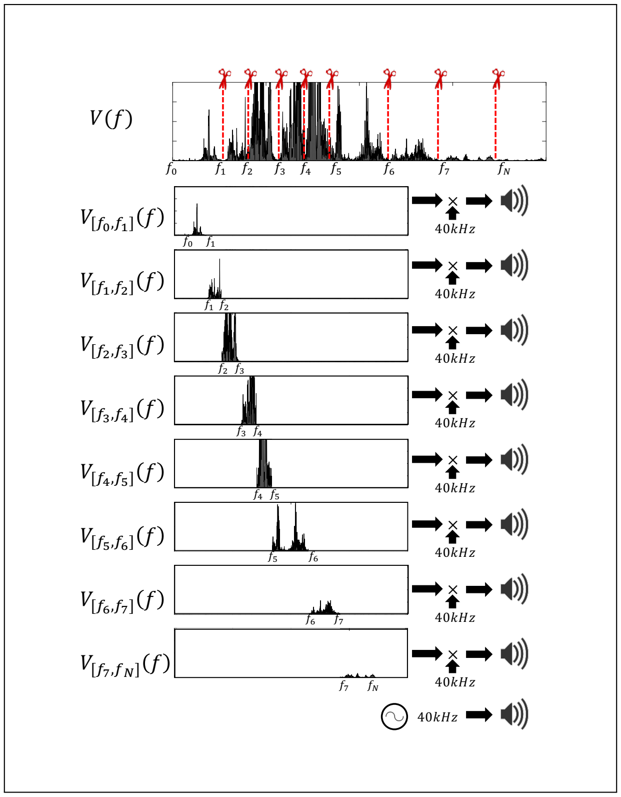


Figure 3.4: Spectrum splicing: optimally segmenting the voice command frequencies and playing it through separate speakers so that the net speaker-output is silent.

In this case, the audible leakage from i^{th} ultrasound speaker will be $s_{low,i}(t) = v^2_{[f_i,f_{i+1}]}(t)$. In the frequency domain, we can write this leakage as:

$$S_{low,i}(f) = \quad V_{[f_i,f_{i+1}]}(f) * V_{[f_i,f_{i+1}]}(f)$$

This leakage has two important properties:

$$\begin{aligned} &(1) \quad E\left[|S_{low,i}(f)|^2\right] \leq E\left[|V(f) * V(f)|^2\right] \\ &(2) \quad BW(S_{low,i}(f)) \leq BW(V(f) * V(f)) \end{aligned}$$

where $E[|.|^2]$ is the power of audible leakage and $BW(.)$ is the bandwidth of the audible leakage due to nonlinearities at each speaker. The above properties imply that splicing the spectrum into multiple speakers reduces the audible leakage from any given speaker. It also reduces the bandwidth and hence concentrates the audible leakage in a smaller band below

50 Hz.

While per-speaker leakage is smaller, they can still add up to become audible. The total leakage power can be written as:

$$L(f) = \left| \sum_{i=1}^{N} V_{[f_i, f_{i+1}]}(f) * V_{[f_i, f_{i+1}]}(f) \right|^2$$

To achieve true inaudibility, we need to ensure that the total leakage is not audible. To address this challenge, we leverage the fact that humans cannot hear the sound if the sound intensity falls below certain threshold, which is frequency dependent. This is known as the "Threshold of Hearing Curve", $T(f)$. Figure 3.5 shows $T(f)$ in dB as function of frequency. Any sound with intensity below the threshold of hearing will be inaudible.

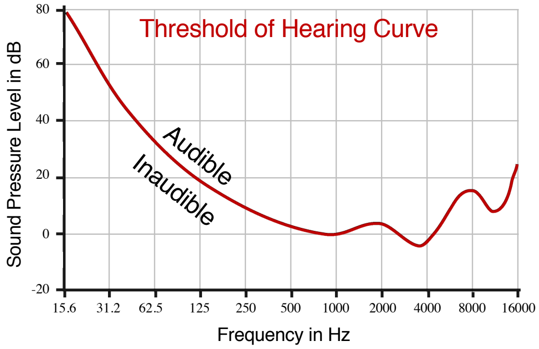


Figure 3.5: Threshold of hearing curve.

BackDoor-II aims to push the total leakage spectrum, $L(f)$, below the threshold of hearing curve $T(f)$. To this end, *BackDoor-II* finds the best partitioning of the spectrum, such that the leakage is below the threshold of hearing. If multiple partitions satisfy this constraint, *BackDoor-II* picks the one that has the largest gap from the threshold of hearing curve. Formally, we solve the below optimization problem:

$$\begin{aligned} &\underset{\{f_1, f_2, \cdots, f_{N-1}\}}{\text{maximize}} \ \min_{f} [T(f) - L(f)] \\ &\text{subject to} f_0 \leq f_1 \leq f_2 \leq \cdots \leq f_N \end{aligned} \tag{3.6}$$

The solution partitions the frequency spectrum to ensure that the leakage energy is below the hearing threshold for every frequency bin. This ensures inaudibility at any human ear.

(5) Increasing Attack Range:
It should be possible to increase attack range with more speakers, while also limiting audible leakage below the required hearing threshold. This holds in principle due to the following reason. For a desired attack range, say r, we can compute the minimum power density (i.e., power per frequency) necessary to invoke the VED. This power P_r needs to be high since the nonlinear channel will strongly attenuate it by the factor A_2. Now consider the worst case where a voice command has equal magnitude in all frequencies. Given each frequency needs power P_r and each speaker's output needs to be below *threshold of hearing* for all frequencies, we can run our *min-max optimization* for increasing values of N, where N is the number of speakers. The minimum N that gives a feasible solution is the answer. Of course, this is the upper bound; for a specific voice signal, N will be lower.

Increasing speakers can be viewed as beamforming the energy toward the VED. In the extreme case for example, every speaker will play one frequency tone, resulting in a strong DC component at the speaker's output which would still be inaudible. In practice, our experiments are bottlenecked by ADCs, amplifiers, speakers, etc., hence we will report results with an array of 61 small ultrasound speakers.

3.4 Defending Inaudible Voice Commands

Recognizing inaudible voice attacks is essentially a problem of acoustic forensics, i.e., detecting evidence of nonlinearity in the signal received at the microphone. Of course, we assume the attacker knows our defense techniques and hence will try to remove any such evidence. Thus, the core problem comes down to this question: *Is there any trace of nonlinearity that just cannot be removed or masked?*

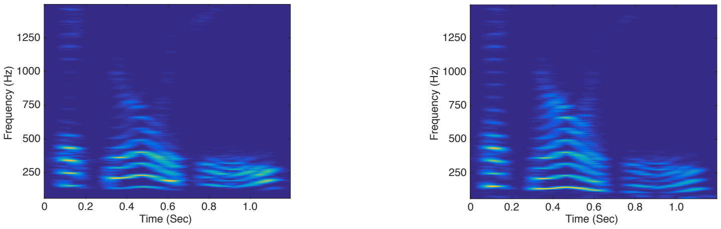
To quantify this possibility, let $v(t)$ denote a human voice command signal, say "Alexa, mute yourself". When a human issues this command, the recorded signal $s_{leg} = v(t) + n(t)$, where $n(t)$ is noise from the microphone. When an attacker plays this signal over ultrasound (to launch the nonlinear attack), the recorded signal s_{nl} is:

$$s_{nl} = \frac{A_2}{2}(1 + 2v(t) + v^2(t)) + n(t) \tag{3.7}$$

Figure 3.6 shows an example of s_{leg} and s_{nl}. Evidently, both are very similar, and both invoke the same response in VEDs (i.e., the text-to-speech converter outputs the same text for both s_{leg} and s_{nl}). A defense mechanism would need to examine any incoming signal s and tell if it is low-frequency legitimate or a shifted copy of the high-frequency attack.

3.4.1 Failed Defenses

Before we describe *BackDoor-II*'s defense, we mention a few other possible defenses which we have explored before converging on our final defense system. We concisely summarize four of these ideas.



(a) (b)

Figure 3.6: Spectrogram for s_{leg} and s_{nl} for voice command "Alexa, mute yourself".

(1) Decompose Incoming Signal $s(t)$:

One solution is to solve for $s(t) = \frac{A_2}{2}(1 + 2\hat{v}(t) + \hat{v}^2(t))$, and test if the resulting $\hat{v}(t)$ produces the same text-to-speech (T2S) output as $s(t)$. However, this proved to be a fallacious argument because, if such a $\hat{v}(t)$ exists, it will always produce the same T2S output as $s(t)$. This is because such a $\hat{v}(t)$ would be a cleaner version of the voice command (without the nonlinear component); if the polluted version s passes the T2S test, the cleaner version obviously will.

(2) Energy at Low Frequencies from $v^2(t)$:

Another solution is to extract portions of $s(t)$ from the lower frequencies – since regular voice signals do not contain sub-50 Hz components, energy detection should offer evidence. Unfortunately, environmental noise (e.g., fans, A/C machines, wind) leaves non-marginal residue in these low bands. Moreover, an attacker could deliberately reduce the power of its signal so that its leakage into sub-50 Hz is small. Our experiments showed non-marginal false positives in the presence of environmental sound and soft attack signals.

(3) Amplitude Degradation at Higher Frequencies:

The air absorbs ultrasound frequencies far more than voice (which translates to sharper reduction in amplitude as the ultrasound signal propagates). Measured across different microphones separated by $\approx$ 7.3 cm in Amazon Echo and Google Home, the amplitude difference should be far greater for ultrasound. We designed a defense that utilized the maximum amplitude slope between microphone pairs – this proved to be a robust discriminator between s_{leg} and s_{nl}. However, we were also able to point two (reasonably synchronized) ultrasound beams from opposite directions. This reduced the amplitude gradient, making it comparable to legitimate voice signals (Alexa treated the signals as multipath). In the real-world, we envisioned two attackers launching this attack by standing at two opposite sides of a house. Finally, this solution would require an array of microphones on the voice enabled device. Hence, it is inapplicable to one or two microphone systems (like phones, wearables, refrigerators).

(4) Phase-Based Separation of Speakers:
Given that long-range attacks need to use at least two speakers (to bypass speaker nonlinearity), we designed an angle-of-arrival (AoA) based technique to estimate the physical separation of speakers. In comparison to the human voice, the source separation consistently showed success, so long as the speakers are more than 2 cm apart. While practical attacks would certainly require multiple speakers, easily making them 2 cm apart, we aimed at solving the short-range attack as well (i.e., where the attack is launched from a single speaker). Put differently, the right evidence of nonlinearity should be one that is present regardless of the number of speakers used.

3.4.2 *BackDoor-II* Defense Design

Our final defense is to search for traces of $v^2(t)$ in sub-50 Hz. However, we now focus on exploiting the structure of human voice. The core observation is simple: voice signals exhibit well-understood patterns of fundamental frequencies, added to multiple higher-order harmonics (see Figure 3.6). We expect this structure to partly reflect in the sub-50 Hz band of $s(t)$ (that contains $v^2(t)$), and hence correlate with carefully extracted spectrum above-50 Hz (which contains the dominant $v(t)$). With appropriate signal scrubbing, we expect the correlation to emerge reliably, however, if the attacker attempts to disrupt correlation by injecting sub-50 Hz noise, the stronger energy in this low band should give away the attack. We intend to force the attacker into this zero-sum game.

(1) Key Question: Why Should $v^2(t)$ Correlate?
Figure 3.7(a) shows a simplified abstraction of a legitimate voice spectrum, with a narrow fundamental frequency band around f_j and harmonics at integer multiples nf_j. The lower bound on f_j is > 50 Hz [64]. Now recall that when this voice spectrum undergoes nonlinearity, each of f_j and nf_j will self-convolve to produce "copies" of themselves around DC (Figure 3.7(b)). Of course, the A_2 term from nonlinearity strongly attenuates this "copy".

However, given the fundamental band around f_j and the harmonics around nf_j are very similar in structure, each of ≈ 20 Hz bandwidth, the energy between $[0, 20]$ kHz superimposes.

This can be expressed as:

$$E_{[0,20]} \approx E\left[A_2 \sum_{n=1}^{N} \left|V_{[nf_j-20,\ nf_j+20]} * V_{[nf_j-20,\ nf_j+20]}\right|^2\right] \tag{3.8}$$

The net result is distinct traces of energy in sub-20 Hz bands, and importantly, this energy variation (over time) mimics that of f_j. For a legitimate attack, on the other hand, the sub-20 Hz is dominantly uncorrelated hardware and environmental noise.

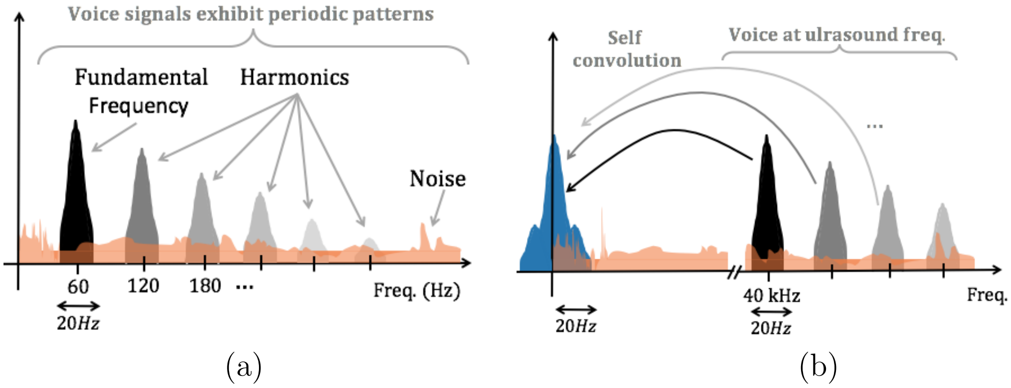


Figure 3.7: (a) A simplified voice spectrum showing the structure. (b) Voice spectra after nonlinear attack.

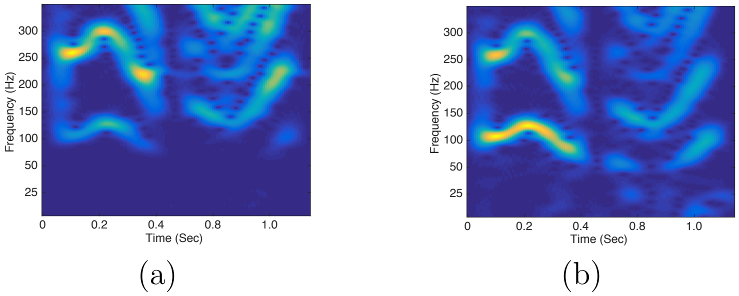


Figure 3.8: (a) Spectrogram of the audible voice. (b) Spectrogram of the inaudible attack voice. The attack signal contains higher power below 50 Hz, indicated by lighter color.

Figure 3.8(a) and (b) zoom into sub-50 Hz and compare the traces of energy for s_{leg} and s_{nl}, respectively. The s_{nl} signal clearly shows more energy concentration, particularly when the actual voice signal is strong. Figure 3.9 plots the power in the sub-50 Hz band with increasing voice loudness levels for both s_{leg} and s_{nl}. Note that loudness level is expressed in "dBSpl", where "Spl" denotes "sound pressure level", the standard metric for measuring sound. Evidently, nonlinearity shows increasing power due to the self-convolved spectrum overlapping in the lower band. Legitimate voice signals generate significantly less energy in these bands, thereby remaining flat for higher loudness.

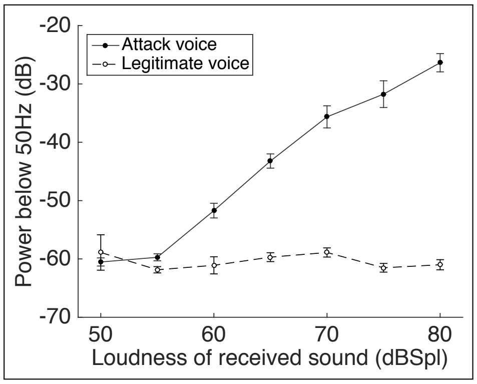


Figure 3.9: The loudness vs. sub-50 Hz band power plot.

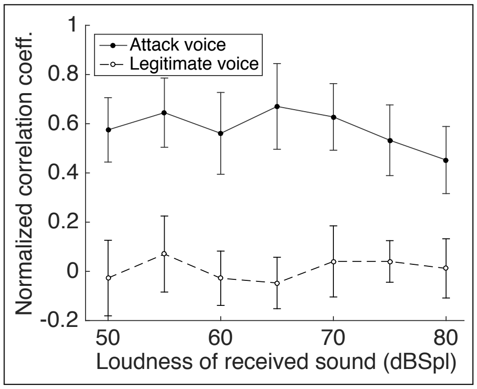


Figure 3.10: The loudness vs. correlation between P_{f_j} and $s_{<B}(t)$, denoting the power variation of the fundamental frequency and the sub 20 Hz band, respectively.

(2) Correlation Design:

The width of the fundamental frequencies and harmonics are time-varying, however, at any given time, if it is B Hz, then the self-convolved signal gets shifted into $[0, B]$ Hz as well. Note that this is independent of the actual values of center frequencies, f_j and nf_j. Now, let $s_{<B}(t)$ denote the sub-B Hz signal received by the microphone and $s_{>B}(t)$ be the signal above B Hz that contains the voice command. *BackDoor-II* seeks to correlate the energy variation over time in $s_{<B}(t)$ with the energy variation at the fundamental frequency, f_j in $s_{>B}(t)$. We track the fundamental frequency in $s_{>B}(t)$ using standard acoustic libraries, but then average the power around B Hz of this frequency. This produces a power profile over time, P_{f_j}. For $s_{<B}(t)$, we also track the average power over time. However, to avoid weak signals and disruption from noise, we remove time windows in which the power of f_j is

below its average. We stitch together the remaining windows from both P_{f_j} and $s_{<B}(t)$ and compute their correlation coefficient. We use an average value of $B = 20$ Hz.

Figure 3.10 shows the correlation for increasing loudness levels of the recorded signal (loudness below 60 dBSpl is not audible). The comparison is against a legitimate voice command. Evidently, we recorded a consistent correlation gap, implying that nonlinearity is leaving some trace in the low-frequency bands, and this trace preserves some structure of the actual voice signal. Of course, we have not yet accounted for the possibility that the attacker can inject noise to disrupt correlation.

(3) Improved Attack via Signal Shaping:

The natural question for the attacker is how to modify/add signals such that the correlation gap gets narrowed. Several possibilities arise:

(1) Signal $-v^2(t)$ can be added to the speaker in the low-frequency band and transmitted with the high-frequency ultrasound $v(t)$. Given that ultrasound will produce $v^2(t)$ after nonlinearity, and $-v^2(t)$ will remain as is, the two should interact at the microphone and cancel. Unfortunately, channels for low frequencies and ultrasound are different and unknown, hence it is almost impossible to design the precise $-v^2(t)$ signal. Of course, we will still attempt to attack with such a deliberately shaped signal.

(2) Assuming the ultrasound $v(t)$ has been up-converted to [40, 44] kHz, the attacker could potentially concatenate spurious frequencies from say [44, 46] kHz. These frequencies would also self-convolve and get "copied" around DC. This certainly affects correlation since these spurious frequencies would not correlate well (in fact, they can be designed to not correlate). The attacker's goal should be to the lower correlation while maintaining a low-energy footprint below 20 Hz.

The attacker can use the above approaches to try to defeat the zero-sum game. Figure 3.11 plots results from 4000 attempts to achieve low correlation and low energy. Of these, 3500 are random noises injected in legitimate voice commands, while the remaining 500 are more carefully designed distortions (such as frequency concatenation, phase distortions, low frequency injection, etc.). Of course, in all these cases, the distorted signal was still correct, i.e., the VED device responded as it should.

On the other hand, 450 different legitimate words were spoken by different humans (shown as hollow dots), at various loudness levels, accents, and styles. Clusters emerge suggesting the promise of separation. However, some commands were still too close, implying the need for a greater margin of separation.

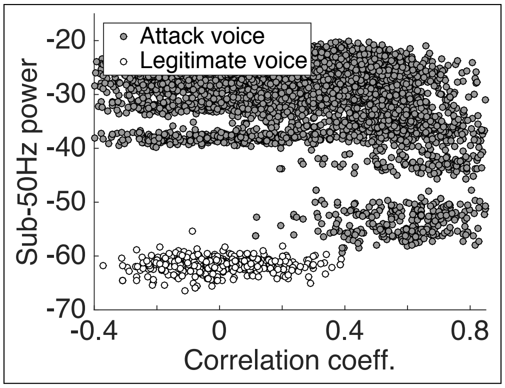
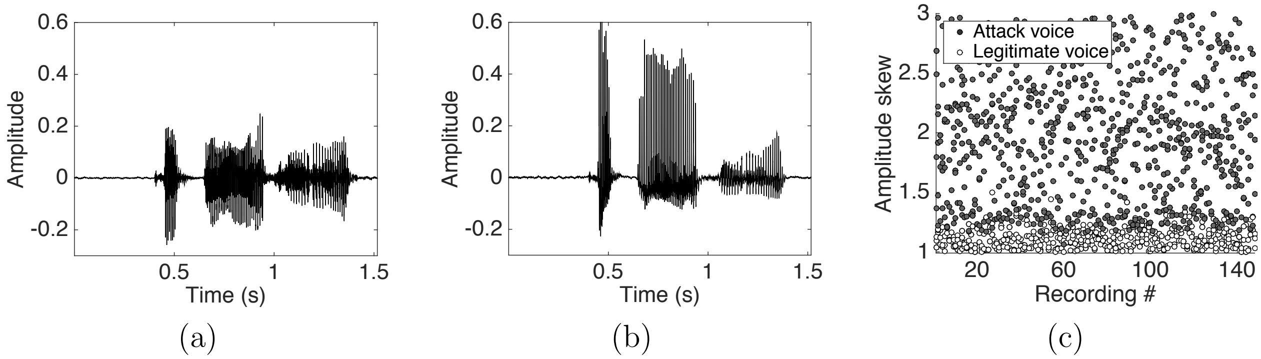


Figure 3.11: Zero-sum game between correlation and power at sub-50 Hz bands. Attacker attempts to reduce correlation by signal shaping or noise injection at sub-50 Hz band.



(a) (b) (c)

Figure 3.12: (a) Sound signals in time domain from s_{leg}. (b) Sound signals in time domain from s_{nl}, demonstrating a case of amplitude skew. (c) Amplitude skew for various attack and legitimate voice commands.

(4) Leveraging Amplitude Skew from $v^2(t)$:

In order to increase the separation margin, *BackDoor-II* leverages the amplitude skew resulting from $v^2(t)$. Specifically, two observations emerge: (1) When the harmonics in voice signals self-convolve to form $v^2(t)$, they fall at the same frequencies of the harmonics (since the gaps between the harmonics are quite homogeneous). (2) The signal $v^2(t)$ is a time domain signal with only positive amplitude. Combining these together, we postulated that amplitudes of the harmonics would be positively biased, especially for those that are strong (since $v^2(t)$ will be relatively stronger at that location). In contrast, amplitudes of legitimate voice signals should be well balanced on the positive and negative sides. Figure 3.12(a,b) shows one contrast between a legitimate voice s_{leg} and the recorded attack signal s_{nl}. In pursuit of this opportunity, we extract the ratio of the maximum and minimum amplitude (we average over the top 10% for robustness against outliers). Using this as the third dimension for separation, Figure 3.12(c) re-plots the s_{leg} and s_{nl} clusters. While the separation margin is close, combining it with correlation and power, the separation becomes satisfactory.

(5) *BackDoor-II*'s Elliptical Classifier:

BackDoor-II leverages three features to detect an attack: power in sub-50 Hz, correlation coefficient, and amplitude skew. Analyzing the *False Acceptance Rate* (FAR) and *False Rejection Rate* (FRR), as a function of these three parameters, we have converged on a ellipsoidal-based separation technique. To determine the optimal decision boundary, we compute FAR and FRR for each candidate ellipsoid. Our aim is to pick the parameters of the ellipse that minimize both FAR and FRR. Figure 3.13 plots the FAR and FRR as intersecting planes in a logarithmic scale. (Note that we show only two features since it is not possible to visualize the 4D graph.) The coordinate with minimum value along the canyon – indicating the *equal error rates* – gives the optimal selection of ellipsoid. Since it targets speech commands, this classifier is designed offline, one time, and need not be trained for each device or individual.

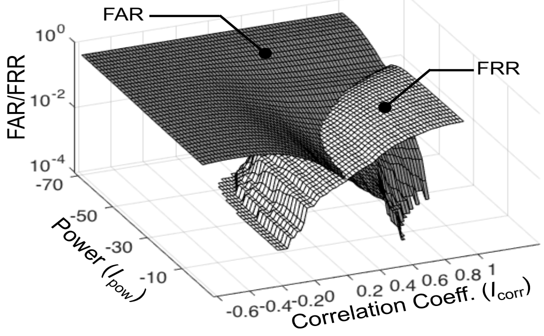


Figure 3.13: The False Acceptance Rate (FAR) plane (dark color) and the False Rejection Rate (FRR) plane (light color) for different sub-50 Hz power and correlation values.

3.5 Evaluation

We evaluate *BackDoor-II* on three main metrics: (1) attack range, (2) inaudibility of the attack, and the recorded sound quality (i.e., whether the attacker's command sounds human-like), and (3) accuracy of the defense under various environments. We summarize our findings as follows:

- We test our attack prototype, shown in Figure 3.14, with 984 commands to Amazon Echo and 200 commands to smartphones – the attacks are launched from various distances with 130 different background noises. Figure 3.15 shows attack success at 24 ft for Amazon Echo and 30 ft for smartphones at a power of 6 watt.

- We record 12 hours of microphone data – 5 hours of human voice commands and 7 hours of attack commands through ultrasound speakers. Figure 3.16(c) shows that attack words are recognized by VEDs with equal accuracy as legitimate human words. Figure 3.16(b) confirms that all attacks are inaudible, i.e., the leakage from our speaker array is 5-10 dB below the human hearing threshold.

- Figure 3.17(a) shows the precision and recall of our defense technique, as 98% and 99%, respectively, when the attacker does not manipulate the attack command. Importantly, precision and recall remain steady even under signal manipulation.

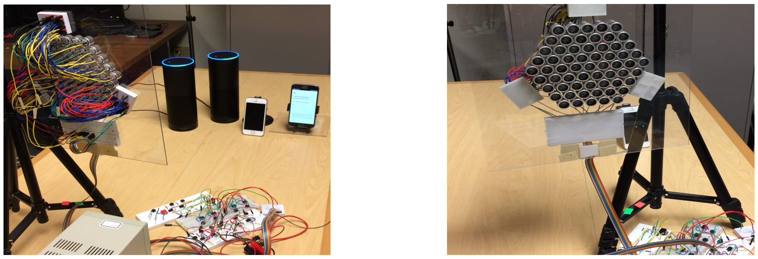
Before elaborating on these results, we first describe our evaluation platforms and methodology.

3.5.1 Platform and Methodology

(1) Attack Speakers: Figure 3.14(b) shows our custom-designed speaker system consisting of 61 ultrasonic piezoelectric speakers arranged as a hexagonal planar array. The elements of the array are internally connected in two separate clusters. A dual channel waveform generator (*Keysight 33500b series* [65]) drives the first cluster with the voice signal, modulated at the center frequency of 40 kHz. This cluster forms smaller sub-clusters to transmit separate segments of the spliced spectrum. The second cluster transmits the pure 40 kHz tone through each speaker. The signals are amplified to 30 volts using a custom-made NE5534AP op-amp based amplifier circuit. This prototype is limited to a maximum power of 6 watts because of the power ratings of the operational amplifiers. More powerful amplifiers are certainly available to a resourceful attacker.

(2) Target VEDs: We test our attack on three different VEDs – Amazon Echo, Samsung $S6$ smartphone running Android v7.0, and Siri on an iPhone $5S$ running iOS v10.3. Unlike Echo, Samsung S-voice and Siri requires personalization of the wake-word with user's voice – adding a layer of security through voice authentication. However, voice synthesis is known to be possible [62, 63], and we assume that the synthesized wake-word is already available to the attacker.

(3) Experiment Setup: We run our experiments in a lab space occupied by five members and also in an open corridor. We place the VEDs and the ultrasonic speaker at various distances ranging up to 30 ft. During each attack, we play varying degrees of interfering signals from six speakers scattered across the area, emulating natural home/office noises. The attack signals were designed by first collecting real human voice commands from ten different individuals; MATLAB is used to modulate them to ultrasound frequencies. For speech quality of the attack signals, we used the open-source *Sphinx4* speech processing tool [6].



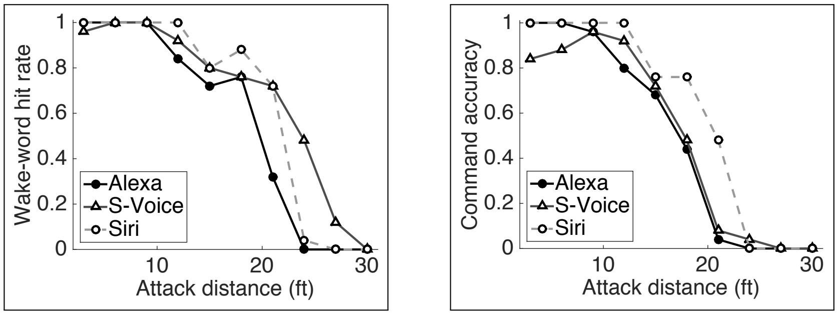
(a) (b)

Figure 3.14: *BackDoor-II* evaluation setup: (a) Ultrasonic speaker and voice enabled devices. (b) The ultrasonic speaker array for attack.

3.5.2 Attack Performance

(1) Activation Distance: This experiment attempts to activate the VEDs from various distances. We repeatedly play the inaudible wake-word from the ultrasound speaker system at regular intervals and count the fraction of successful activation. Figure 3.15(a) shows the activation hit rate against increasing distance – higher hit-rates indicate success with less number of attempts. The average distance achieved for 50% hit rate is 24 ft, while the maximum for Siri and Samsung S-voice are measured to be 27 and 30 ft respectively.

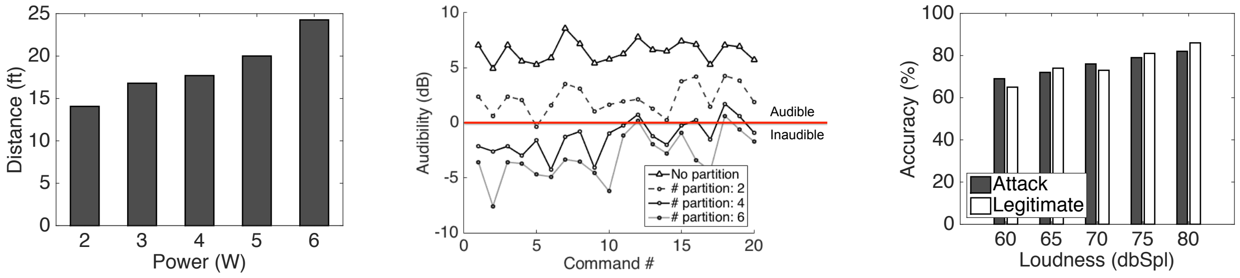
Figure 3.15(b) plots the attack range again, but for the entire voice command. We declare "success" if the text to speech translation produces every single word in the command. The range degrades slightly due to the stronger need to decode every word correctly.



(a) (b)

Figure 3.15: (a) The wake-word hit rate. (b) The command detection accuracy against increasing distances.

Figure 3.16(a) reports the attack range to Echo for increasing input power to the speaker system. As expected, the range continues to increase, limited by the power of our 6 watts amplifiers. More powerful amplifiers would certainly enhance the attack range, however, for the purposes of prototyping, we designed our hardware in the lower power regime.



(a) (b) (c)

Figure 3.16: (a) Maximum activation distance for different input power. (b) Worst-case audibility of the leakage sound after optimal spectrum partitioning. (c) Word recognition accuracy with automatic speech recognition software for attack and legitimate voices.

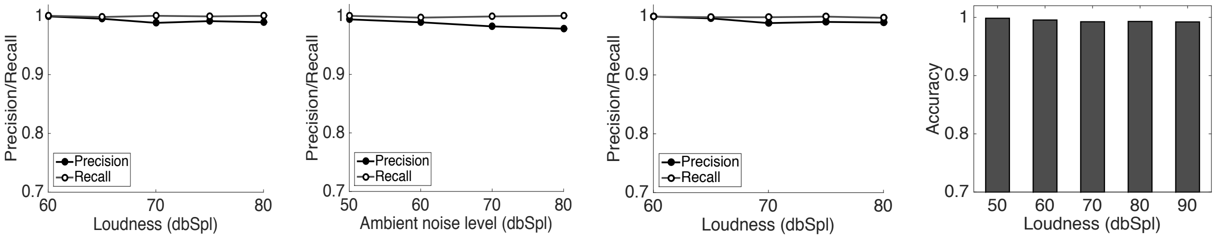
(2) Leakage Audibility: Figure 3.16(b) plots the efficacy of our spectrum splicing optimization, i.e., how effectively does *BackDoor-II* achieve speaker-side inaudibility for different ultrasound commands. Observe that without splicing (i.e., "no partition"), the ultrasound voice signal is almost 5 dB above the human hearing threshold. As the number of segments increase, audibility falls below the hearing curve. With 60 speakers in our array, we use six segments, each played through five speakers; the remaining 31 were used for the second $cos(2\pi f_c t)$ signal. Note that the graph plots the minimum gap between the hearing threshold and the audio playback, implying that this is a conservative worst-case analysis. Finally, we show results from 20 example attack commands – the other commands are below the threshold.

(3) Received Speech Quality: Given six speakers were transmitting each spliced segment of the voice command, we intend to understand if this distorts speech quality. Figure 3.16(c) plots the word recognition accuracy via Sphinx [6], an automatic speech recognition software. Evidently, *BackDoor-II*'s attack quality is comparable to human quality, implying that our multi-speaker beamforming preserves the speech's structure. In other words, speech quality is not the bottleneck for the attack range.

3.5.3 Defense Performance

(1) Metrics: Our defense technique essentially attempts to classify the attack scenarios distinctly from the legitimate voice commands. We report the "*Recall*" and "*Precision*" of this classifier for various sound pressure levels (measured in *dBSpl*), varying degrees of ambient sounds as interference, and deliberate signal manipulation. Recall that our metrics refer to: • *Precision:* What fraction of our detected attacks are correct?

• *Recall:* What fraction of the attacks did we detect? We now present the graphs beginning



(a) (b) (c) (d)

Figure 3.17: Precision and Recall of defense performance. (a) Basic performance without external interference. (b) Performance under ambient noise. (c) Performance under injected noise. (d) Overall accuracy across all experiments.

with the basic classification performance.

(2) Basic Attack Detection: Figure 3.17(a) shows the attack detection performance in a normal home environment without significant interference. The average precision and recall of the system is 99% across various loudness of the received voice. This result indicates the best-case performance of our system with minimum false alarm.

(3) Impact of Ambient Noise: In this section we test our defense system for common household sounds that can potentially mix with the received voice signal and change its features leading to misclassification. To this end, we played 130 noise sounds through multiple speakers while recording attack and legitimate voice signals with a smartphone. We replayed the noises at four different sound pressure levels starting from a typical value of 50 dBSpl to extremely loud 80 dBSpl, while the voice loudness is kept constant at 65 dBSpl. Figure 3.17(b) reports the precision and recall for this experiment. The recall remains close to 1 for all these noise levels, indicating that we do not miss attacks. However, at higher interference levels, the precision slightly degrades since the false detection rate increases a bit when noise levels are extremely high which is not common in practice.

(4) Impact of Injected Noise: Next, we test the defense performance against deliberate attempts to eliminate nonlinearity features from the attack signal. Here the attackers strategy is to eliminate the $v^2(t)$ correlation by injecting noise in the attack signal. We considered four different categories of noise – white Gaussian noise to raise the noise floor, band-limited noise on the sub-50 Hz region, water-filling noise power at low frequencies to mask the correlated power variations, and intermittent frequencies below 50 Hz. As shown, in Figure 3.17(c), the process does not significantly impact the performance because of the power-correlation tradeoff exploited by the defense classifier. Figure 3.17(d) shows that the overall accuracy of the system is also above 99% across all experiments.

3.6 Points of Discussion

We discuss several dimensions of improvement.

Lack of Formal Guarantee: We have not formally proved our defense. Although *BackDoor-II* is systematic and transparent (i.e., we understand why it should succeed) it still leaves the possibility that an attack may breach the defense. Our attempts to mathematically model the self-convolution and correlation did not succeed since frequency and phase responses for general voice commands were difficult to model, as were real-world noises. A deeper treatment is necessary, perhaps with help from speech experts who can model the phase variabilities in speech. We leave this to future work.

Generalizing to Any Signal: Our defense is designed for the class of voice signals, which applies well to inaudible voice attacks. A better defense should find the true trace of nonlinearity, not just for the special case of voice. This remains an open problem.

Is Air Nonlinear as Well? There is literature that claims air is also a nonlinear medium [28, 29, 30]. When excited by adequately powerful ultrasound signals, self-convolution occurs, ultimately making sounds audible. Authors in [2, 31] are designing *acoustic spotlighting systems* where the idea is to make ultrasound signals audible only along a direction. We have encountered traces of air nonlinearity, although in rare occasions. This certainly call for a separate treatment in the future.

Through-Wall Attack: Due to the limited maximum power (6 watts) of our amplifiers, we tested our system in non-blocking scenarios. If the target device is partially blocked (e.g. furnitures in the room blocking line-of-sight), the SNR reduces and our attack range will reduce. This level of power has not allowed us to launch through-wall attacks yet. We leave this to future work.

3.7 Related Work

(1) Attack on Voice Recognition Systems: Recent research [55, 66] shows that spoken words can be mangled such that they are unrecognizable to humans, yet decodable by voice recognition (VR) systems. *GVS-Attack* [67] exploits this by creating a smartphone app that gives adversarial commands to its voice assistant. More recently, *BackDoor* [3] has taken advantage of the microphone's nonlinearity to design ultrasonic sounds which are inaudible to humans, but becomes recordable inside the off-the-shelf microphones. The application includes preventing acoustic eavesdropping with inaudible jamming signals. As follow up, [52, 53] show that the principles of *BackDoor* can be used to send inaudible attack commands to a VED, but requires physical proximity to remain audible. *BackDoor-II* demonstrates the feasibility to increase the inaudible attack range, but more importantly, designs a defense against the inaudible attacks.

In the past, researchers use near-ultrasound [68, 69, 70, 71, 72, 73] and exploited aliasing to record inaudible sounds with microphones. A number of techniques use other sounds to camouflage audible signals in order to make it indistinguishable to human [74, 75, 76]. *CovertBand* [77] uses music to hide audible harmonic components at the speaker. *BackDoor-II*, on the other hand, uses high-frequency ultrasound as inaudible signals and leverages hardware nonlinearity to make them recordable to microphone.

(2) Speaker Linearization: A number of research [78, 79, 80] studies explore the possibility of adaptive linearization of general speakers. Through simulations, the authors have shown that by pre-processing the input signal, they can achieve as much as 27 dB reduction [80] of the nonlinear distortion in the noise-free case. Their techniques are not yet readily applicable to real speakers, since they have all assumed very weak nonlinearities, and over-simplified electrical and mechanical structures of speakers. With real speakers, especially ultrasonic piezoelectric speakers, it is difficult to fully characterize the parameters of the nonlinear model. Of course, if future techniques can fully characterize such models, our system can be made to achieve longer range with fewer speakers.

3.8 Chapter Summary

This chapter builds on the *BackDoor* signaling concept presented in Chapter 2 to show that inaudible voice commands are viable from distances of 25+ ft. Of course, careful design is necessary to ensure the attack is truly inaudible – small leakages from the attacker's speakers can raise suspicion, defeating the attack. This chapter also presents a defense against inaudible voice commands that exploit microphone nonlinearity. We show that nonlinearity leaves traces in the recorded voice signal, that are difficult to erase even with deliberate signal manipulation.

Chapter 4

Inaudible Communication through Vibrations

4.1 Overview

Data communication has been studied over a wide range of modalities, including radio frequency (RF), in-air and underwater acoustic, visible light, etc. This chapter envisions vibration as a new mode of communication. We explore the possibility of using *vibration motors*, present in all cell phones today, as a transmitter, while *accelerometers*, also popular in mobile devices, as a receiver. By carefully regulating the vibrations at the transmitter, and sensing them through accelerometers, two mobile devices should be able to communicate via physical touch.

We are not the first to recognize this opportunity. Acoustic communication operates on the same fundamental principles and has been studied for decades (over air [82, 83] and under water [84]). In recent years, authors in [85] identified the possibility of using vibra-motors and accelerometers in mobile phones, as an opportunity to exchange information. The benefits were identified as security and zero-configuration, meaning that the two devices need not discover each other's addresses to communicate. The act of physical contact would serve as the implicit address. However, authors identified the drawbacks of such a system to be low bit rates ($\sim$ 5 bits/s), based on the "Morse-code" style of ON/OFF communication with vibrations. Still, researchers conceived creative applications, including secure smartphone pairing and keyless access control [86].

This chapter is aimed at improving the data rates of vibratory communication, as well as its security features. We design *Ripple*, a system that breaks away from the intuitive *morse-code* style ON/OFF pulses and engages techniques such as orthogonal multi-carrier modulation, gray coding, adaptive calibration, vibration braking, side-channel suppression, etc. While

This chapter revises the publication "Ripple: Communicating through Physical Vibrations," in NSDI 2015 [81].

some techniques are borrowed from RF/acoustic communication, unique challenges (and opportunities) emerge from the vibra-motor/accelerometer platform, as well as from solid-materials on which they rest. For instance, the motor and the materials exhibit resonant frequencies that need to be adaptively suppressed; accelerometers sense vibration along three orthogonal axes, offering the opportunity to use them as parallel channels, with some degree of leakage. In addition to such techniques, we also design a receiver cradle – a wooden cantilever structure – that amplifies/dampens the vibrations in a desired way. A vibration-based product in the future, say a point-of-sale equipment for credit card transactions, may potentially benefit from such a design.

From a security perspective, *Ripple* recognizes the threat of acoustic leakage due to vibration, i.e., an eavesdropper could listen to the sound of vibration and decode the transmitted bits. To thwart such side channel attacks, we design the transmitter to also listen to the sounds and adaptively play a synchronized acoustic signal (through its speaker) to cancel the sound. The transmitter also superimposes a jamming sequence, ultimately offering inherent protection from acoustic eavesdroppers. We observe that application layer securities may not apply in all such scenarios – public/symmetric key-based encryption infrastructure may not scale to billions of phones and other use-cases such as the Internet of Things (IoT). Blocking access to the signal, at the physical layer itself, is desirable in these spontaneous, peer-to-peer, and perhaps disconnected situations [1].

It is natural to wonder what kind of applications will use vibratory communication, especially in light of NFC. We do not have a killer app to propose, and even believe that most applications would prefer NFC, mainly due to its higher data rates. (NFC uses 1.8 MHz bandwidth achieving more than 100 kbps, in contrast to 800 Hz with today's vibra-motors.) However, we conjecture that bringing the vibratory bit rates to a respectable level – say credit card transactions in a second – may trigger new ideas and use-cases. In particular, strict security-sensitive applications may be the candidates. Despite the very short communication range in NFC, recent results [87, 88] confirm that security threats are real. Authors decode NFC transmissions from 1 m away [89, 90, 91] and conjecture that high-gain beamforming antennas can further increase the separation. With the natural security benefits of touch-based communication (over RF), and supplemented with acoustic cancellation and jamming, we attempt to set a higher security bar for *Ripple*.

Moreover, the ubiquity of vibration motors in every cell phone, even in developing regions, presents an immediate market for vibratory communication. Peer-to-peer money exchange with recorded logs is a global problem, recently recognized by the Gates Foundation; hidden camera attacks on ATM kiosks have been rampant in India and South Asia [92]. Paying local cab drivers with phone-vibrations, or using phones as ATM cards can perhaps be of interest in developing countries. Clandestine operations may benefit where information need to be exchanged without leaving any trace in the wireless channel or in the Internet. Finally, if link capacity proves to be the only bottleneck, perhaps improved vibration motors can be included to mitigate it in the next phone models. While it is difficult to anticipate the needs of the future, we focus our attention on enabling and pushing forward this new modality of vibratory communication. To this end, our main contributions may be summarized as follows:

- Harnessing the vibration motor hardware and its functionalities, from a communication perspective.

- Developing an orthogonal multi-carrier communication stack using vibra-motor and accelerometer chips, and repeating the same for Samsung smartphones. Design decisions for the latter are different due to software/API limitations on smartphones, where vibra-motors were mainly integrated for simple alerts/notifications.

- Identifying acoustic side channel attacks and using signal cancellation and jamming to offer physical layer protection to eavesdropping.

4.2 Vibration Motors and Accelerometers

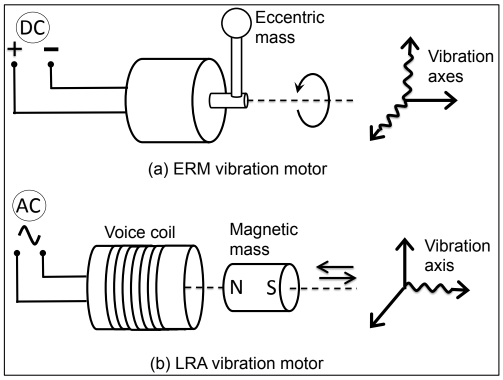
We begin with a high-level overview of vibration motors and accelerometers (substantial details in [93, 94, 95]).

4.2.1 Vibration Motor

A vibration motor (also called “vibra-motor”) is an electro-mechanical device that moves a metallic mass around a neutral position to generate vibrations. The motion is typically periodic and causes the center of mass (CoM) of the system to shift rhythmically. There are mainly two types of vibra-motors depending on their working principle:

(1) Eccentric Rotating Mass (ERM): This type of vibration generators uses a DC motor to rotate an eccentric mass around an axis as depicted in Figure 4.1. As the mass is not symmetric with respect to its axis of rotation, it causes the device to vibrate during the motion. Both the amplitude and frequency of vibration depend on the rotational speed of the motor, which can in turn be controlled through an input DC voltage. With increasing input voltages, both amplitude and frequency increase almost linearly and can be measured by an accelerometer.

(2) Linear Resonant Actuators (LRA): generate vibration by linear movement of a magnetic mass, as opposed to rotation in ERM (Figure 4.1). With LRA, the mass is attached to a permanent magnet which is suspended near a coil, called "voice coil". Upon applying AC current to the motor, the coil also behaves like a magnet (due to the generated electromagnetic field) and causes the mass to be attracted or repelled, depending on the direction of the current. This generates vibration at the same frequency as the input AC signal, while the amplitude of vibration is determined by the signal's peak-to-peak voltage. Thus LRAs allow for regulating both the magnitude and frequency of vibration separately. Fortunately, most mobile phones today use LRA vibra-motors.



(a) ERM vibration motor

(b) LRA vibration motor

Figure 4.1: Basics of ERM and LRA vibra-motors.

(3) PWM-based Motor Control: Ideally, a controller should be able to regulate the vibra-motor at fine granularities using any analog waveform. Unfortunately, microcontrollers produce digital voltage values limited to a few discrete levels. A popular technique to approximate analog signals with binary voltage levels is called Pulse Width Modulation (PWM) [96]. This technique is useful to drive analog devices with digital data without a digital-to-analog converter (DAC).

The core idea in PWM is to approximate any given voltage V by rapidly generating square pulses and configuring the pulse's duty cycle appropriately. For example, to create a 1 V signal with binary voltage levels of 5 V and 0 V, the duty cycle needs to be 20%. Now, if the period of the square pulse is made very small (i.e., high frequency), the effective output voltage will appear as 1 V. Toward this goal, the PWM frequency is typically set much higher than the response time of the target device so that the device experiences a continuous average voltage. Importantly, it is also possible to generate varying voltages with

PWM, say a sine wave, by gradually changing the duty cycles in a sinusoidal fashion.

4.2.2 Accelerometer

The accelerometer is a micro electro-mechanical (MEMS) device that measures acceleration caused by motion. While the inner workings of accelerometers can vary [97], the core working principle pertains to a movable seismic mass that responds to the vibration of the object it is attached to. Capacitive accelerometers, shown in Figure 4.2, are perhaps most popular in smartphones today. When vibrated, the seismic mass moves between fixed electrodes, causing differences in the capacitance c_1 and c_2, ultimately producing a voltage proportional to the experienced vibration.

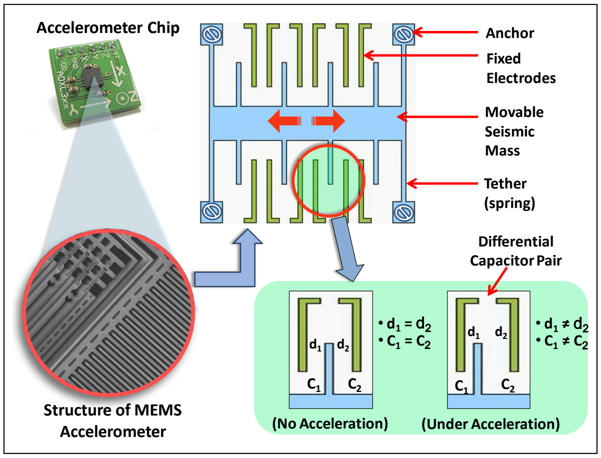


Figure 4.2: The internal architecture of MEMS accelerometer chip used in smartphones [98].

Sensing Acceleration: Modern accelerometers sense the movement of the seismic mass along three orthogonal axes, and report them as an $< X, Y, Z >$ tuple. The gravitational acceleration appears as a constant offset along the axis pointed toward the floor. The newest accelerometer chips support a wide range of adjustable sampling rates, typically from 100 mHz to 3.2 kHz. For the system explained in this chapter, we choose the ADXL345 [99] capacitive MEMS accelerometer, not only because it is used in most smartphones, but also because of programmability and frequency range.

4.3 Vibratory Transmission and Reception

Software/API limitations in smartphones prevent fully exploiting the vibra-motors and accelerometers. We design a custom hardware prototype using the same chips that smartphones use, and characterize/evaluate the system. We develop the constrained smartphone version in the next section.

4.3.1 Custom Hardware Setup

We control the vibra-motor and accelerometer through Arduino boards [100], an open source hardware development platform equipped with a ATmega328 8-bit RISC micro-controller [101]. Our first step is to precisely control the vibration frequency (and amplitude) through a time-varying sequence of voltage levels fed to the vibra-motor. Unfortunately, the micro-controller's output current fluctuates, leading to errors in the transmitted vibratory signals. Therefore, we power the vibra-motor with a standalone 6V DC power supply and use the Arduino micro-controller signal to operate a switch that regulates the voltage to the motor. We develop a simple circuit shown in Figure 4.3 – a NPN Darlington transistor (TIP122) serves as the switch and the controller signal goes to its base.

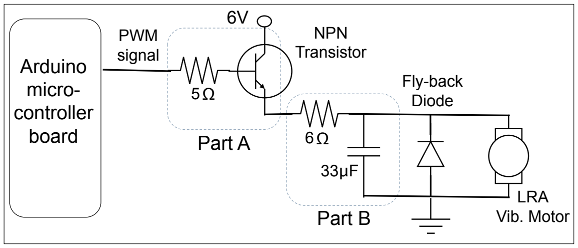
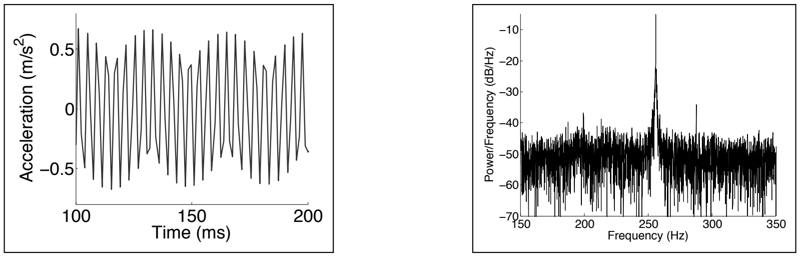


Figure 4.3: Transmitter hardware: the micro-controller controls a switch that regulates the 6V DC input.

Assume that we intend to regulate the vibra-motor in a sinusoidal fashion. We pre-load digital samples of the sine waveform into memory, and PWM uses them to determine the width of the square waves. When the sine wave frequency needs to be increased, the same digital samples need to be drawn at a faster rate and at precise timings. The switch uses the PWM output to regulate the 6V DC signal. We mitigate a number of engineering problems to run the set up correctly, including harmonic distortions due to the square pulses, spikes due to back EMF, etc. We move the PWM frequency to a high 32 kHz and use an RC filter (part B Figure 4.3) to remove the distortions; we use a 1N4001 fly-back diode to smooth out the spikes. We omit further details in the interest of space.

The accelerometer receiver is also controlled through Arduino via the I2C protocol [102] at 115200 baud rate. We set the accelerometer's sampling rate to 1600 Hz and 10-bit output resolution. While higher sampling rates are possible, we refrain from doing so since the micro-controller records the accelerometer data at a slower rate. In particular, the chip produces a sample per 0.625 ms, but the micro-controller takes around $8-12$ ms to periodically read and write in memory. We handle this with a FIFO mode of the accelerometer, such that the queued-up data is read in a burst. We also mount an on-board SD card to store data via the SPI protocol.

Figure 4.4 shows the accelerometer output when the vibra-motor is driven by the sinusoid input and made to touch the accelerometer. The final system functions correctly, and the platform is now ready for design and experimentation.



(a) (b)

Figure 4.4: (a) Accelerometer output in time domain. (b) Accelerometer output in frequency domain. Here the vibra-motor is fed with a 250 Hz sine wave.

4.3.2 Transmitter and Receiver Design

Ripple's design firmed up after multiple rounds of iterations. In the final version, the transmitter performs amplitude modulation on 10 different carrier signals uniformly spaced from 300 to 800 Hz – each carrier is modulated with a bandwidth of 40 Hz. Further, the vibrations are also parallelized on orthogonal motion dimensions (X and Z) with appropriate signal cancellation. The design details are presented next.

(1) Selecting the Carrier Signal:

To reason about how data bits should be transmitted, we first carry out an analysis of the available spectrum. This available spectrum is actually bottlenecked by the maximum sampling rate of the accelerometer receiver – since this rate is 1600 Hz, the highest frequency the transmitter can use is naturally 800 Hz. Now, to test the system's frequency response in the $[0, 800]$ band, we perform a "sine sweep" test. The transmitter, with the help of a waveform generator, produces continuously increasing frequencies from 1 Hz to 800 Hz with constant amplitude (the frequency increments are at 1 Hz). Figure 4.5 shows the corresponding vibration magnitudes recorded by the accelerometer. Evidently, the response is weak up to 60 Hz (called the "inert band"), followed by improvements until around 200

Hz, followed by a large spike at around 231 Hz. This spike is near the resonant frequency of the vibra-motor (confirmed in the data sheet).

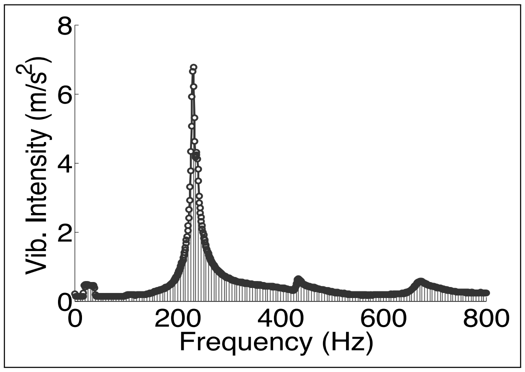


Figure 4.5: The vibra-motor's frequency response with the resonant frequency at around 231 Hz.

Intuitively, frequencies near the resonant band can serve as good carriers for amplitude modulated data because of a larger vibration range. However, when we plot the frequency versus time spectrogram of the sine sweep test (Figure 4.6), we find that the vigorous vibration around the resonant frequency spills energy in almost the entire spectrum. Therefore, transmitting on the resonant band can be effective for a single carrier system, but the interference ruins the opportunity to transmit data in parallel carriers. In light of this, we define a "resonant band" of 100 Hz around the peak, and move the carrier signals outside this band. We select 10 orthogonal carriers separated by 40 Hz from the non-resonant frequencies between 300 Hz and 800 Hz. The 40 Hz separation ensures the non-overlapping sidebands for the carriers, allowing reliable symbol recovery with software demodulation.

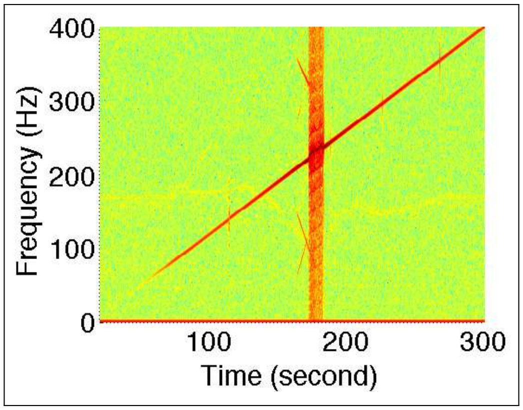


Figure 4.6: When excited with the resonant frequency, the vibra-motor spills energy across a wide frequency range.

(2) Synchronization:
Micro-controllers inject timing errors at various stages – variable delay in fetching digital samples from memory, during time-stamping the received samples, and due to oscillator/crystal frequency shifts with temperature. The timing errors manifest as fluctuations in vibration frequency, causing error in demodulation. To synchronize time between the transmitter and receiver, we introduce a pilot frequency at 70 Hz and transmit it in parallel to data bits. We choose 70 Hz to be above the inert band and lower than the resonant band. During reception, the receiver detects the pilot frequency, measures the offset in sampling rate, and interpolates the received signal by adjusting for this offset. Of course, this operation also corrects all other frequencies in the spectrum needed for demodulation.

(3) Modulating and Demodulating the Carrier Signal:
The carrier frequencies are modulated with Amplitude Shift Keying (ASK) in light of its bandwidth efficiency and simplicity over Frequency Shift Keying (FSK). We modulate each of the 10 carriers with binary data at a symbol rate of 20 Hz. To prevent inter-carrier interference, we shape the pulses with a raised cosine filter for each carrier individually; the modulated carriers are then combined and fed to the vibration motor transmitter. The receiver senses the energy in the pilot carrier, calibrates and synchronizes appropriately to identify the beginning of transmission. We again filter the received spectrum with (the same) raised cosine filter to isolate each carrier, and proceed to demodulate individual carriers separately. Figures 4.7(a) and (b) show a part of the spectrum before and after filtering, for an example carrier frequency at 405 Hz. The demodulation is performed with envelope detection and precise sampling at bit intervals. We will evaluate this custom-designed system later in Section 4.6 and show ∼200 bits/second data rates through vibration.

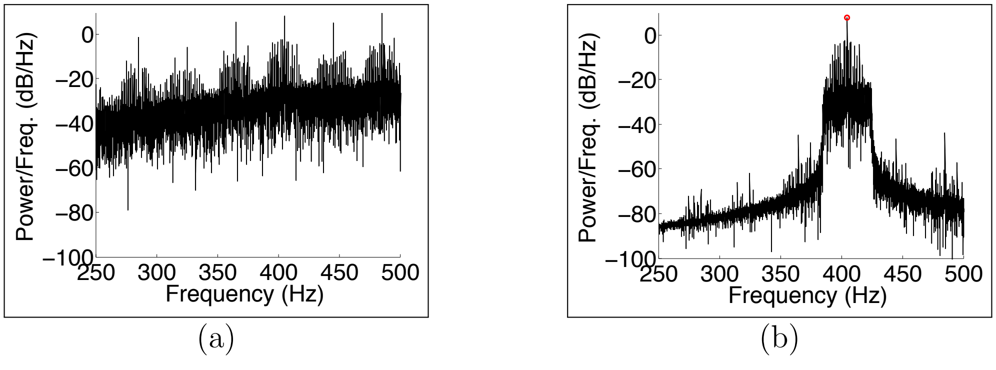


Figure 4.7: (a) The spectrum of the received signal. (b) The spectrum after filtering for a single carrier frequency at 405 Hz.

4.3.3 Orthogonal Vibration Dimensions

The above schemes, although adapted for vibra-motors, are grounded in the fundamentals of radio design. In an attempt to augment the bit rate, we observed that a unique property of accelerometers is its ability to detect vibration on three orthogonal dimensions (X, Y, and Z). Although vibra-motors only produce signals on a single dimension, perhaps multiple vibra-motors could be used in parallel. Unfortunately, due to some rigidity in our custom setup, accelerometer's motion along the X-axis is minimal, precluding it for communication. Therefore, we orient two vibra-motors in the Y and Z dimensions and execute the exact multi-carrier amplitude modulated transmissions discussed above.

Measurements show that vibration from one dimension spills into the other. However, rather interestingly, this spilled interference exhibits a 180° phase lag with respect to the original signal, as well as an attenuation in the amplitude. Figure 4.8 shows an example in which the Z-axis signal (solid black) has a spill on the Y-axis, with a reversed phase and halved amplitude. The vice versa situation also occurs. Now, to remove Z's spilled interference and decode the Y signal, we scale the Y signal so that the interference matches Z's actual amplitude, and then add it to the Z signal. The Z signal is removed quite precisely, leaving an amplified version of Y, which is then decoded through the envelope detector. The reverse is performed with Z's signal, resulting in a 2x improvement in data rate, evaluated later.

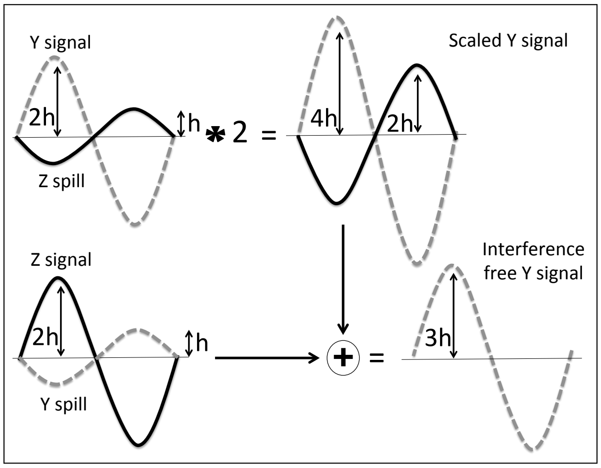


Figure 4.8: Orthogonal vibrations in X- and Z-axes.

4.4 Smartphone Prototype

This section shifts focus to vibratory communication on Android smartphones. Android is of interest since it offers APIs to a kernel-level PWM driver for controlling the ON/OFF timings. We develop a user space module that leverages third-party kernel space APIs [103] to control the vibration amplitudes as well. However, this still does not match the custom setup in the previous section. The PWM driver in Samsung smartphones is set to operate on the resonant band of the LRA vibra-motor, and the vibration frequency cannot be changed. This is understandable from the manufacturer's viewpoint, since vibra-motors are embedded to serve as a 1 bit alert to the user. However, for data communication, the nonlinear response at the resonant frequencies presents difficulties. Nonetheless, *Ripple* has to operate under these constraints and hence is limited to a single carrier frequency, modulated via amplitude modulation.

4.4.1 Smartphone Tx and Custom Rx

One advantage of the resonant frequency is that it offers a larger amplitude range, permitting n-ary symbols as opposed to binary (i.e., the amplitude range divided into n levels). To further amplify this range, we also design a custom smartphone cradle – a cantilever-based wooden bridge-like framework – that in contact with the phone amplifies specific vibration frequencies. While we will evaluate performance without this cradle, we were curious if (deliberately designed) auxiliary objects bring benefits to vibratory communication. Figure 4.9 shows the design – when the transmitter phone is placed on a specific location on this bridge, and the accelerometer connected to the other end, we indeed observe improved SNR. The key idea here is to make the "channel" resonate along with the smartphone to improve transmission capacity. We elaborate on the cantilever-based design next, followed by the communication techniques.

(1) Cantilever-based Receiver Setup:

Observe that every object has a *natural frequency* [104] in which it vibrates. If an object is struck by a rod, say, it will vibrate at its natural frequency no matter how hard it is struck. The magnitude of the strike will increase the amplitude of vibration, but not its frequency. However, if a periodic force is applied at the same natural frequency of the object, the object exhibits amplified vibration – resonance. In our setup, we use a 1 foot long wooden beam supported at one end, called a cantilever (Figure 4.9). The smartphone transmitter placed near the supported end, impinges a periodic force on the beam, calculated precisely based on the beam's resonant frequency (inversely proportional to $\sqrt{weight}$). We adjust the weight of the structure so that its natural frequency matches that of the phone's vibra-motor (which lies between 190 Hz to 250 Hz). This creates the desired resonance.

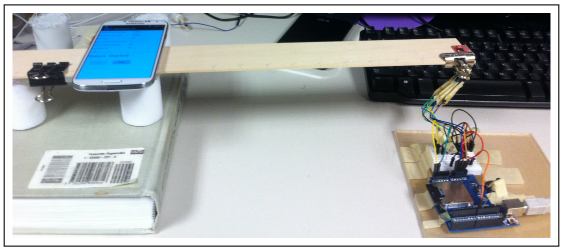


Figure 4.9: Cantilever-based receiver platform for vibration amplification.

The accelerometer is attached at the unsupported end of the beam. Figure 4.10 plots the measured amplitude variation (over three axes of the accelerometer) as the smartphone is placed on different positions on the beam. We choose the position located six inches from the supported end, as it induces maximal amplification on all three axes of the accelerometer.

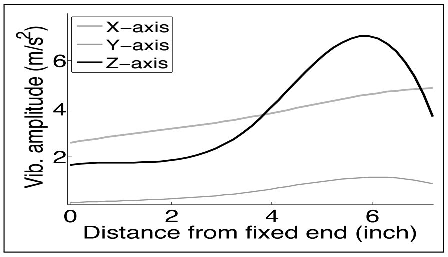
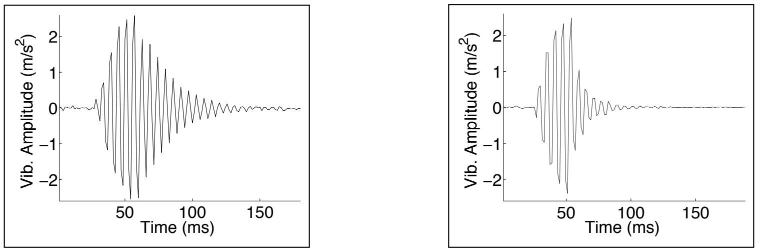


Figure 4.10: Vibration highest at a specific phone location.

(2) Symbol Duration and the Ringing Effect:

Ripple communicates through amplitude modulation – pulses of n-ary amplitudes (symbols) are modulated on the carrier frequency for a symbol duration. Ideally, the effect of a vibration should be completely limited within this symbol duration to avoid interference with the subsequent symbol (called inter-symbol interference). In practice, however, the vibration remains in the medium even after the driver stops the vibrator, known as the *ringing effect*. This is an outcome of inertia – the vibra-motor mass continues oscillating or rotating for some period after the driving voltage is turned off. Until this extended vibration dampens down substantially, the next symbol may get incorrectly demodulated (due to this heightened noise floor). Moreover, the free oscillation of the medium also contributes to ringing. Figure 4.11(a) shows a vibratory pulse of the smartphone, where the vibra-motor is activated from 20 to 50 ms. Importantly, the motor consumes 30 ms to overcome static inertia of the movable mass and reach its maximum vibration level. Once the voltage is turned off (at 50 ms) the vibration dampens slowly and consumes another 70 ms to become negligible.

This dictates the symbol duration to be around $30 + 70 = 100$ ms to avoid inter-symbol interference.



(a) (b)

Figure 4.11: (a) Ringing effect in the channel. (b) Reduced ringing using a braking voltage.

(3) Vibration Dampening:

To push for greater capacity, we attempt to reduce the symbol duration by dampening the ringing vibration. The core observation is that the ringing duration is a function of the amplitude of the signal – a higher amplitude signal rings for a longer duration. If, however, the amplitude can be deliberately curbed, ringing will still occur but will decay faster. Based on this intuition, we apply a small *braking-voltage* to the vibra-motor right after the signal has been sampled by the demodulator (30 ms). This voltage is deliberately small so that it does not manifest into large vibrations, and is applied for 10 ms. Once braking is turned off, we allow another 10 ms for the tail of the ringing to die down, and then transmit the next symbol. Thus the symbol duration is 50 ms now (half of the original) and there is still some vibration when we trigger the next symbol. While this adds slightly to the noise floor of the system, the benefits of a shorter symbol duration out-weighs the losses. Moreover, an advantage arises in energy consumption – triggering the vibra-motor from a cold start requires higher power. As we see later, activating it during the vibration tail saves energy.

(4) Modulation and Demodulation:

The modulation-demodulation technique is mostly similar to a single carrier of the custom hardware prototype. The only difference is that it uses multiple levels of vibration amplitudes (up to 16), unlike the binary levels earlier. Figure 4.12 shows how we can vary the voltage levels (as a percentage of maximum input voltage) to achieve different vibration amplitudes. If adequately stable, the amplitude at each voltage level can serve as separate symbols. Given the linear amplitude slope from voltage levels 15 to 90%, we divide this range into n-ary equi-spaced amplitude levels, each corresponding to a symbol. However, due to various placements and/or orientations of the phone, this slope can vary to some degree. While this does not affect up to 8-ary communication, 16 symbols are susceptible to this because of inadequate gaps between adjacent amplitude levels. To cope, we use a preamble of two symbols. At the beginning of each packet the transmitter sends two symbols with the highest and lowest amplitudes (15 and 90). The receiver computes the slope from these two symbols, and calibrates all the other intermediate amplitude levels from them. The receiver then decodes the bits with a maximum likelihood-based symbol detector.

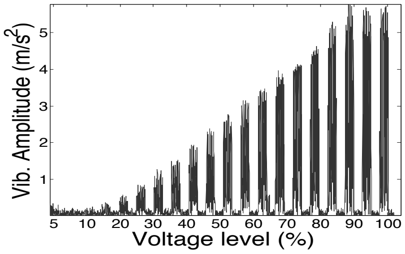


Figure 4.12: The change of vibration amplitude with the percentage of maximum input voltage.

4.5 Security

Vibrations produce sound and can leak information about the transmitted bits to an acoustic eavesdropper [105, 106, 107]. This section is aimed at designing techniques that thwart such side channel attacks. We design this as a real-time operation on the smartphone.

4.5.1 Acoustic Side Channel

The source of noise that actually leaks information is the rattling of the loosely attached parts of the motor – the unbalanced mass and metals supporting it. Our experiments show that this *sound of vibration* (SoV) exhibits correlations of ∼0.7 with the modulated frequency of the data transmission. Although SoV decays quickly with distance, microphone arrays and other techniques can be employed to still extract information. *Ripple* attempts to prevent such attacks.

4.5.2 Canceling Sounds of Vibration (SoV)

One way to defend against eavesdropping is to jam the acoustic channel with a *pseudorandom noise* sequence, thus decreasing the SNR of the SoV. Since this jamming signal will not interfere with physical vibrations, it does not affect throughput. Upon implementation, we realized that the jamming signal was audible, and annoying to the ears. The more effective approach is perhaps to cancel/suppress the SoV from the source, and then jam faintly, to camouflage the residue.

Ideally, *Ripple* should produce an “anti-noise” signal that cancels out the SoV to ultimately create silence. The transmitter (and not the receiver) should generate this anti-noise since it knows the exact bit sequence that is the source of the SoV. Of course, acoustic noise cancellation is a well-studied area – several headphones today use a microphone to capture ambient sounds and blends a negative version of it through the headphone speakers. The challenge of course is in detecting the ambient sound in real-time and producing the precise negative (phase shifted) signals. However, unlike *Ripple*, headphones need to cancel the ambient noise only at the human ear, and not at all other locations around the human.

With *Ripple*, the problem is easier in the sense that the transmitter exactly knows the bit sequence that is causing the SoV. This can help in modeling the sound waveform ahead in time, and can potentially be synchronized. The issue, however, is that the SoV varies based on the material medium on which the phone is placed; also the SoV needs to be cancelled at all locations in the surrounding area. Further, the phase of the SoV remains unpredictable as it depends on the starting position of the mass in the vibra-motor and the delay to attain the full swing. Finally, Android offers little support for real-time audio processing [108], posing a challenge to develop SoV cancellation on off-the-shelf phones.

4.5.3 Ripple Cancel and Jam

The overall technique is composed of three sub-tasks: anti-noise modeling, phase alignment, and jamming.

(1) Anti-noise Modeling: The core challenge is to model the analog SoV waveform corresponding to the data bits that will be transmitted through vibration. Since the motor’s vibration amplitude and frequency are known (i.e., the carrier frequency), the first approximation of this model is simple to create. However, as mentioned earlier, the difficulty arises in not knowing how the unknown material (on which the phone is placed) will impact the SoV. Apart from the fundamental vibration frequency, the precise SoV signal depends also on the strength and count of the *overtones* produced by the material. To estimate this, the *Ripple* transmitter first transmits a short “preamble”, listens to its FFT, and picks the *top-K* strongest overtones. These overtones are combined in the revised signal model. Finally, the actual data bits are modeled in the time domain, reversed in sign, and added to create the final “anti-noise” signal. This is ready to be played on the speaker, except that the phase of anti-noise needs to precisely match the SoV.

(2) Phase Alignment with Frequency Switch: Unfortunately, Android introduces a variable latency of up to 10 ms to dispatch the audio data to the hardware. This is excessive since a 2.5 ms lag can cause constructive interference between the anti-noise and the SoV. Fortunately, two observations help in this setting: (1) the audio continues playing at the specified sample rate without any significant fluctuation, and (2) the sample rate of the active audio stream can be changed in real-time. Thus, we can now control the frequency of the online audio by changing the playback sample rate.

We leverage this frequency control to match the phase of anti-noise with the SoV. The key idea is to start the anti-noise as close as possible to the SoV, but increase the sampling frequency such that the fundamental frequency of the anti-noise increase by δf. When this anti-noise combines in the air with the SoV, it creates the amplitude of the sound to vary because of the small difference in the fundamental frequencies. Obviously, the maximum suppression of the SoV occurs when the amplitude of this combined signal is at its minimum. The phase difference between the SoV and anti-noise is almost matched at this point. At exactly this "phase-lock" time, *Ripple* switches the fundamental frequency of the anti-noise to its original value (i.e., lower by δf). It recognizes this time instant by tracking the envelope of the combined signal and switching frequencies at the minimum point on the envelope. Figure 4.13 illustrates the various steps leading up to the frequency switch, and the sharp drop in signal amplitude. The suppressed signal remains at that level thereafter.

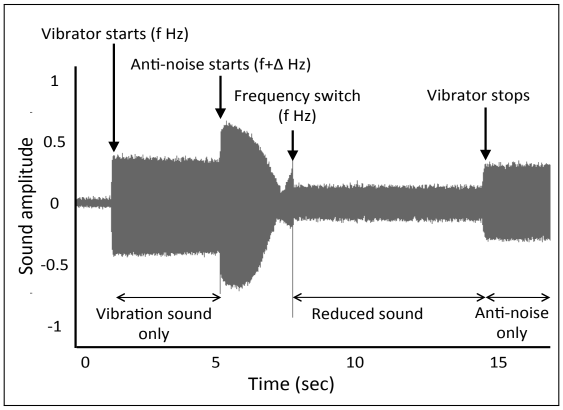
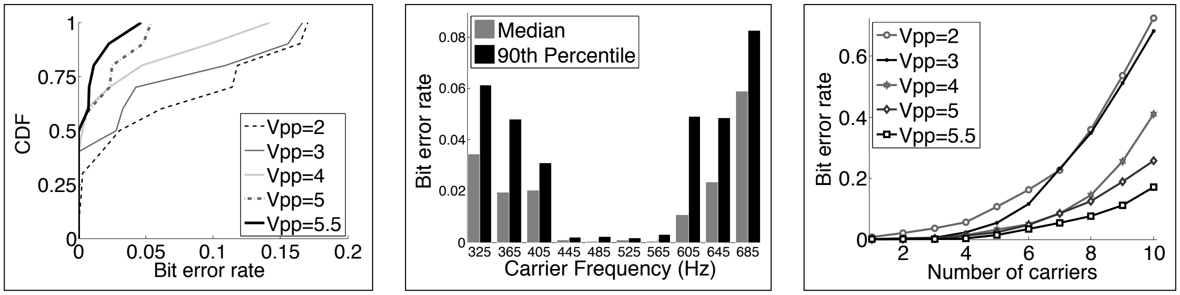


Figure 4.13: The anti-noise partially cancels the SoV, however, some mismatches result in some residual signal.

(3) Jamming: The cancellation is not perfect because the timing of the operations are not instantaneous; microphone and speaker noise also pollute the anti-noise waveforms, leaving a small residue. To prevent attacks on this residue, *Ripple* superimposes a jamming signal – the goal is to camouflage the sound residue. Conceptually it is simple, since a



(a) (b) (c)

Figure 4.14: (a) BER as a function of the input signal peak-to-peak voltage (Vpp). Overall data rate ∼200 b/s. (b) Per-carrier BER across 10 frequencies. (c) BER as a function of the number of carriers used (each carrier bit rate = 20 bits/s).

pseudorandom noise sequence can be added to the anti-noise waveform once it has phase-locked with the vibration sound. Unfortunately, Android does not allow loading a second signal on top of a signal that is already playing. Note that if we load the jamming signal upfront (along with the modeled anti-noise signal), the precise phase estimation will fail. We develop an engineering work-around. When modeling the anti-noise waveform, we also add the jamming noise sequence, but pre-pad the latter with a few *zeros*. Thus, when the SoV and anti-noise combine, the zeros still offer opportunities for detecting the time when the signals precisely cancel. We phase-lock at these times and the outcome is the residual signal from imperfect cancellation, plus the jamming sequence. We will show in the evaluation how the SoV's SNR degrades due to such cancellation and jamming, offering good protection to eavesdropping. Of course, the tradeoff is that we need a longer preamble now for this phase alignment process. However, this is only an issue arising from current Android APIs.

4.6 Evaluation

We evaluate *Ripple* in three phases – the custom hardware, the smartphone prototype, and security.

4.6.1 Custom Hardware

(1) Bit Error Rate (BER):

Recall that the custom hardware is composed of vibra-motors and accelerometer chips controlled by Arduino boards. We bring the two devices in contact and initiate packet transmission of various lengths (consuming between 1 to 10 seconds). Each packet contains pseudo-random binary bits at 20 Hz symbol rate on 10 parallel carriers. The bits are demodulated at the receiver and compared against the ground truth. We repeat the experiment for increasing signal energy (i.e., by varying the peak-to-peak signal voltage, Vpp, from 1V to 5V). Figure 4.14(a) plots the BER as a function of peak-to-peak input voltage (Vpp) to the vibra-motor and demonstrates how it diminishes with higher SNR. At the highest SNR, and aggregated over all carrier frequencies, *Ripple* achieves the 80^{th} percentile BER of 0.017 translating to an average bit rate of 196.6 bits/s.

(2) Behavior of Carriers:

In evaluating BERs across different carrier frequencies, we observe that not all carriers behave similarly. Figure 4.14(b) shows that carrier frequencies near the center of the spectrum perform consistently better than those near the edges. One of the reasons is *aliasing noise*. Ideally, the accelerometer should low-pass-filter the signal before sampling, to remove signal components higher than the Nyquist frequency. However, inexpensive accelerometers do not employ anti-aliasing filters, causing such undesirable effects. Carriers near the resonant band also experience higher noise due to the spilled-over energy.

Increasing the number of carriers will enable greater parallelism (bit rate), at the expense of higher BER per carrier. To characterize this tradeoff, we transmit data on increasing number of carriers, starting from the middle of our spectrum and activating carriers on both sides, one at a time. Figure 4.14(c) shows BER variations with increasing number of carriers, for varying signal energy (peak-to-peak voltage, Vpp). As each carrier operates at fixed 20 Hz symbol rate, this also shows the bit rate vs BER characteristics of our system. Figure 4.15 zooms on the best four carriers.

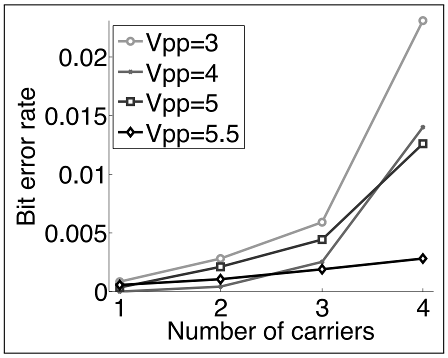


Figure 4.15: BER vs. number of carriers (four carriers shown).

(3) Temporal Stability:

Given that vibra-motors and accelerometers are essentially mechanical systems, we intend to evaluate their properties when they are made to operate continuously for long durations.

Given the low bit rates, this might be the case when relatively longer packets need to be transmitted. Toward this end, we continuously transmit data for 50 sessions of 300 seconds each. Figure 4.16 plots the per-carrier BER (computed in the granularity of 10 second periods) of a randomly selected session – the Y-axis shows each of the carriers and the X-axis is time. The BERs vary between 0.02 near the center to 0.2 near the edge. Overall results, omitted for the interest of space, show no visible degradation in BER even after running for 300 seconds.

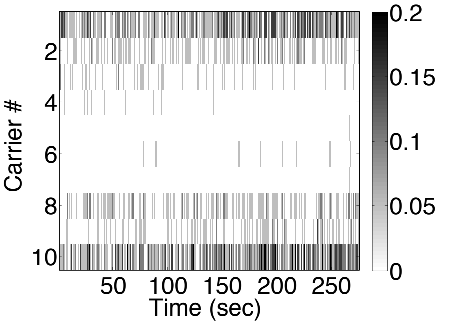
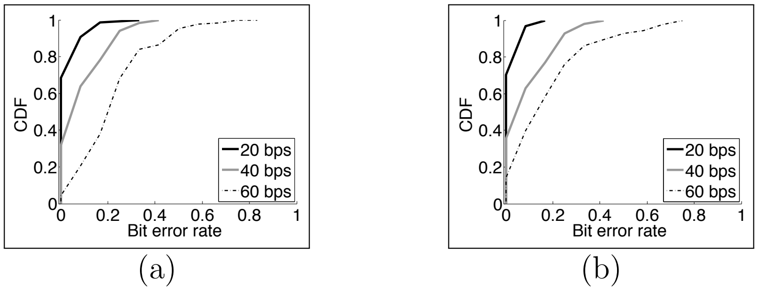


Figure 4.16: The BER per-carrier does not degrade after the motor is run for long durations.

(4) Exploiting Vibration Dimensions:

Recall that *Ripple* used two vibra-motors in parallel to exploit the orthogonality of vibrations along the Y- and Z-axes of the accelerometer. Figure 4.17(a) and (b) show the distribution of BER achieved across carrier frequencies on the Y- and Z-axes, respectively. We also attempt to push the limits by modulating greater than 20 bits/s, however, the BER begins to degrade. In light of this, *Ripple* achieves median capacity of around 400 bits/s (i.e., 20 bits/s per carrier x 10 carriers x 2 dimensions). While the tail of the BER distribution still needs improvement, we believe coding can be employed to mitigate some of it.



(a)

(b)

Figure 4.17: BER per carrier for parallel transmissions on orthogonal dimensions: (a) Y-axis. (b) Z-axis.

4.6.2 Smartphone Prototype

(1) Calibration:

Vibrations will vary across transactions due to phone orientation, humans holding it, different vibration medium, etc. As discussed earlier, the demodulator calibrates for these factors, but pays a penalty whenever the calibration is imperfect. We evaluate accuracy of calibration using the error between the estimated amplitude for a symbol, and the mean amplitude computed across all received symbols. Figure 4.18 plots the normalized error for various n-ary modulations – the normalization denominator is used as the difference between adjacent amplitudes.

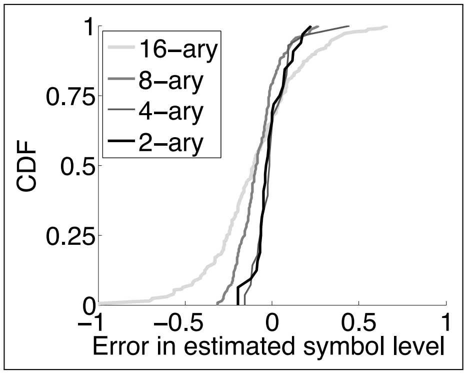
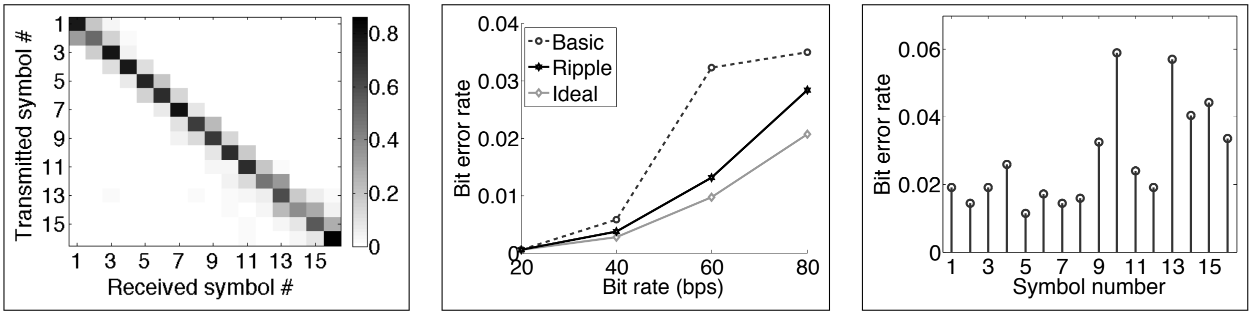


Figure 4.18: CDF of estimated symbol level error as a fraction of the mean inter-symbol difference.



(a) (b) (c)

Figure 4.19: (a) This heat-map shows the confusion matrix of the transmitted and received symbols. (b) *Ripple*'s BER compared to the Basic, Ideal. (c) Per symbol BER with 16-ary communication.

(2) BER with Smartphones:

Figure 4.19(a) plots the confusion matrix of transmitted and received (or demodulated) symbols, for 16-ary modulation. While some errors occur, we observe that they are often the symbol adjacent to the one transmitted. In light of this, *Ripple* uses Gray codes to minimize such well-behaved errors. With these codes and calibration, Figure 4.19(b) shows the estimated BER for different bit rates, for each of the four modulation schemes. As comparison points, the “Basic” symbol detector uses predefined thresholds for each symbol and maps the received sample to the nearest amplitude. The “Ideal” scheme identifies the bits using the knowledge of all received symbols. *Ripple*’s performs well even at higher bit rates, which is not the case with Basic.

Figure 4.19(c) shows the BER per symbol for 16-ary modulation, showing that symbols corresponding to the high vibration amplitudes experience higher errors. The reason is that the consistency of the vibration motor degrades at high amplitudes – we have verified this carefully by observing the distribution of received vibration amplitudes for large data traces.

(3) Impact of Phone Orientation:

The LRA vibra-motor inside Galaxy S4 generates linear vibration along one dimension – the teardown of the phone [109] shows the motor’s axis aligned with the Z-axis of the phone. Thus, an accelerometer should mostly witness vibration along the Z-axis. The other two axes do not exhibit sufficient vibration at higher bit rates. This is verified in Table 4.1 where the first four data points are from when the phone is laid flat on top of the cantilever. However, once the phones are made to stand vertically or on the sides, its X- and Y-axes align with the accelerometers Z-axis, causing an increase in errors. This suggests that the best contact points for the phones are their XY planes, mainly due to the orientation the vibration motor.

Table 4.1: BER with 16-ary modulation for various orientations.

Orientation	Hor. A	Hor. B	Hor. C	Hor. D	Ver. A	Ver. B
Mean BER	0.025	0.029	0.002	0.029	0.197	0.178

(4) Phone Held in Hand (No Cantilever):

We experiment a scenario in which the accelerometer-based receiver is on the table, and the handheld phone is made to touch the top of the receiver. The alignment is crudely along the Z-axis. This setup adversely affects the system by (1) eliminating the amplitude gain due to the cantilever, and (2) the dampens vibration due to the hand’s absorption. Figure 4.20 shows the results – unsurprisingly, the total vibration range is now smaller, pushing adjacent symbol levels to be closer to each other, resulting in higher BER.

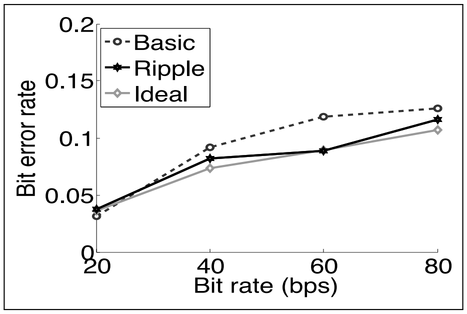
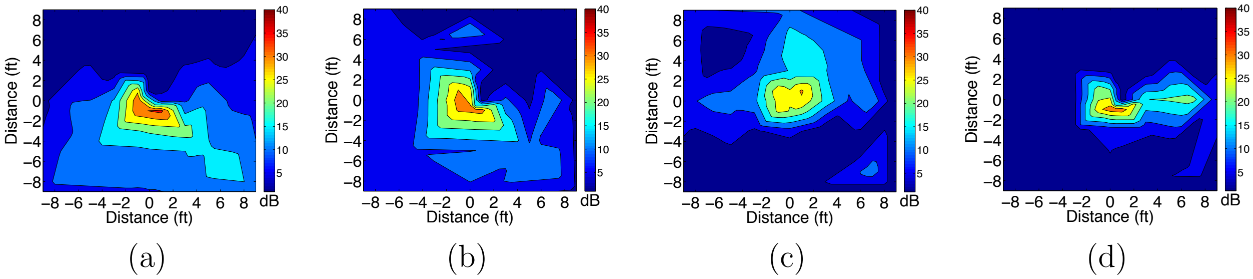


Figure 4.20: The BER with a hand-held phone.

4.6.3 Security

(1) Acoustic Signal Leakage:

To characterize the maximum acoustic leakage from vibrations, we run the vibra-motor at its highest intensity and record the SoV at various distances, using smartphone microphones sampled at 16 kHz. This leakage is naturally far higher than a typical vibratory transmission (composed of various intensity levels), so mitigating the most severe leakage is stronger security. We also realize that the material on which the smartphone is placed matters, therefore, repeated the same experiment by placing the phone on (a) glass plate, (b) metal plate (aluminum), (c) on the top of another smartphone, and (d) our custom wooden cantilever setup. Figure 4.21 shows the contour plots for each scenario. Evidently, glass causes the strongest side channel leak, and wood is minimum. Following experiments are hence performed on glass.



(a) (b) (c) (d)

Figure 4.21: Acoustic side channel leakage on: (a) glass, (b) metal, (c) on another phone, and (d) wood.

Results indicate that the SoV is well below the socially acceptable noise level. At a distance of 2 ft, SoV is less than 25 dB, comparable to a soft whisper as per human perception of loudness [110]. We further quantify this by comparing SoV against the ambient noises recorded in five common locations – departmental store, inside a moving car, coffee shop, class room, and computer laboratory. Table 4.2 shows that the ratio remains close to 2.

Table 4.2: Ratio of power of SoV signals to ambient noise at public places.

Location	Dep. Store	Car	Coffee Shop	Class	Lab
Power ratio	1.57	1.81	2.01	2.10	2.31

(2) Acoustic Leakage Cancellation:

Recall that the *Ripple* receiver records the sound and produces a synchronized phase-shifted signal to cancel the sound, and superimposes a jamming sequence to further camouflage the leakage. Figure 4.22 shows the impact of cancellation using a ratio of the power of the residual signal to the original signal, measured at different distances. Evidently, the cancellation is better with increasing distance. This is because the generated "anti-noise" approximates the first few strong harmonics of the sound. However, the SoV also contains some other low-energy components that fade with distance making the anti-noise signal more similar to the vibration's sounds. Hence the cancellation is better at a distance, until around 4ft, after which residual signal drops below the noise floor and our calculated power becomes constant. The original signal also decreases but is still above the noise floor past 4 ft, hence, the ratio increases.

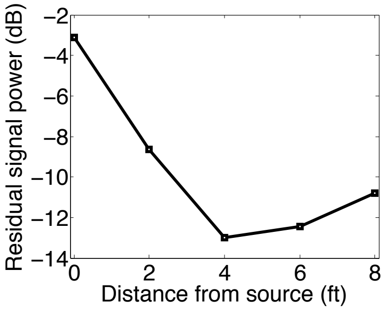


Figure 4.22: Ratio of residual to original signal power (in dB) at increasing distances from the source.

(3) Acoustic Jamming:

Ripple applies jamming to further camouflage any acoustic residue after the cancellation. To evaluate the lower bound of jamming efficiency, we make the experiment more favorable to the attacker. We transmit only two amplitude levels (binary data bits) at 10 bits per second. We place the phone on glass, the scenario that creates loudest sound. The eavesdropper microphone is placed as close as possible to the transmitter, without touching it. To quantify the efficacy of the jamming, we correlate the actual transmitted signal with the received jammed signal and plot the correlation coefficient in the Table 4.3. A high correlation coefficient indicates high probability of correctly decoding the message by the adversary, and the vice versa. The table shows the correlation values for various ratios of the jamming to signal power. Evident from the table, the correlation coefficient sharply decreases when *Ripple* increases the jamming power.

Table 4.3: The mean and standard deviation of the correlation coefficient for increasing jamming to signal power ratio.

Power ratio	0	0.4	0.8	1.2	1.6	2
Corr. mean	0.68	0.55	0.35	0.19	0.18	0.09
Corr. std. dev.	0.027	0.015	0.017	0.008	0.003	0.003

4.7 Points of Discussion

Needless to say, the system presented in this chapter is an early step – some aspects need deeper treatment, as discussed below.

Bounds and Optimality: We have not derived an upper bound on the capacity of vibratory communication, nor do we believe that our design decisions are optimal. We have taken an engineering approach and developed an end-to-end solution using techniques borrowed from RF/acoustic communication. Further work is needed to "tighten" the design towards optimality, including gains from coding and cancellation (on X, Y, Z dimensions).

Energy: Given that vibra-motors can be energy consuming, its important to characterize the energy versus throughput tradeoffs. For smartphone applications, vibrations are likely to be used occasionally for short exchanges, so perhaps energy is not a major hurdle. Nonetheless, when the phone battery is low, the ability to adapt can be a valuable feature.

Other Side-Channels: An attacker could exploit the visual channel with a high-speed camera [111] to decode the vibratory bits. Even physical eavesdropping may be a threat, where the attacker sneakily attaches an accelerometer to the surface on which the *Ripple* devices are located. A probable solution to such attacks can be "vibratory jamming". Essentially, the receiver's vibra-motor could generate a pseudo-random jamming vibration while receiving the data from the transmitter. Of course, the transmitter is unaware of this and performs normal transmission. The net vibration video-recorded by the attacker's camera is actually the sum of two vibrations, hiding the actual transmitted bits. However, since the receiver knows the pseudo-random jamming sequence it has deliberately injected, it can cancel it out. Of course, this pseudo-random vibration should have enough power to create desirable entropy at the transmitter, else the eavesdropper can focus only on the transmitter's vibration. We leave the viability of these attacks and mitigations to future work.

4.8 Related Work

Vibration Generation and Sensing: Applications in haptic HCI for assisted learning, touch-augmented environments, and haptic learning have used vibrations for communication to humans [112, 113, 114, 115, 116]. However, the push for high communication data rates between vibrators and accelerometers is relatively unexplored. Recently, personal/environment sensing on mobile devices has gained research attention. Applications like (sp)iPhone [117] and TapPrints [118] demonstrate the ability to infer keystrokes through background motion sensing. While many more efforts are around activity recognition from vibration signatures, this chapter aims to modulate vibration for communication.

Vibratory Communication: The systems [119] and [85] are probably closest to *Ripple*. They both encode vibrations through ON-OFF keying, with ON/OFF durations in the range of a second (i.e., around 1 bits/s). This is adequate for applications like secure pairing between two smartphones, or sending a tiny URL over tens of seconds. However, unlike *Ripple*, they do not focus on the wide range of PHY and cross-layer radio design issues and possible security leaks. Dhwani [1] is an elegant work on acoustic NFC and addresses conceptually similar problems, however, their acoustic platform are appreciably different from *Ripple*.

Technologies like Bump [120, 121, 122, 123, 124, 125, 126] use accelerometer/vibrator-motor response to facilitate secure pairing between devices. However, these techniques are primarily designed to exchange small signatures, as opposed to the arbitrary data transmission in *Ripple*. As indicated by researchers [119, 127], the lack of the dynamic secret message in Bump-like techniques makes them less secure in the wild. These modes also require Internet connectivity and trusted third-party servers to function, none of which is needed in *Ripple*.

4.9 Chapter Summary

This chapter is an attempt to explore a new modality of communication – vibration. Through multi-carrier modulation, orthogonal vibration division, and leakage cancellation, our system, *Ripple*, is able to achieve 200 bits/s alongside a strong level of security against side-channel attacks. While there is room for improvement, we believe the presented system could serve as a stepping stone for exciting vibration-based technologies and applications.

Chapter 5

Faster Communication through Vibrations

5.1 Overview

Motivation: Project *Ripple* [129], described in Chapter 4, is an attempt to enable communication through physical vibrations. The core idea is to harness the vibration motor (present in all smartphones and wearable devices) as a transmitter, and a motion sensor (like an accelerometer) as a receiver. When two smartphones come in physical contact to each other, the transmitter phone can vibrate to transfer bits of information. Transmission is even possible through other solid channels, such as between devices placed on a tabletop, or a finger ring communicating to a smartphone through bone conduction. While the exact application remains an open question (especially in the presence of NFC-like technologies), areas such as Internet of Things (IoT), intra-body networks, wearable security, and mobile payments are calling for new forms for short-range communication. Qualities of a vibratory radio, including zero RF radiation, contact-only authentication, mass-scale availability, and intuitive usability, may together fill an emerging business need. This project is motivated by this "bottom up" thinking and focuses on pushing forward the vibratory capabilities.

Prior Work: Of course, the fundamental idea of utilizing vibration as a communication modality dates back to acoustics – speakers modulate bits of information into air vibrations that are picked up by microphones. Air vibrations were later extended to water, enabling under water communication [130, 131, 132] and various applications, such as SONAR [133]. In recent years, vibration through solids has been of interest, motivated primarily by the need for proximal communication. Authors in [85, 119] used Morse-style communication at 5 bit/s to exchange security keys between two mobile phones in contact. *Ripple* [129], presented in Chapter 4, broke away from ON/OFF communication, and developed a viable

This chapter revises the publication "Ripple II: Faster Communication through Physical Vibration," in NSDI 2016 [128].

radio through techniques such as multi-carrier amplitude modulation, vibration braking, and simultaneous transmission over the three axes of the accelerometer. A self-sound cancellation technique also prevented acoustic eavesdroppers from decoding the sounds of vibration, offering improved security over RF-based approaches. As a first attempt to vibratory radio design, *Ripple* achieved data rates of ≈ 200 bits/s, but left various challenges and opportunities unaddressed. This chapter presents a subsequent work – *Ripple-II* – aimed at a far more mature radio stack and two example applications.

Technical Core: *Ripple-II*'s core redesign entails the following: (1) Replacing the accelerometer with the microphone as a receiver of vibrations. The key challenge pertains to separating vibrations from ambient sounds "picked up" by the microphone. While the availability of a second microphone offers the opportunity for sound cancellation, vibrations partly pollute the second microphone as well. Moreover, techniques such as active noise cancellation are inadequate since residual phase mismatches – often tolerable in human hearing applications [134] – seriously affect demodulation. We develop variants of adaptive filtering schemes, enhanced with an understanding of the interference conditions. (2) We also discover an opportunity that allows the vibra-motor to partially sense ambient sound interference, through a phenomenon called back-EMF in electronic circuits. The transmitter extrapolates from this partial information, using curve fitting techniques, and develops a proactive symbol retransmission scheme. The problem is new to the best of our knowledge – unlike existing wireless systems, here the transmitter is aware of the receiver's interference conditions and can adapt at the granularity of symbols. This opens both challenges and opportunities.

System and Applications: We engineer a completely functional prototype, which entails a full OFDM stack, coping with ADC saturation, synchronization, error coding, interleaving, etc. toward real applications, we develop a (clunky) wearable finger ring and demonstrate the viability of transmitting vibratory signals through finger bones. While signals attenuate through human tissues and muscles, effective bit rates of 7.41 Kbps is still possible, adequate for applications like two-factor authentication (i.e., when the user unlocks the phone, the vibratory password decoded by the phone serves as a second channel of authentication). We also explore a second application where devices are placed on tabletops, allowing for one-to-many multicast communication (e.g., a presenter sharing slides with all members in the meeting). Lastly, we include a video demo on our project website [135] – the demo shows the transmitter streaming music through OFDM packets over vibrations and the receiver's speaker playing it in real-time.

Platform and Evaluation: Our evaluation platform is composed of laptops, signal generators, vibra-motor chips, microphone chips, and home-grown circuits that interconnect them. In the basic scenario, the vibra-motor is attached to a short pencil to emulate a "stylus" like device, which then touches a microphone chip to transfer information. We generate various ambient sounds in the lab, including soft and loud music, people talking, machine hums, loud thuds and vibrations, and their combinations. Our PHY and MAC layer schemes are evaluated in these settings, against metrics such as SNR gain, bit error rate (BER), throughput, etc. At the application layer, we compute end-to-end data rate under modestly realistic settings, such as the human wearing the (vibra-motor embedded) ring and touching the microphone chip. We emulate wrist watches as well (2.23 Kbps), and perform an informal user study to understand if they feel the vibrations. We also explore achievable bit rates for tabletop communication, with devices placed at increasing distances on wooden surfaces.

Next Steps: There is much room for continued research and improvement. First, we have little understanding of PHY capacity and MAC layer optimality; intuitively, we believe that modeling the devices and the channel can yield reasonable performance bounds. Second, the sound cancellation techniques can perhaps benefit from deeper signal processing expertise – we have initiated collaboration toward this goal. Third, microphones and accelerometers may together present new opportunities that remain untapped in this chapter. Fourth, while the Chapter 4 discussed techniques to mitigate attacks on vibratory sounds, visual attacks still remain a threat – a high-speed camera, with line of sight to the device, may be able to decode vibrations. Finally, we need guidance on other possible use-cases and applications [136] of vibratory radios. Our ongoing work is focused on all these aspects.

In summary, the contributions of this chapter are:

- *An OFDM-based vibratory radio with microphones as the receiver.* The PHY layer uses variants of adaptive filtering to isolate vibrations from ambient sounds at the microphone; the MAC layer develops a transmitter side carrier sensing mechanism and uses it for proactive symbol retransmission.

- *A completely functional system borne out of significant engineering effort.* The effort includes hardware circuits on bread boards, to drivers for the vibra-motor, to bone conduction and real-time music streaming. Instantiation of the system in two applications: touch-based authentication and surface communication.

The overall architecture of *Ripple-II* is illustrated in Figure 5.1. The rest of the chapter expands on the main modules (shaded in gray) and briefly touches upon the techniques borrowed from the literature, and the engineering effort in building the prototype.

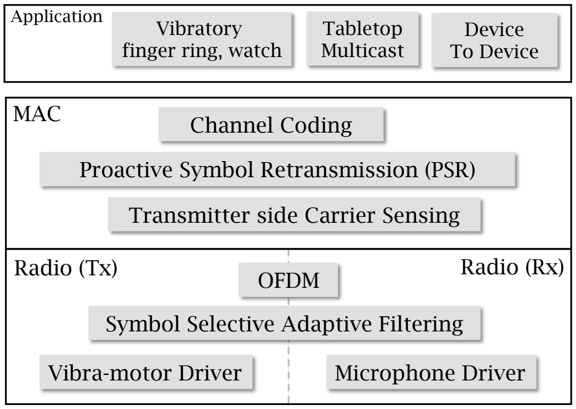

Application
Vibratory finger ring, watch
Tabletop Multicast
Device To Device
MAC
Channel Coding
Proactive Symbol Retransmission (PSR)
Transmitter side Carrier Sensing
Radio (Tx)
OFDM
Radio (Rx)
Symbol Selective Adaptive Filtering
Vibra-motor Driver
Microphone Driver

Figure 5.1: *Ripple-II*'s system architecture.

5.2 Development Platform

5.2.1 Vibratory Transmitter

As described in Chapter 4, an LRA vibration motor (or "vibra-motor") is an electromechanical device that moves a metallic mass rhythmically around a neutral position to generate vibrations. We control this vibra-motor through an Agilent 33500B waveform generator, which is indirectly controlled by MATLAB running on a laptop. The laptop generates the desired digital samples; the waveform generator converts the samples to an analog wave and transmits to the vibra-motor. The peak-to-peak output voltage is stabilized at 5V, the maximum supported by the vibra-motor chip. We generate OFDM symbols through MATLAB and drive the motor as desired.

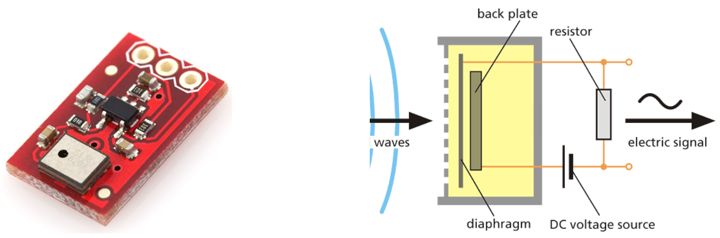
5.2.2 Microphone as a Receiver

Our prior work [129] used a vibra-motor as the transmitter and an accelerometer as the receiver.[1] The accelerometer demodulated vibratory QPSK symbols and corrected for errors using simple gray coding techniques. The low bandwidth of accelerometer chips (800 Hz) proved to be the main bottleneck to link capacity, resulting in $\approx$ 200 bits/s. This chapter breaks away from accelerometers and identifies the possibility of using microphones as a vibration receiver.

Like accelerometers, microphones also transduce physical motion to electrical signals using a diaphragm that responds to changes in (acoustic) air pressure. Figure 5.2 shows a micro-

[1] Accelerometers are MEMS devices that transduce physical motion into electrical signals by measuring the extent to which a tiny seismic mass moves inside fixed electrodes (see [93] for details).

phone chip and the basic internal architecture – as the diaphragm vibrates inside a magnetic field, the produced electrical signals are amplified and sampled by an ADC. *Unsurprisingly, the diaphragm can also be made to vibrate by physically touching a vibra-motor to the microphone chip.* Since microphones are designed for greater sensitivity and operate over a wider frequency range, they can serve as a better receiver (an alternative to accelerometers). The tradeoff, however, is that the vibration measured at the ADC is actually an aggregate of the physical vibration and the air vibration from ambient sounds (e.g., people talking). *Ripple-II* needs to isolate physical from acoustic vibrations to accomplish high-bandwidth vibratory communication.



(a) (b)

Figure 5.2: (a) MEMS microphone chip, the diaphragm hole near bottom left. (b) Microphone circuit sketch.

Figure 5.3 shows our overall hardware setup. The vibra-motor is taped to the back of a short pencil and the tip of the pencil now acts like a stylus, touching the microphone chip. Transmission bits produced by the laptop are converted to a signal waveform by the signal generator, which then drives the transmitter; the microphone decodes these bits through real-time processing on a laptop. The following subsections detail the technical modules in the PHY, MAC, and application layers.

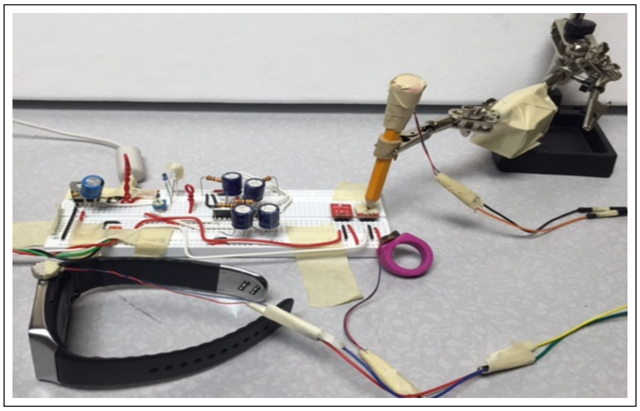


Figure 5.3: *Ripple-II*'s experimentation setup (three vibra-motors attached to a pencil, ring, and watch). The stylus touching a microphone, the second microphone nearby.

5.3 PHY: Vibratory Radio

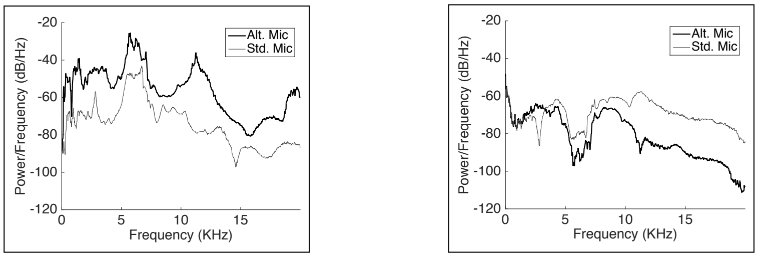
We begin with the design of the microphone-receiver, followed by our implementation of OFDM.

5.3.1 Separating Vibration from Sound

While the microphone offers a larger bandwidth compared to the accelerometer, its sensitivity to ambient sound is a disadvantage. Unless filtered out, the vibration SINR will be low, especially in loud environments. We attempted various techniques (algorithms and hacks); we detail the ones that worked and touch upon the failures.

(1) Covering the Sound Hole:

The microphone chip has a circular opening (like a small hole) that exposes the diaphragm to air pressure. To prevent ambient sounds from polluting *Ripple-II*'s vibratory signals, we covered the hole with a stiff synthetic rubber sheet (somewhat like a stethoscope). However, when a vibrating object comes in contact with this rubber sheet, the air trapped inside the hole still oscillates, causing the diaphragm to produce the desired signals. Figure 5.4 compares the frequency responses of the altered and the standard microphones for vibration and sound, respectively. Figure 5.4(a) shows an average 18.2 dB gain for vibration signals over the standard microphone; at some frequencies the difference is 43.8 dB. On the other hand, Figure 5.4(b) shows that the average sound attenuation at the altered microphone is around 12.3 dB. For both the signal (i.e., vibration) and the noise (i.e., ambient sounds), the higher frequency proves better (useful later in Section 5.5).



(a) (b)

Figure 5.4: (a) Covering the sound hole offers improved vibration signal. (b) Covered sound hole attenuates sound signals in comparison to the standard microphone.

(2) Canceling Ambient Sound:

Let us denote the vibration signal from the stylus as $V(t)$ and the ambient sound signal as $S(t)$. *Ripple-II* aims to subtract $S(t)$ from the aggregate signal ($A(t) = V(t)+S(t)$) received through the microphone. A second microphone present in many devices today is a natural opportunity. In an ideal case, the second microphone should only receive the ambient sound $S(t)$ and none of the physical vibration $V(t)$ since the stylus is not in direct contact with it. In reality, however, physical vibrations also leak into the second microphone. Also, both microphones are affected by a high-intensity electrical noise, $E(t)$, caused by their common supply voltage. Frequencies of this noise range from 300 Hz to 2500 Hz and its amplitude is comparable to $V(t)$. Finally, the microphone output also includes a native hardware noise, typically assumed to be uncorrelated additive Gaussian, denoted N_1 and N_2 for the respective microphones.

Based on the above factors, the overall system can be modeled as shown in Figure 5.5. The signal output from the i^{th} microphone, Y_i can thus be expressed as:

$$Y_i = H_{v_i}V + H_{s_i}S + H_eE + N_i$$

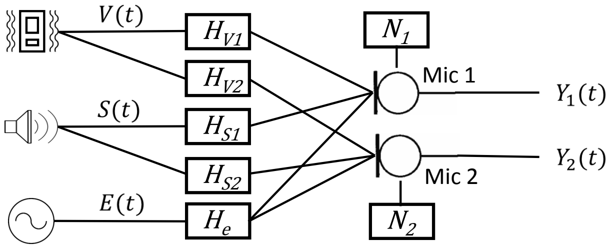


Figure 5.5: Modeling the signals and interferences at each of the microphones; H denotes the channel matrix and V, S, E, denote vibration, sound, and electrical noise, respectively.

We note that extraneous physical vibrations may occur when *Ripple-II* is transmitting information (for example, in a moving vehicle). Such vibrations are included in S since it is likely to affect both the microphones similarly. We also note that the electrical noise E is highly correlated and synchronized at both microphones, since they share a common power source. Under this model, our goal is to extract V from Y_1 and decode the content.

(3) Failed Attempts (MIMO, NC, rPCA):

We discovered early that electrical noise E can be removed effectively by low-pass filtering Y_2 and subtracting from Y_1. Since E dominates and is phase matched across both microphones, the residue after subtraction minimally impacts V. Thus, we can rewrite $Y_i = H_{v_i}V + H_{s_i}S + N_i$. This appears to be in the form of MIMO and hence solvable without difficulty. Unfortunately, the channel matrix for ambient sound, H_{s_i}, cannot be easily measured since *Ripple-II* has no control over the sound sources. Also, due to the time-varying nature, statistical estimates are difficult.

Classical noise cancellation seems applicable [137], however, the statistical nature of this algorithm does not mitigate phase mismatches. The result after subtraction does preserve the amplitude of the desired signal, which is often adequate for human perception [134]. In *Ripple-II*, however, we need phase alignment too, or else, QAM-based demodulation falters. Put differently, requirements to improve human hearing experience is less stringent than the requirements for data communication.

Robust PCA is an algorithm from 2009 used for background separation [138]. The technique builds on the result that, under certain conditions, a given matrix can be factorized into a sparse and low-rank matrix. For instance, in a talk show video, static background walls could serve as the low-rank matrix (due to high similarity across video frames) and the talking people could make up the sparse matrix. In our case, we envisioned the ambient sound to be sparse and the vibration to be low-rank (since the cyclic prefix of OFDM symbols can be organized to look identical across time).[2] Unfortunately, we could not design the matrices to attain adequate amount of both sparsity and low-rank-ness. During the short time shifts for which the OFDM vibration symbols were identical, the sound signal changed enough that they were not sparse. When sound proved to be sparse over longer time frames, the low-rank-ness disappeared. The outcomes of factorization yielded marginal gain.

(4) Symbol Selective Adaptive Noise Filtering:
Adaptive filtering (AF) is an established technique that can accept the two microphones' signals as inputs, say ($Y_1 = V_1 + S_1$) and ($Y_2 = V_2 + S_2$), and can attempt to adapt the filter coefficients for Y_2 such that the $Y_1 - Y_2$ is V_1. Conceptually, AF bolsters Y_2 in the regions where it correlates well with S_1, and then subtracts from Y_1. This works best when S_1 and Y_2 are somewhat correlated to each other, but neither is correlated to V_1. However, in our system, when ambient interference is low (i.e., V dominates S), then Y_2 correlates well with V_1 – this is why AF subtracts away the vibratory signals from Y_1, defeating the purpose. However, we observe that if we could identify OFDM symbols that are in error (i.e., S dominates V), then perhaps *only the erroneous symbols* could be subjected to AF. Since S_1 and Y_2 would correlate well in such cases, the result of $Y_1 - Y_2$ could converge to V_1. Using this intuition, we design *Symbol Selective Adaptive Noise Filtering* (SANF), sketched in Figure 5.6.

Erroneous Symbol Detection: The main opportunity emerges from measurements that revealed that the vibratory channel responses at the primary and secondary microphones –

[2] Without too many details, note that time domain signals can be shifted by several samples and yet, by OFDM design, they will map to the same frequency domain symbol – this is why we could generate low-rank-ness

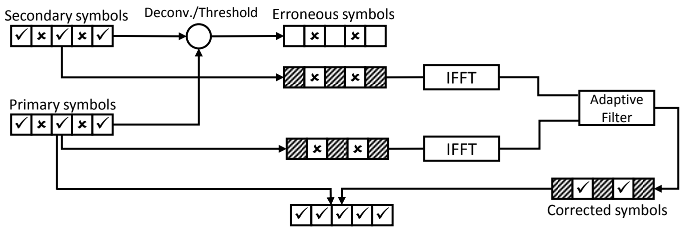


Figure 5.6: SANF infers erroneous symbols and only feeds these to the AF module.

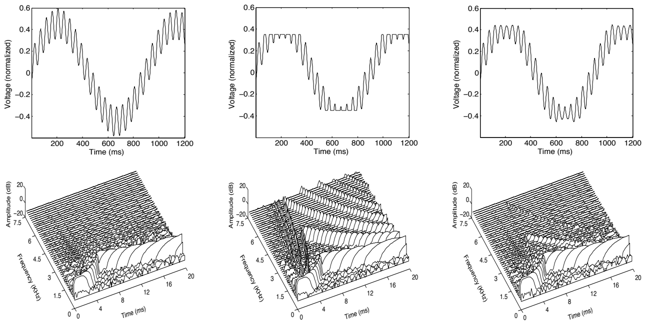
H_{v_1} and H_{v_2} – maintain a constant ratio under light or no interference. This is likely due to the same solid channel between the two microphones. In the presence of sound, however, the same ratio gets polluted and thereby loses the constancy property (since sound varies over time). Thus, we first perform channel estimation for pilot subcarriers scattered across the OFDM symbol. We synchronize the secondary microphone and estimate the channel for that same pilot (the slight time offset does not affect due to the protection from the cyclic prefix). Now, deconvolution of the primary and secondary signal in the frequency domain yields the complex gain, α_p for each pilot p.

Recall, the goal is to estimate the pristine ratio of H_{v_1} and H_{v_2} in the presence of sound interference; the α_p we have is still polluted by sound interferences. Thus, we perform a least square estimation of the ratio and compute α^* for each subcarrier. Now, for any non-pilot symbol to be erroneous, the computed complex gain between the primary and secondary must be far from α^* for that sub-carrier. Once the erroneous symbols are identified, we convert only those to the time domain, leaving the error-free subcarriers untouched. We obtain the time domain signals from both of the primary and secondary microphone and feed them to an adaptive filter for noise cancellation. The output of the adaptive filter is then demodulated to recover the vibratory symbols.

(5) Amplifier Gain and Clipping:

To maximize the power of the vibratory signal, we operate the receiver signal amplifier at near-maximum gain and leave just enough headroom for typical ambient sound (measured empirically). Of course, sometimes the ambient sound exceeds the headroom and drives the amplifier to *saturation* [139]. Figure 5.7(a) shows the output of the unsaturated amplifier; Figure 5.7(b) shows the saturated case – a truncated waveform. Unsurprisingly, this "clipping" effect spills energy into other frequencies, causing interference in an OFDM system. We alleviate such frequency distortion effects by replacing the flat saturation region with a cubic spline interpolation of the signal – Figure 5.7(c).

Our measurements also recorded consistent interference at lower frequencies (< 500 Hz), caused by a combination of winds from air vents, thermal noise from electrical equipment,



(a) (b) (c)

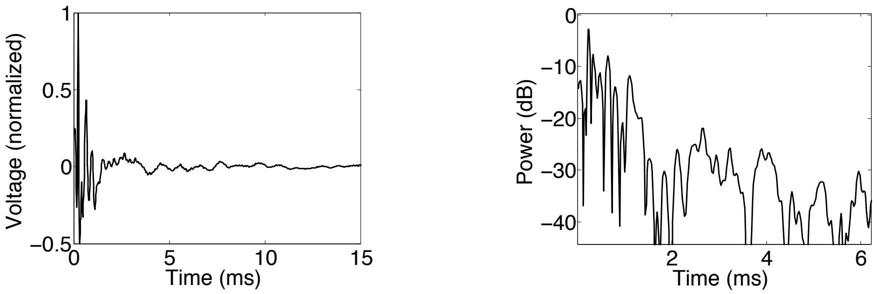
Figure 5.7: The waveform (first row) and spectrogram (second row) of the signal at various stages: (a) actual signal, (b) distorted signal after clipping, (c) corrected signal after spline interpolation.

as well as vibrations of the human hand while holding the transmitter. The vibra-motor also exhibits resonance frequency at around 232 Hz, causing the system to destabilize due to the high power gain. We deemed it suitable to sidestep these problems and moved the transmission band to begin from 500 Hz.

5.3.2 OFDM over Vibration

We implement OFDM [140] over the vibra-motor and microphone link. Although an engineering effort, we briefly summarize the parameter selection process, particularly those influenced by the vibratory channel.

(1) Channel Impulse Response:
Although the vibratory channel is dominantly time-invariant and frequency selective, human factors such as hand movements and varying angle of contact inject variability. Measurements suggest similarity to a Rician fading model [141], with a strong line of sight path. The weaker multipath components are caused by the inertial movement of the motor mass – reverberation of the medium distorts the signal and multiple reflected/delayed replicas combine to create an elongated decaying response at the output. We measure the impulse response of our system using the *exponential sine-sweep* method [142] during which sinusoids of exponentially increasing frequency drives the motor. The output from the microphone is de-convolved with the weighted reverse sine-sweep to obtain the impulse response (the technique offers robustness against noise and nonlinear distortions). Figure 5.8(a) and (b) show the measured impulse response and the corresponding *power delay profile* (PDP).



(a) (b)

Figure 5.8: (a) Channel impulse response of the vibratory channel. (b) Power delay profile of the vibratory channel.

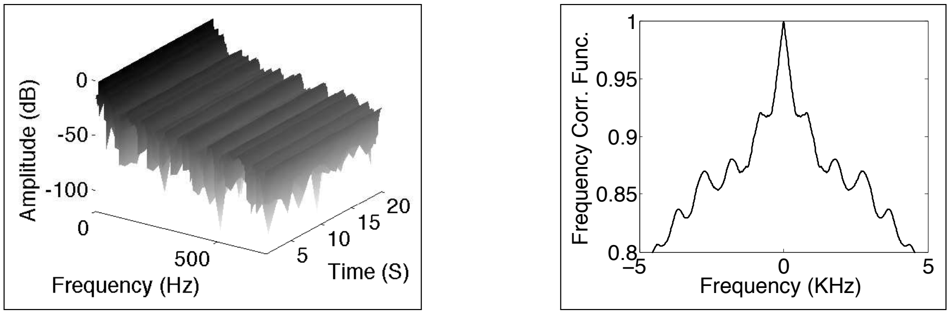
(2) Parameter Selection:

Cyclic Prefix: The PDP shows 0.4 ms before the multipath energy falls below 10 dB of the highest peak, called "10 dB maximum excess delay". This should be the separation between symbols to avoid inter symbol interference (ISI). We set the guard interval conservatively to 1ms, however, instead of leaving the channel idle during this interval, we insert 1 ms of the last part of the symbol. This is called the cyclic prefix (CP) which helps cope with time synchronization errors without affecting the orthogonality of sub-carriers.

Subcarrier Bandwidth: The vibratory channel, as mentioned earlier, offers long channel coherence time, allowing for small subcarrier spacing. In practice, however, due to unpredictable phase noise, the inter carrier interference (ICI) becomes severe with small subcarrier spacing. On the other hand, the subcarriers become frequency selective for bandwidths larger than the coherence bandwidth of the channel. In such cases, the channel is no longer flat and hence equalization techniques falter [143, 144]. We measure the coherence bandwidth to be 480 Hz (see Figure 5.9) – this is the width of the frequency-correlation function using a threshold of 0.95. We then choose the subcarrier bandwidth conservatively to 40 Hz, less than the $\frac{1}{10}^{th}$ of the coherence bandwidth.

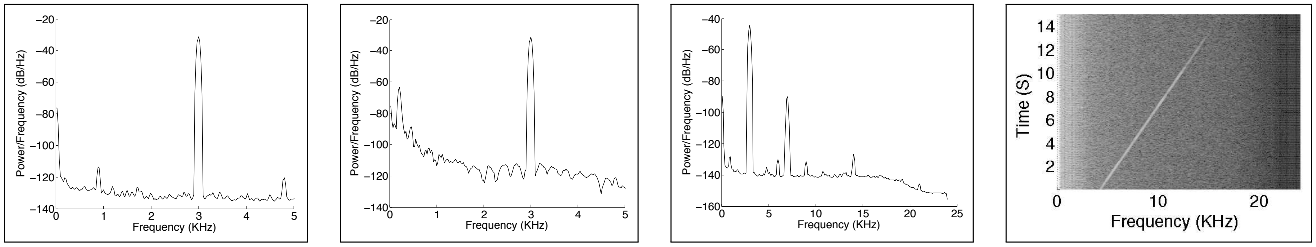
Total Bandwidth: We choose the total bandwidth to be 12 kHz, equal to the coherence bandwidth at correlation threshold of 0.7.

With this PHY layer in place, we now focus on a vibratory MAC layer, with the goal of reliably delivering packets to the receiver even under interference.



(a) (b)

Figure 5.9: (a) Temporal stability of the channel. (b) The frequency-correlation function indicates the coherence bandwidth of 480 Hz for the width threshold of 0.95.



(a) (b) (c) (d)

Figure 5.10: (a) Vibra-motor driven by a 3 kHz voltage and no interference in the environment. (b) Interference introduced in the environment raises the noise floor, especially at lower-frequency bands. (c) Clear detection of 7 kHz interference caused by a nearby vibrator. (d) Spectrogram of acoustic chirp detected through back-EMF – the chirp was played through a speaker placed 4 ft away.

5.4 MAC Layer Design

Reliable packet delivery entails retransmitting a packet when it is received in error. In wireless systems, since the transmitter is unaware of the receiver's channel conditions, the error detection happens reactively, through an ACK from the receiver. Vibra-motors offer a new opportunity – we find that the receiver's interference conditions can be sensed at the transmitter through what is known as *back-EMF*. Thus, the transmitter could potentially transmit and listen at the same time, infer symbol collisions, and retransmit symbols proactively. Efficiency can improve but some issues need mitigation.

5.4.1 Sensing Interference from Back-EMF

Back-EMF is an electro-magnetic effect observed in magnet-based motors where relative motion occurs between the current carrying armature/coil and the magnetic field. In our vibra-motor, when the permanent magnet oscillates near the coil, the flux linkage with the coil changes due to the driving voltage and/or vibration noise. According to the Faraday's law of electromagnetic induction [145], this changing flux induces an electromotive force in the coil. Lenz's law [146] says this electromotive force acts in the reverse direction of the driving voltage, called *back-EMF of the motor*. As the rate of change of the magnetic flux is proportional to the speed of the magnetic mass, the back-EMF serves as an indicator of the extraneous vibration experienced by the mass.

Unsurprisingly, the interfering vibrations generate subtle movements of the vibra-motor mass, causing the voltage changes around a small resistor to be in milli-volts (below the ADC noise floor).[3] We design a low noise amplifier, limiting the parasitic inductance/capacitance, to amplify this voltage $100x$ before feeding it to the ADC sampling circuit. Figures 5.10(a,b) show the difference between interference-free and interfered transmissions, as sensed through back-EMF. The noise floor increases, especially at lower frequencies where the interference is dominant. Figure 5.10(c) shows another case where a 7 kHz interferer – a second interfering vibra-motor – is placed on the same table as our experiment; the transmitting vibra-motor detects the corresponding spike at 7 kHz. We also played an acoustic chirp on a speaker 4 feet away from our devices – Figure 5.10(d) shows the chirp spectrogram, a reasonable reproduction of the actual. The findings extend hope that back-EMF can be useful to designing transmitter-side collision inference protocols.

5.4.2 Vibratory Interference

Before moving into protocol design, we characterize the nature of vibratory interference experienced by the microphone. Interferences are broadly of two kinds. (1) Ambient acoustic sounds, such as people talking, background music, machine hums, etc. and (2) physical vibrations caused by objects such as running table fans, taps and thuds on tabletops, and even natural vibration of human hands when they are holding the devices. Figure 5.11 shows the spectral graph of several example interferences, measured in isolation. The key observation is that interferences are heavily biased to the lower-frequency bands; frequencies higher than 6 kHz are rarely impacted.

Figure 5.12 shows the 3D contour of acoustic interference across frequency and time – the interference stems from loud human voices. The key observation is that for any given frequency, the signal amplitude of the interference rises with time, reaches a peak, and decays again. This characteristic is highly common in a wide range of interferences, primarily

[3] The measuring circuit samples the induced current as a voltage drop across a series resistor. We keep this resistor value below 0.02% of the motor's coil resistance so that the electrical property of the system remains unaffected.

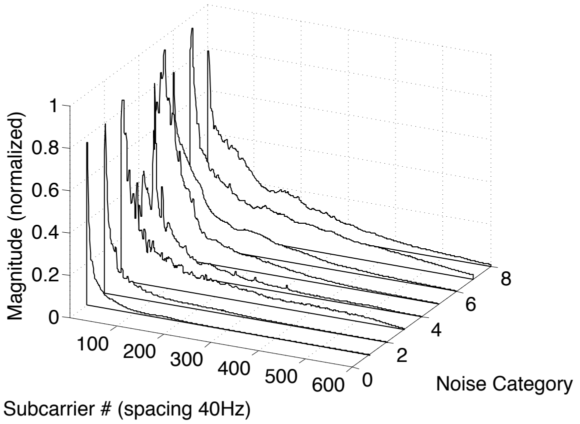


Figure 5.11: Spectral properties of various interferences occurring in the natural environment.

because instantaneously starting or stopping strong signals is difficult. Occasionally, we find certain machines capable of producing a sudden spike, however, their decay is still slow. We leverage back-EMF along with these properties of the interference to design a MAC protocol, called Proactive Symbol Recovery (PSR).

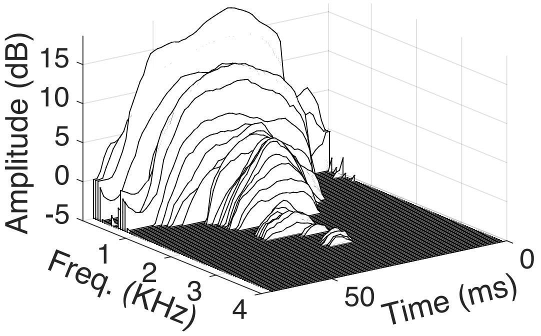


Figure 5.12: 3D contour of acoustic interference across frequency and time.

5.4.3 PSR Protocol: The Problem Definition

The protocol problem can be abstracted as follows. Consider a packet P composed of many OFDM symbols, $[S_1, S_2, S_3, ...]$, each symbol composed of n subcarriers $[f_1, f_2, f_3, ...f_n]$. Figure 5.13 shows the pictorial representation of such a packet, in the form of a time-frequency grid. Assume that the gray region denotes the incidence of interference, essentially the top view of Figure 5.12. Now, with back-EMF, the transmitter is able to sense receiver-side interference, however, the sensing is not accurate. To be able to reliably detect interference (i.e., reduce false positives), the transmitter can increase the sensing threshold – interference detected above this threshold is strongly indicative of actual interference. Assume that the interference above a given threshold is the black region in Figure 5.13.

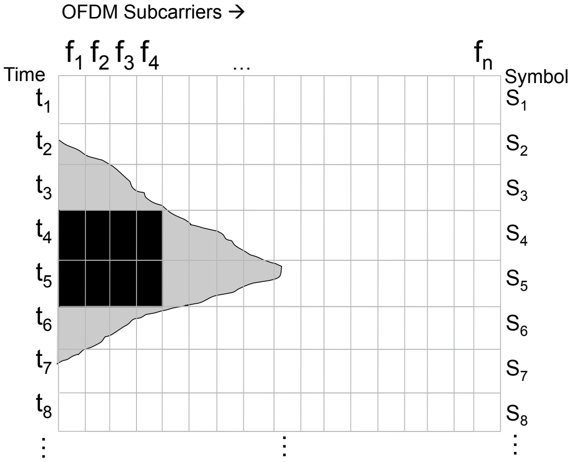

OFDM Subcarriers →
f_1 f_2 f_3 f_4 ... f_n
Time
Symbol
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8
S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8

Figure 5.13: A packet represented in terms of OFDM symbol, each symbol to be transmitted over time.

The protocol question is: *Which symbols should the transmitter retransmit, and when?* Transmitting only the symbols that are affected by the black color may still leave too many erroneous symbols – the coding scheme at the receiver may not be able to recover the packet. The transmitter essentially needs to estimate the symbols affected by the gray region too, and retransmit a subset of those symbols [147, 148]. Clearly, not all the gray-color affected symbols need to be transmitted since the coding scheme can indeed correct for some errors.

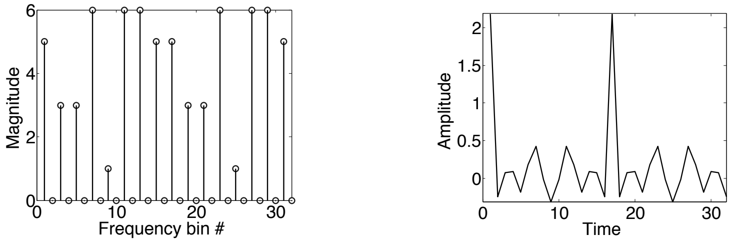
A second question pertains to interference adaptation. Once interference is detected at time t_4, the transmitter must adjust the subsequent transmissions to cope with the interference. Any adjustments – such as rate control – would need to be communicated to the receiver through some control information. However, unlike packets, symbols are not prefaced with headers; dedicating some subcarriers to a control channel will be wasteful in general. Under this constraint, the protocol needs to adapt to interference and concisely convey its adaptations to the receiver. The basic problem is new to the best of our knowledge, since existing protocols assume that the receiver has better estimates of error than the transmitter [147, 148]. In our case, the transmitter is better aware of the interference but has no control bits to convey its adaptations.

5.4.4 Proactive Symbol Recovery Protocol

The PSR protocol develops two heuristics – interference extrapolation and implicit control signaling – described next.

(1) Interference Extrapolation: Only the contour of the interference within the black region (in Figure 5.13) is visible to the transmitter – one could metaphorically envision it as the "part of the iceberg above water". Based on the visible shape, the transmitter may be able to extrapolate the "submerged" shape, generating an estimate of the gray region. Our measurements have consistently indicated that the interference decay is well-behaved, of course with some jitter. Hence, we model this as a curve fitting problem, and use a third-order cubic spline (the high-frequency jitters are not captured). Given multiple silhouettes, one per-subcarrier, we pick the silhouette whose peak is at 80^{th} percentile among all peaks. Using this we develop an estimate of the gray region.

(2) Implicit Control Signaling: As mentioned earlier, the transmitter needs some control bits for signaling its actions to the receiver. To this end, we use a simple interleaving idea from the basics of signal processing. Specifically, when alternate subcarriers are loaded with data (and the ones in-between left empty), the time domain representation of the OFDM signal exhibits two identical copies (Figure 5.14). We call this the $2x$ interleaving mode. When every fourth subcarrier is loaded, the time domain signal shows four identical copies of the same signal. The receiver recognizes these identical copies in time domain and decodes the control information. In frequency domain, it extracts the data from every second (or fourth) subcarrier and ignores the others. Of course, we are aware that the control bits are not free – the $2x$ and $4x$ interleaving modes reduce the bandwidth. However, we also note that energy on the loaded subcarriers increases – a $2x$ mode exhibits a 3 dB gain (nearly double), lowering chances of demodulation error.



(a)

(b)

Figure 5.14: (a) The 2x interleaving in frequency. (b) Identical signal parts in time domain.

(3) Protocol Design: We now describe the basic operation of the PSR protocol (we continue to refer to the toy example in Figure 5.13). When no interference is detected by the transmitter's back-EMF sensor (i.e., until time t_4), symbols are sent as usual. Upon detecting interference at t_4, the transmitter records the symbol that was affected (namely, S_4), and performs the subsequent symbol transmissions (S_5) at $2x$ interleaving mode. This continues until the interference has subsided below the transmitter's threshold. At this point, the transmitter performs the extrapolation using the interference decay data, starting from the last-observed interference peak. The interpolation suggests that the receiver may continue to experience interference until some time in the future, say until t_7. Therefore, the transmitter continues symbol transmissions in $2x$ mode, after which it falls back to no-interleaving. Observe that this interleaving mechanism is akin to halving the rate, except that it helps inform the receiver about the rate reduction.

Ideally, the interference extrapolation may help recover the symbols S_6 and S_7, however, symbols S_2 and S_3 could also be heavily interfered. To this end, the transmitter also extrapolates the front portion of the interference, and remembers the symbols that need retransmission. Once all the symbols have been transmitted, it now retransmits these symbols (S_2, S_3, and S_4 in this toy case), at the appropriate interleaving mode permissible by the then channel conditions. Importantly, the receiver must identify that these symbols are actually duplicates of prior symbols. Hence, the transmitter marks the start of these retransmissions with a $4x$ interleaved packet – the packet includes indices of all symbols that are being retransmitted. The encoding of indices is efficiently done to utilize the fewest bits possible, telling the receiver how many retransmissions to expect and which prior symbols to replace. The receiver demodulates all the symbols, performs the appropriate replacements, and feeds them through the decoder.

(4) Coding for Error Correction: Needless to say, extrapolation will incur errors, and back-EMF sensing will experience false negatives. This will leave erroneous symbols at the receiver even after retransmissions. In fact, it would be inefficient for the transmitter to recover all symbols since the decoder at the receiver would be able to correct for some of them anyway. We implement a standard 2/3 convolutional code, with constrain length 7, to cope with inherent symbol errors in the transmission. We implement a hard decision Viterbi decoder with trace back depth of 30 to recover the bits. To cope with heavy bursts in error, we use an interleaver to spread out the bursts.

5.5 Evaluation

5.5.1 Complete Hardware Prototype

Figure 5.15 shows the complete interconnection of the hardware elements in *Ripple-II*. Very briefly, the receiver (on the left side) draws power from the USB port of a Dell laptop (or any mobile device or raspberry-pi/arduino) serving as the controller. Instead of using a separate ADC, we abuse the *line-in audio input port* of the laptop, which comes equipped with a high-speed ADC and a driver to push samples to user space. We connect signals from each microphone to one of the channels in the line-in port with the help of a three-conductor (TRS) audio jack. We run the appropriate driver to sample the signal at 48 kHz, 16-bit stereo mode.

The transmitter (shown on the right side) also uses a similar approach. The software controller generates digital samples that are converted to analog via the DAC of the audio port. This output signal (with appropriate amplification and shaping) feeds into the vibra-motor, which is in turn attached to the stylus or ring. We sample this line-in port at 48 kHz to collect the back-EMF signal along with the reference voltage. Offline processing is performed in MATLAB; real-time music streaming is performed on GNURadio.

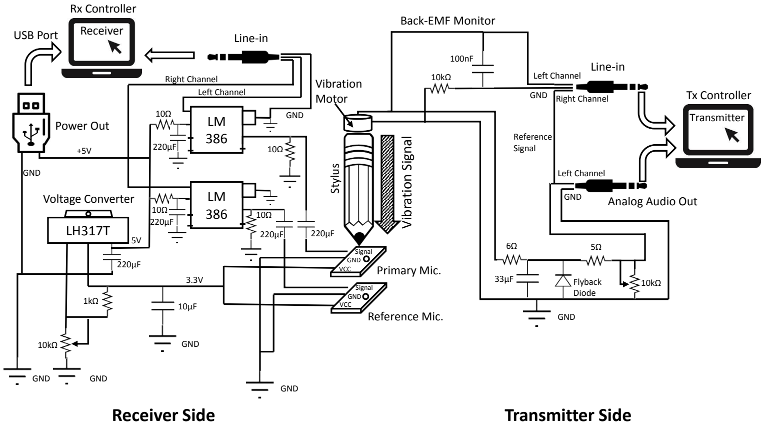


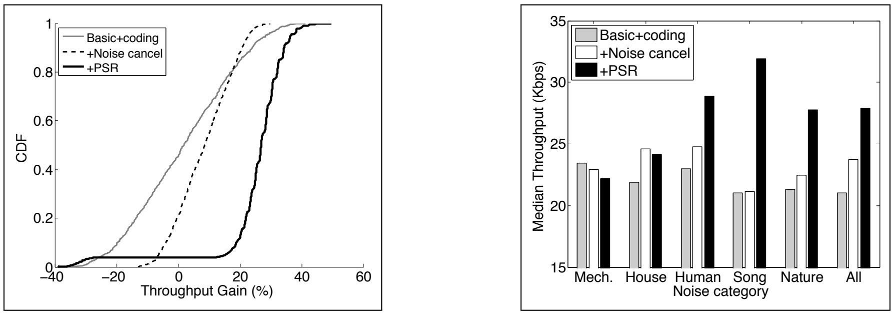
Figure 5.15: The complete hardware internals of *Ripple-II*.

5.5.2 Performance Results

We present end-to-end results first, followed by zoomed-in results from acoustic noise cancellation (SANF) and proactive symbol retransmission (PSR). Our final results are drawn from 100+ sessions of experiments, each session either $1-3$ min. long, and entails vibratory transmission against diverse ambient sounds, ambient vibrations, modulations, etc. We collected 800 samples of ambient sound (e.g., supermarket ambience, in classroom noise, music nearby, etc.) and 15 ambient vibrations (e.g., walking, moving in a car, tapping on table). Half of sessions were against the natural lab sound conditions; for the other half, we played external ambience sounds through a speaker and generated vibrational noise through an external vibra-motor placed on the table. As a baseline we use the basic OFDM microphone receiver running on our hardware platform (including the covered sound hole). We compare this baseline against (1) baseline + coding, (2) baseline + coding + SANF, and (3) baseline + coding + SANF + PSR.

(1) *Ripple-II* Results:

Figure 5.16(a) shows the CDF of throughput gain computed from all the experimentation data, across all possible noise environments. The communication link operates in a high bit error rate (BER) regime and coding schemes perform worse than expected. The median gain with SANF is around 10%, with a small fraction of cases leading to negative gain. However, PSR brings appreciable benefits, mainly from retransmitting erroneous symbols and bringing the errors below the tolerable threshold. Median throughput gain with PSR is 26.6%. Figure 5.16(b) reports the breakup of raw throughput under various ambient sound categories. Under mechanical sound spikes alone, the performances of SANF and PSR are weak – the interpolation in PSR falters, while SANF's symbol error detection scheme is not sensitive enough. However, in other categories of noises, throughput improves – the median throughput in the "All" noise category is ≈ 27 Kbps.



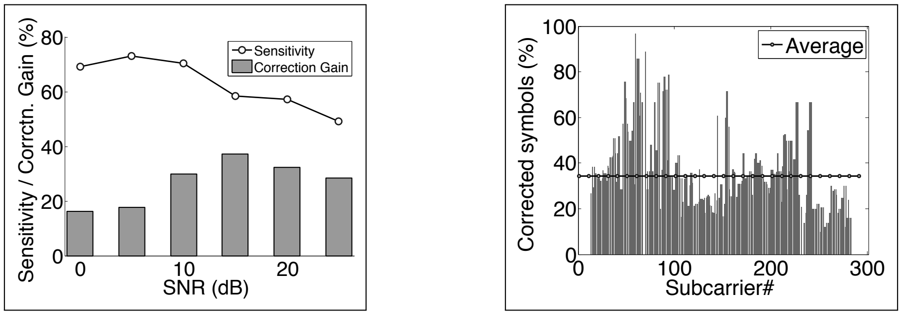
(a) (b)

Figure 5.16: (a) Throughput gain across all experiments. (b) Median throughput across different ambient sound categories.

(2) SANF Results:

Figure 5.17(a) zooms into symbol selective aspect of SANF, and shows the fraction of symbols corrected over normal adaptive noise filtering. The correction gain improves with higher

SNR, but falls beyond 15dB. This is because at > 15dB SNR, SANF is unable to detect the symbol errors correctly since the interference is less pronounced – the inability to identify the erroneous symbols derails adaptive noise filtering. The sensitivity curve captures this behavior, suggesting that the symbol correction efficacy is both a function of SNR and sensitivity. Figure 5.17(b) shows the gain across each subcarrier – the graph is for the best SNR, 15 dB.



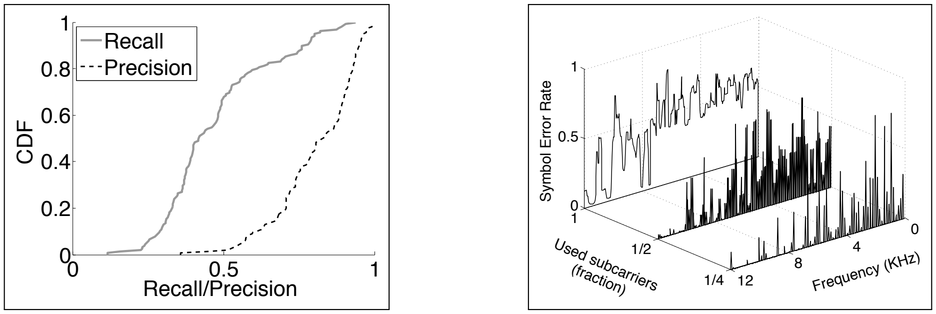
(a) (b)

Figure 5.17: (a) Variation of SANF's cancellation gain and sensitivity against increasing SNR; sensitivity is the fraction of erroneous symbols detected by SANF. (b) The noise cancellation gain as the percentage of erroneous symbol per subcarrier.

(3) PSR Results:

The core design elements in PSR pertains to (1) back-EMF-based sensing and extrapolation of the interference, and (2) reducing symbol errors via $2x/4x$ interleaving (expected to increase energy). To evaluate extrapolation, we first identify the set of truly erroneous symbols that should have been retransmitted by the transmitter. We know the set of symbols that PSR actually retransmitted. From these two sets, we compute the precision and recall of PSR, reflecting the combined efficacy of back-EMF sensing and interpolation. Figure 5.18(a) shows the results – the precision is strong but the recall is weak, indicating that PSR is conservative. This is expected/desirable since we intend to not retransmit excessively, which reduces inflation of the packet and also allows the decoder to correct for the residual errors. Of course, there is room to tune the interpolation scheme and the back-EMF sensitivity – we leave this to future work.

Figure 5.18(b) shows the reduction in symbol error rate when half and one-forth subcarriers are loaded with data (recall we denoted this as $2x$ and $4x$ modes of transmission). Under heavy channel interference, $2x$ mode substantially reduces symbol errors, offering effects similar to rate control. However, the $2x$ mode also implicitly includes a control bit that the receiver can recognize. Measurements show that the control signaling was near perfect, meaning the receiver almost always extracted the correct data from $2x$ and $4x$ transmissions.



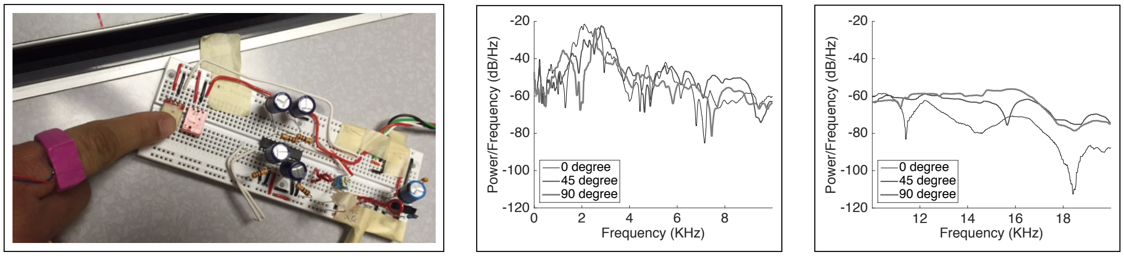
(a) (b)

Figure 5.18: (a) Precision and recall to evaluate the back-EMF sensing and interference extrapolation scheme. (b) The per subcarrier symbol error rates using all, 1/2, and 1/4 of the subcarriers, while the noise power is constant.

5.5.3 Applications and Capabilities

We explore potential applications of *Ripple-II*, namely a vibratory ring and watch; tabletop communication; and device-to-device transfers.

(1) Finger Ring for Authentication:
We envision touch-based two-factor authentication – a user wearing a *Ripple-II* ring or watch could touch the smartphone screen and the vibratory password can be conducted through the bones. The core notion generalizes to other scenarios, including unlocking car doors, door knobs, etc. While a usable system would need maturity in interfaces, energy, etc., this section only discusses the communication aspects of through-bone transmission. Figure 5.19(a) shows the crude finger ring prototype, placed on the index finger of the user. For our prototype, the ring is powered by a battery located outside the ring and connected via long wires. The cylindrical vibra-motor is placed horizontally on the finger to maximize area of contact, however, placement influences communication.



(a) (b) (c)

Figure 5.19: (a) Finger ring operated at 8 kHz. (b) Incidence angles affect lower frequencies less. (c) Higher frequencies in a piston oscillator become directional and hence delivers less energy in unaligned directions.

Figure 5.19(b,c) shows the variation of signal power for three different incidence angles between the vibra-motor and the finger – incidence angle defined as the angle between the finger bone and the direction along which the vibrator mass oscillates (which is perpendicular to the base of the cylinder). Evidently, at lower frequencies, the incidence angle does not impact the signal, however, at higher frequencies the higher incidence angles reduce SNR. Moreover, higher frequencies are also less effective for signal propagation through the human body. Thus, we decide to operate the ring at 90° incidence but focus the power budget to within 8 kHz.

We also performed similar experiments with a watch – pasting the vibra-motor on the wrist bone below the watch. Performance degrades as expected, due to a longer conduction path from the wrist to the microphone. Table 5.1 summarizes results. Five student volunteers experimented with our prototype and none of them were able to feel or hear the vibrations at all.

Table 5.1: Performance of the vibratory communication prototype with wearable devices.

	Bandwidth	Modu.	Code	Tput:Kbps
Ring	8 kHz	QPSK	1/2	7.41
Watch	3 kHz	QPSK	1/2	2.23

(2) Tabletop Communication:

Multicast communication is often useful – a group picture at a restaurant needs to be shared with everyone in the group; presentation slides need to be shared in a meeting. We envision placing all phones on the table, near each other, and performing one vibratory multicast. Figure 5.20 shows the outcome of such an experiment – we used the stylus to touch different locations on a table, while two microphone receivers were at fixed locations on this otherwise empty table. Even at nearly 2 feet away, the throughput is around 4 kbps (the X-axis has duplicate values since there were multiple distinct locations at the same distance from the microphone).

(3) P2P Money Transfer:

In developing regions, mobile payments may be viable with basic phones with vibra-motor and microphones. Perhaps a USB stick can transfer data to phones/tablets on physical contact. Such apps obviously need higher data rates and some may require real-time operation. Table 5.2 shows possibilities when vibra-motor on a stylus and a smartphones are touched to microphones. We have also built a demonstration of a real-time music streaming system over vibrations.

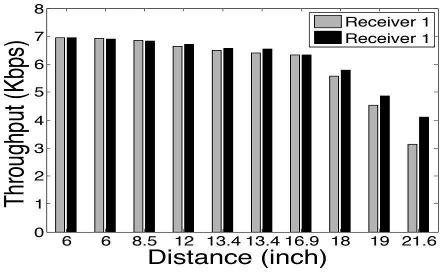


Figure 5.20: Throughput against varying tabletop range.

Table 5.2: Performance of the vibratory communication prototype with stylus and mobile phone.

	Bandwidth	Modu.	Code	Tput:Kbps
Stylus	12 kHz	16 QAM	2/3	29.19
Phone	12 kHz	16 QAM	2/3	26.13

5.6 Related Work

(1) Vibratory Communication: Authors in [119] and [85] were the first to conceive the idea of communicating through physical vibrations. They both encode vibrations through (ON/OFF) Morse code, with pulse durations of around one second (i.e., 1 bits/s). This is adequate for applications like secure pairing between two smartphones, or sending a tiny URL over tens of seconds. Our prior work in NSDI 2015 [129] developed a fuller vibratory radio through multi-frequency modulation, self-jamming-based security, and resonance braking, ultimately translating to 200 bits/s. *Ripple-II* is a push-forward of the *Ripple* project, but with microphone as the receiver, and augmented with a new PHY/MAC layer offering 150x throughput gain. *Ripple-II* still preserves *Ripple*'s security properties via self-sound cancellation.

Dhwani [1] and Chirp [149] address conceptually similar problems, although on the acoustic platform; vibra-motors bring about new set of challenges and opportunities. Technologies like Bump [120, 121, 122, 123, 124, 125, 126] use accelerometer/vibrator-motor responses to facilitate secure pairing between devices. TagTile [150] uses high-frequency sound to achieve association between phones and point-of-sale devices. However, these techniques are primarily designed for few bits of exchange; *Ripple-II* aims high bitrate transmission with the same ease as Bump and Tagtile. Further, as indicated by researchers [119, 127], the lack of the dynamic secret message in Bump-like techniques makes them less secure in the wild. These modes also require Internet connectivity and trusted third party servers to function, none of which is needed in *Ripple-II*.

(2) Vibration Generation and Sensing: Creative research in the domain of haptic feedback has investigated the state-of-the-art in electro-mechanical vibrations [115, 116]. Applications in assisted learning, touch-augmented environments, and haptic learning have used vibrations for communication to humans [112, 113, 114, 115, 116]. However, the push for high communication data rates between vibrators and microphones/accelerometers is unexplored to the best of our knowledge. Off late, personal/environment sensing on mobile devices has gained research attention. Applications like (sp)iPhone [117] and TapPrints [118] demonstrate the ability to infer keystrokes through background motion sensing. While many more efforts are around activity recognition from vibration signatures, this chapter aims to modulate vibration for communication.

5.7 Chapter Summary

Ripple-II is an attempt to enable touch-based vibratory communication between a vibra-motor and a microphone. We develop a vibratory radio at the PHY and MAC layer, and explore a few possible applications in authentication, device to device streaming, and tabletop communication. While additional work is needed to attain maturity, we believe this chapter is a concrete step toward demonstrating an alternative communication mode, that has remained relatively unexplored in the past.

Chapter 6

Recovering Voice from Vibrations

6.1 Overview

Vibration motors, also called “vibra-motors”, are small actuators embedded in all types of phones and wearables. These actuators have been classically used to provide tactile alerts to human users. This chapter identifies the possibility of using vibra-motors as a sound sensor, based on the observation that the same movable mass that causes the pulsation, should also respond to changes in air pressure. Even though the vibra-motor is likely to be far less sensitive compared to the (much lighter) diaphragm of an actual microphone, the question we ask is: *To what fidelity can the sound be reproduced?*

Even modest reproduction could prompt new applications and threats. On the one hand, wearable devices like fitbits, that otherwise do not have a microphone, could now respond to voice commands. Further, in devices that already have microphones, perhaps better SNR could be achieved by combining the uncorrelated (noise) properties of the vibra-motor and microphone. On the other hand, leaking sound through vibra-motors opens new side-channels – a malware that has default access to a phone’s vibra-motor may now be able to eavesdrop into every phone conversation. Toys that have vibra-motors embedded could potentially listen into the ambience. This chapter is an investigation into the vibra-motor’s efficacy as a sound sensor, speech in particular.

Our work follows a recent line of work in which motion sensors in smartphones have been shown to detect sound. Authors of Gyrophone [152] first demonstrated the feasibility of detecting sound signals from the rotational motions of smartphone gyroscopes. A recent work [153] reported how accelerometers may also be able to detect sound, in fact, classify spoken keywords such as “OK Google” or “Hello Siri”. Authors rightly identified the applicability to continuous sound sensing – the energy-efficient accelerometer could always stay active,

This chapter revises the publication “Listening through a Vibration Motor,” in MobiSys 2016 [151].

and turn on the energy-hungry microphone only upon detecting a keyword. While certainly useful, we observe that these systems run pattern recognition algorithms on the features of the signals. The vocabulary is naturally limited to less than three keywords, trained by a specific speaker. *VibraPhone* is attempting a different problem altogether – instead of learning a motion signature, it attempts to reconstruct the inherent speech content from the low-bandwidth, highly distorted output of the vibra-motor. Hence, there are no vocabulary restrictions, and the output of *VibraPhone* should be decodable by speech-to-text softwares.

As a first step toward converting a vibra-motor into a sound sensor, *VibraPhone* exploits the notion of *reverse electromotive force* (back-EMF) in electronic circuits. Briefly, the A/C current in the vibra-motor creates a changing magnetic field around a coil, which in turn causes the vibra-motor mass to vibrate. However, when an external force impinges on the same mass – say due to the pressure of ambient sound – it causes additional motion, translating into a current in the opposite direction. This current, called back-EMF, can be detected through an ADC after sufficient amplification. Of course, the signal extracted from the back-EMF is noisy and at a lower bandwidth than human speech. However, given that human speech obeys an "acoustic grammar", we find an opportunity to recover the spoken words even from the back-EMF's signal traces. *VibraPhone* focuses on exactly this problem, and develops a sequence of techniques, including *spectral subtraction, energy localization, formant extrapolation, and harmonic reconstruction*, to ultimately distill out legible speech.

Our experimentation platform is both a Samsung smartphone and a custom circuit that uses vibra-motor chips purchased online (these chips are exactly the ones used in today's phones and wearables). We characterize the extent of signal reconstruction as a function of the loudness of the sound source. Performance metrics are defined by the accuracy with which the reconstructed signals are intelligible to humans and to (open-source) automatic speech recognition softwares. We use the smartphone microphone as an upper bound, and for fairness, record the speech at the same sound pressure level (SPL) [154, 155, 156] across all the devices. We experiment across a range of scenarios within our university building, and observe that results are robust/useful when the speaker is less than 2 meters from the vibra-motor.

Finally, we emphasize that smartphone vibra-motors cannot be used as microphones today, primarily because the actuator is simply not connected to an ADC. To this end, launching side-channel attacks is not immediate. However, as discussed later, we find that enabling the listening capability requires almost trivial rewiring (just soldering at four clearly visible junctions). This chapter sidesteps these immediacy questions and concentrates on the core nature of the information leakage. One intent is for this work to draw attention to the permission policies on vibra-motors, which today are open to all apps by default. We have made various audio demos of *VibraPhone* available on our website [157] – we request the readers to listen to them to better experience the audio effects and reconstructions. In closing, the main contributions in this chapter may be summarized as:

- *Recognizing that ambient sound manifests itself as back-EMF inside vibra-motor chips.* This leads to an actuator becoming a sound sensor with minimal changes to the current mobile device hardware.
- *Designing techniques that exploit constraints and structure of human speech to decode words from a noisy, low-bandwidth signal.* Building the system on a smartphone and custom hardware platform, and demonstrating decoding accuracy of up to 88% when a male user is speaking in normal voice near his phone.

The rest of the chapter expands on these contributions. We begin with a brief introduction to vibra-motors and our hardware platform.

6.2 Understanding Vibra-Motors

As explained in Chapter 4, a vibra-motor is an electro-mechanical device that moves a magnetic mass rhythmically around a neutral position to generate vibrations [129]. While there are various kinds of vibra-motors, a popular one is called Linear Resonant Actuators (LRA) shown in Figure 6.1. With LRA, vibration is generated by the linear movement of the magnetic mass suspended near a coil, called the "voice coil". Upon applying AC current to the motor, the coil also behaves like a magnet (due to the generated electromagnetic field) and causes the mass to be attracted or repelled, depending on the direction of the current. This generates vibration at the same frequency as the input AC signal, while the amplitude of vibration is dictated by the signal's peak-to-peak voltage. Thus LRAs offer control on both the magnitude and frequency of vibration. Most smartphones today use LRA-based vibra-motors.

6.2.1 Sound Sensing through Back-EMF

Back-EMF is an electro-magnetic effect observed in magnet-based motors when relative motion occurs between the current carrying armature/coil and the magnetic mass's own

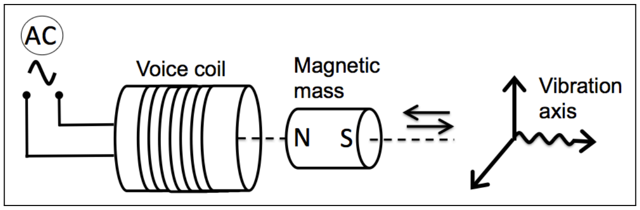


Figure 6.1: Basic sketch of an LRA vibra-motor.

field. According to Faraday's law of electromagnetic induction [145], this changing magnetic flux induces an electromotive force in the coil. Lenz's law [146] says this electromotive force acts in the reverse direction of the driving voltage, called *back-EMF of the motor*. As the rate of change of the magnetic flux is proportional to the speed of the magnetic mass, the back-EMF serves as an indicator of the extraneous vibration experienced by the mass.

Since sound is a source of external vibration, the movable mass in the vibra-motor is expected to exhibit a (subtle) response to it. Our experiments show that, when the vibra-motor is connected to an ADC, the back-EMF generated by the ambient sound can be recorded. This is possible even when the vibra-motor is passive (i.e., not pulsating to produce tactile alerts). We call this ADC output *vibra-signal* to distinguish it from the microphone signal that we will later use as a baseline for comparison. We now describe our platform to record and process the vibra-signal.

6.2.2 Experiment Platform

Custom hardware setup: Today's smartphones offer limited exposure/API to vibra-motor capabilities and other hardware components (e.g., amplifiers). To bypass these restrictions, we have designed a custom hardware setup using off-the-shelf LRA vibra-motor chips connected to our own ADC and amplifier. Figure 6.2 shows our setup – we mount this vibration motor adjacent to a standard microphone that serves as a comparative baseline. The vibra-signal is amplified and sampled at 16 kHz. Test sounds include live speech from humans at varying distances, as well as sound playbacks through speakers at varying loudness levels.

Smartphones: While the custom hardware offers better programmability, we also use a smartphone setup to understand the possibilities with today's systems. Figure 6.3 shows our prototype – terminals of the built-in vibra-motor of a Samsung Galaxy S-III smartphone is connected to the audio line-in input port with a simple wire. The rewiring is trivial – for someone familiar with the process, it can be completed in less than 10 minutes. Once rewired, we collect the samples of the vibra-signal from the output channels of the earphone jack, using our custom Android application.

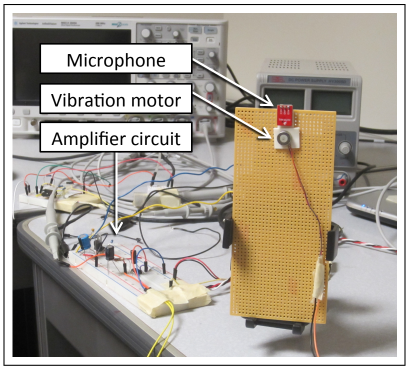


Figure 6.2: The custom hardware setup with collocated vibration motor and microphone.

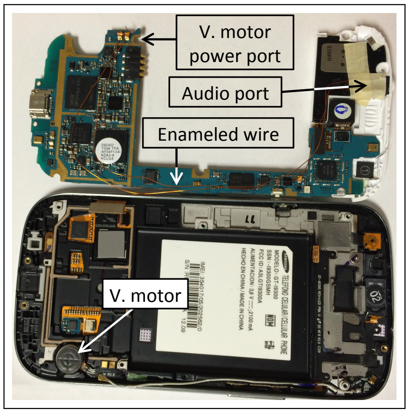


Figure 6.3: The smartphone setup with a simple wire connected between the vibra-motor's output to the audio line-in port.

Electromagnetic Coupling: We conduct a microbenchmark test to verify that the vibration motor signal is not influenced by the electromagnetic coupling from the nearby microphone or speakers in our test setup. We remove the speakers and microphones from the test environment and directly record human speech with a vibration motor (find sample clips at project website [157]). Later we compare them with the recordings of the standard test setup to find no noticeable difference in signal quality.

6.3 Sounds and Human Speech

This section is a high-level introduction to speech production in humans, followed by a discussion on the structure of speech signals.

6.3.1 Human Speech Production

Human speech can be viewed as periodic air waves produced by the lungs, modulated through a sequence of steps in the throat, nose, and mouth. More specifically, the air from the lungs first passes through the *vocal cords* – a pair of membranous tissue – that constricts or dilates to block or allow the air flow (Figure 6.4). When the vocal cords are constricted, the vibrations induced in the air-flow are called *voiced* signals. The voiced signals generate high energy pulses – in the frequency domain, the signal contains a fundamental frequency and its harmonics. All vowels and some consonants like "b" and "g" are sourced in voiced signals.

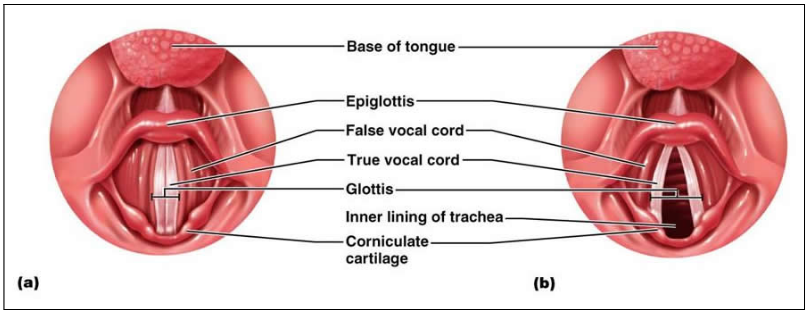


Figure 6.4: The vocal cords constricted in (a) and dilated in (b), creating *voiced* and *unvoiced* air vibrations, that are then shaped by the glottis and epiglottis.

On the other hand, when the vocal cords dilate and allow the air to flow through without heavy vibrations, the outcome is called *unvoiced* signals. This generates sounds similar to noise, and is the origin of certain consonants, such as "s", "f", "p", "k", "t". Both voiced and unvoiced signals then pass through a flap of tissue, called *glottis*, which further pulsates to add power to the signal as well as distinctiveness to an individual's voice. These *glottal pulses* travel further and are finally modulated by the oral/nasal cavities to produce fine-tuned speech [158]. The overall speech production process is often modeled as a "source-filter" in literature, essentially implying that the human trachea/mouth applies a series of filters to the source sound signal. This source-filter model will later prove useful, when *VibraPhone* attempts to reconstruct the original speech signal.

6.3.2 Structure in Speech Signals

While the above discussions present a biological/linguistics point of view, we now discuss how they relate to the recorded speech signals and their structures. Figure 6.5 shows the spectrogram when a human user pronounces the alphabets "sa" – the signal was recorded

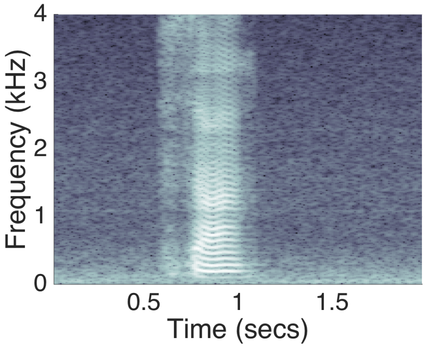


Figure 6.5: The spectrogram of the spoken consonant “s” followed by the vowel “a” recorded with microphone.

through a smartphone microphone (not a vibra-motor).[1] Although a toy case, the spectrogram captures the key building blocks of speech structure. We make a few observations that will underpin the challenges and the designs in the rest of the chapter.

- The first visible signal (between 0.6 and 0.75 seconds) corresponds to the *unvoiced* component, the consonant “s”. This signal is similar to noise with energy spread out rather uniformly across the frequency band. The energy content in this signal is low to moderate.

- The second visible signal corresponds to the vowel “a” and is an example of the *voiced* component. The signal shows a low fundamental frequency and many harmonics all the way to 4 kHz. Fundamental frequencies are around 85–180Hz for males and 165–255 Hz for females [159]. The energy content of this signal is far stronger than the *unvoiced* counterpart.

- Within the *voiced* signal, the energy content is higher in the lower frequencies. These strong low-frequency components determine the intelligibility of the spoken phonemes (i.e. the perceptually distinct units of sound [160]), and are referred to as *formants* [161]. The first two formants (say, $F1$, $F2$) remain between 300–2500 Hz and completely form the sound of the vowels, while some consonants have another significant formant, $F3$, at a higher frequency. Figure 6.6 shows examples of two vowel formants – “i” and “a” – recorded by the microphone.

In extracting human speech from the vibra-motor’s back-EMF signal, *VibraPhone* will need to identify, construct, and bolster these formants through signal processing.

[1] The Y-axis shows up to 4 kHz, since normal human conversation in non-tonal languages like English is dominantly confined to this band.

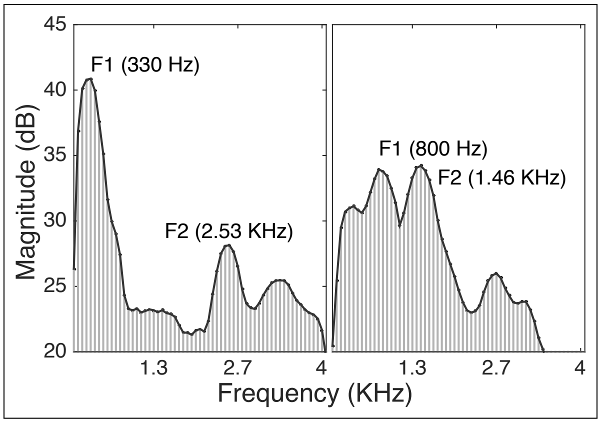
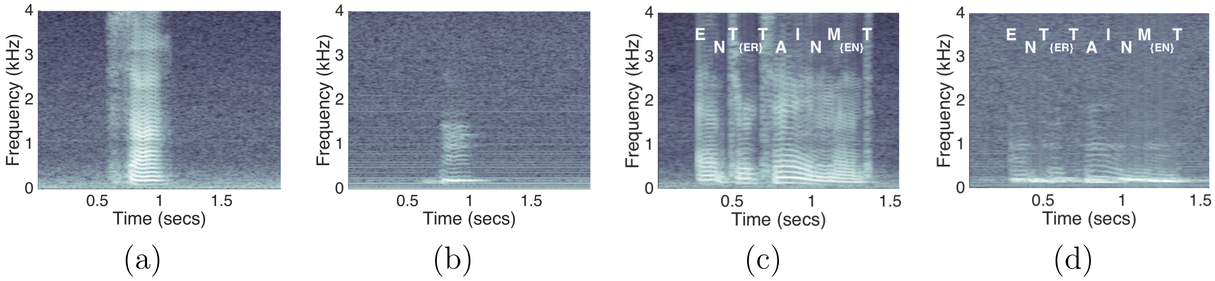


Figure 6.6: The locations of the first two formants (F1 and F2) for (left) the vowel sound *"i"* and (right) the vowel sound *"a"*, both recorded with a microphone.

6.4 Challenges

Figure 6.7(a,b) compares the spectrogram of the microphone and the vibra-motor for the same spoken phoneme, "sa". Figure 6.7(c,d) shows the same comparison for a full word, namely, "entertainment". The reader is encouraged to listen to these sound clips at our project website [157]. Evidently, the vibra-motor's response is weak and incomplete, and on careful analysis, exhibits various kinds of distortions even where the signal is apparently strong. The goal in this chapter is to reconstruct, to the extent possible, the left columns of Figure 6.7 from the right columns. We face four key challenges discussed next.



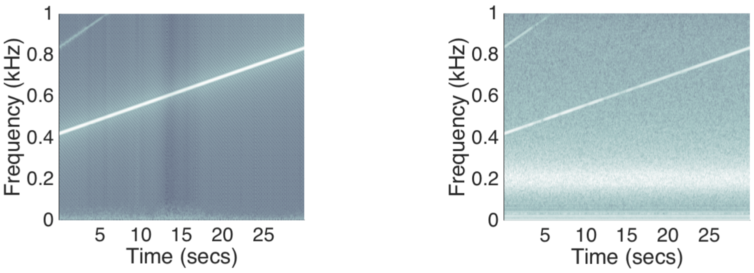
(a) (b) (c) (d)

Figure 6.7: The spectrogram for "sa" as recorded by: (a) the microphone and (b) the vibra-motor. The spectrogram for the full word "entertainment" as recorded by: (c) the microphone and (d) the vibra-motor. The vibra-motor's response is weak and partially missing.

6.4.1 Over-Sensitivity at Resonance Frequency

All rigid objects tend to oscillate at a fixed natural frequency when struck by an external force. When the force is periodically repeated at a frequency close to the object's natural frequency, the object shows exaggerated amplitude of oscillation – called *resonance* [162]. Resonance is often an undesirable phenomenon, destabilizing the operation of an electro-mechanical device. Microphones, for example, carefully avoid resonance by designing its diaphragm at a specific material, tension, and stiffness – that way, the resonance frequencies lie outside the operating region [163, 164]. In some cases, additional hardware is embedded to damp the vibration at the resonant frequencies [164].

Unfortunately, vibra-motors used in today's smartphones exhibit sharp resonance between 216 to 232 Hz, depending on the mounting structure. Some weak components of *speech formants* are often present in these bands – these components get amplified, appearing as a *pseudo-formant*. The *pseudo-formants* manifest as unexpected sounds within uttered words, affecting intelligibility. The impact is exacerbated when the fundamental frequency of the voiced signal is itself close to the resonant band – in such cases, the sound itself gets garbled. Figure 6.8 shows the effect of resonance when the vibration motor is sounded with different frequency tones in succession (called a *Sine Sweep* [165, 142]. Observe that for all tones in the *Sine Sweep*, the vibra-motor exhibited appreciable response in the resonance band. This is because the tones have some frequency tail around the 225 Hz, and this always gets magnified. The microphone exhibits no such phenomenon. *VibraPhone* will certainly need to cope with resonance.

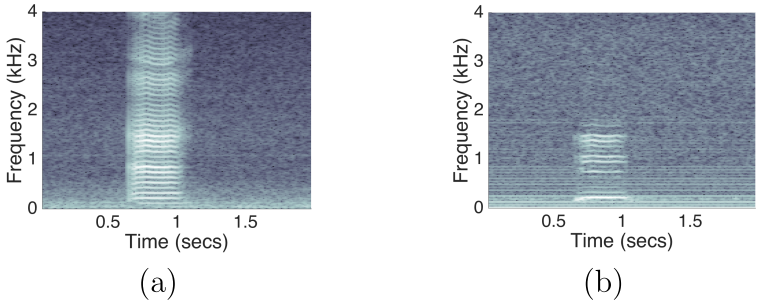


(a) (b)

Figure 6.8: The spectrogram of (a) the microphone and (b) the vibra-motor, in response to a *Sine Sweep* (i.e., tones played at increasing narrow band frequencies). The vibration motor signal shows an over-sensitive resonance frequency band near 220 Hz.

6.4.2 High-Frequency Deafness

The vibra-motor's effective diaphragm – the area amenable to the impinging sound – is around 10mm, almost 20x larger than that of a typical MEMS microphone (0.5 mm). This makes the vibration motor directional for the high-frequency sounds, i.e., the high frequencies arriving from other directions are suppressed, somewhat like a directional antenna. Unfortunately, human voices contain lesser energy at frequencies higher than 2 kHz, thereby making the vibra-motor even less effective in "picking up" these sounds. Some consonants and some vowels – such as "i" and "e" – have formants close to or higher than 2 kHz, and are severely affected. Figure 6.9 compares the spectrogram when just the vowel "*a*" was spoken – evidently, the vibra-motor is almost "deaf" to higher frequencies.



(a)

(b)

Figure 6.9: The spectrogram of the spoken vowel "*a*" recorded with (a) a microphone and (b) a vibration motor. The vibra-motor exhibits near-deafness for frequencies > 2 kHz.

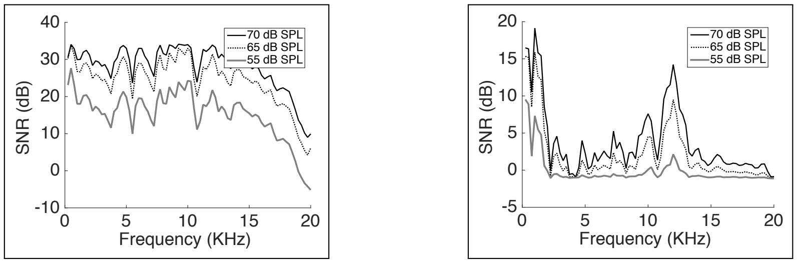
6.4.3 Higher Energy Threshold

A microphone's sensitivity, i.e., the voltage produced for a given sound pressure level, heavily depends on the weight and stiffness of its diaphragm. The spring-mass arrangement of the vibra-motor is considerably more stiff, mainly due to the heavier mass and high spring constant. While this is desirable for a vibration actuator, it is unfavorable to sound sensing. Thus, using the actuator as a sensor yields low sensitivity in general, and particularly to certain kinds of low-energy consonants (like *f, s, v, z*), called *fricatives* [166]. The effect is visible in Figure 6.7 (a,b) – the *fricative* consonant "*s*" goes almost undetected with vibra-motors.

6.4.4 Low Signal-to-Noise Ratio (SNR)

In any electrical circuit, *thermal noise* is an unavoidable phenomenon arising from the Brownian motion of electrons in resistive components. Fortunately, the low 26 ohm terminal resistance in vibra-motors leads to 10 dB lower thermal noise than the reference MEMS microphone. However, due to low sensitivity, the strength of the vibra-signal is significantly lower, resulting in poor SNR across most of the spectrum. Figure 6.10 compares the SNR at different sound pressure levels – except around the resonance frequencies, the SNR of the vibra-signal is significantly less compared to the microphone.

Sound Pressure Level (SPL) is a metric to measure the effective pressure caused by sound waves with respect to a reference value, and is typically expressed in dBSpl [155]. This gives a standard estimate of the sound field at the receiver, irrespective of the location of the sound source.



(a)

(b)

Figure 6.10: The SNR of (a) the microphone and (b) the vibra-motor at various frequencies for varying sound pressure levels (dB SPL). Note the unequal Y-axis range.

6.5 System Design

Our system design is modeled as a *source-filter*, i.e., we treat the final output of the vibra-motor as a result of many filters applied serially to the original air-flow from the lungs. Figure 6.11 illustrates this view, suggesting that an ideal solution should perform two broad tasks: (1) "undo" the vibra-motor's distortions for signal components that have been detected, and (2) reconstruct the undetected signals by leveraging the predictable speech structure in conjunction with the slight "signal hints" picked up by the vibra-motor. *VibraPhone* realizes these tasks through two corresponding modules, namely, *signal pre-processing* and *partial speech synthesis*. We describe them next.

6.5.1 Signal Pre-Processing

All of our algorithms operate on the frequency domain representation of the signal. Therefore, we first convert the amplified signal to the time-frequency domain using the *Short Time*

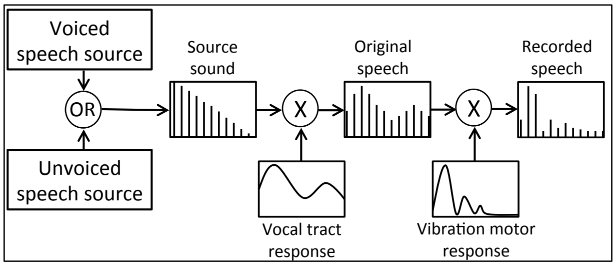


Figure 6.11: The source-filter model of the speech generation and recording.

Fourier Transform (STFT), which basically computes the complex FFT coefficients from 100 millisecond segments (80% overlapped, Hanning windowed) of the input time signal. The result is a 2D matrix that we call time-frequency signal and illustrated in Figure 6.12 – each column is a time slice and each row is a positive frequency bin. We will refer to this matrix for various explanations.

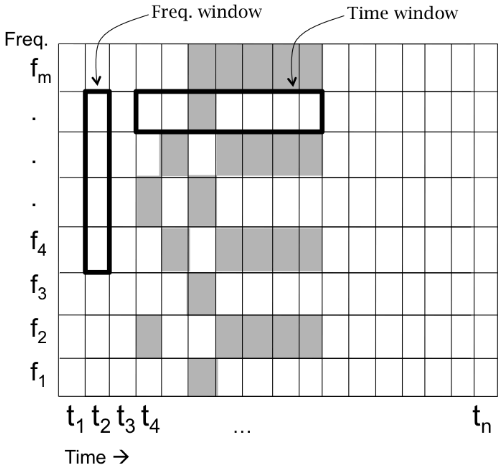


Figure 6.12: 2D time-frequency matrix.

(1) Frequency Domain Equalization:

When a microphone is subject to a *Sine Sweep* test, the frequency response is typically flat, meaning that the microphone responds almost uniformly to each frequency component. The vibra-motor's response, on the other hand, is considerably jagged, and thereby induces distortions into the arriving signal. Figure 6.13 shows a case where the vowel "u" is recorded by both the microphone and vibra-motor. The vibra-motor distortions on "u" are quite dramatic, altering the original formants at 266 and 600 Hz to new formants at 300 Hz and 1.06 kHz. In fact, the altered formants bear resemblance to the vowel "aa" (as in "father"), and in reality, do sound like it. More generally, the vibra-motor's frequency response exhibits this rough shape, thereby biasing all the vowels to the sound of "aa" or "o".

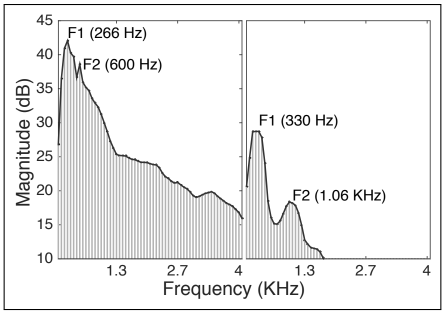


Figure 6.13: Formants of vowel *"u"* recorded through (left) microphone and (right) vibra-motor. The vibra-motor introduces a spurious formant near 1 kHz.

Fortunately, the frequency response of the vibra-motor is only a function of the device and does not change with time (at least until there is wear and tear of the device). We tested this by computing the correlation of the *Sine Sweep* frequency response at various sound pressure levels – the correlation proved strong, except for a slight dip at the resonant frequencies due to the nonlinearities. Knowing the frequency response, we apply an equalization technique, similar to channel equalization in communication. We estimate the inverse gain by computing the ratio of the coefficients from the microphone and the vibra-motor, and multiply the inverse gain with the frequency coefficients of the output signal.

(2) Background Noise Removal:

Deafness in vibra-motors implies that the motor's response to high-frequency signals (i.e., > 2 kHz) is indistinguishable from noise. If this noise exhibits a statistical structure, a family of *spectral subtraction* algorithms can be employed to improve SNR. However, two issues need attention. (1) The pure noise segments in the signal needs to be recognized, so that its statistical properties are modeled accurately. This means that noise segments must be discriminated from speech. (2) Within the speech segments, *voiced* and *unvoiced* segments must also be separated so that spectral subtraction is only applied on the *voiced* components. This is because *unvoiced* signals bear noise-like properties and spectral subtraction can be detrimental.

To reliably discriminate the presence of speech segments, we exploit the exaggerated behavior in the resonance frequency band. We consistently observed that speech brings out heavy resonance behavior in vibra-motors, while noise elicits a muted response. Thus, resonance proved to be an opportunity. Once speech is segregated from noise, the next step is to isolate the *voiced* components in speech. For this, we leverage its well-defined harmonic structure. Recall the 2D matrix in Figure 6.12. We consider a time window and slide it up/down to compute an autocorrelation coefficient across different frequencies. Due to the repetition of the harmonics, the autocorrelation spikes periodically, yielding robust detection accuracy. When autocorrelation does not detect such periodic spikes, they are deemed as the *unvoiced* segments.

The final task of spectral subtraction is performed on the *voiced* signal alone. For a given *voiced* signal (i.e., a set of columns in the matrix), the closest noise segments in time are selected – these noise segments are averaged over a modest time window. Put differently, for every frequency bin, the mean noise floor is computed, and then subtracted from the corresponding bin in the *voiced* signal. For zero-mean Gaussian noise, this does not offer any benefit, however, the noise is often not zero-mean. In such cases, the SNR improves and alleviates the deafness. Figure 6.14 shows the beneficial effect of spectral subtraction when "yes" is spoken.

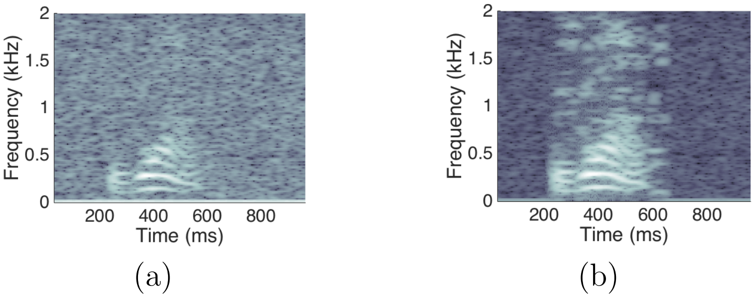


Figure 6.14: The spectrogram of the spoken word "yes" (a) before and (b) after the spectral subtraction.

(3) Speech Energy Localization:

Observe that noise removal described above brings the *mean noise* to zero, however, noise still exists and the SNR is still not adequate. In other words, deafness is still a problem. However, now that noise is zero-mean and Gaussian, there is an opportunity to exploit its diversity to further suppress it. Even localizing the speech signal energy in the spectrogram would be valuable, even if the exact signal is not recovered in this step.

Our core idea is to average the signals from within a frequency window, and slide the frequency window all the way to 10 kHz. Referring to the 2D matrix, we compute the average of W elements in each column (W being the window size), and slide the window vertically; the same operation is performed for each column. Each element is a complex frequency coefficient, containing both the signal and the noise. With sufficiently large W, the average converges to the average of the signal content in these elements since the (average) noise sum up to zero. Mathematically, if C_i denotes the signal at frequency f_i, and $C_i = S_i + N_i$, where S_i is the speech signal and N_i the noise, then the averaged C_i^* is computed as:

$$C_i^* = \frac{1}{W} \sum_{f=i-\frac{W}{2}}^{i+\frac{W}{2}} C_i = \frac{1}{W} \sum_{f=i-\frac{W}{2}}^{i+\frac{W}{2}} S_i + \frac{1}{W} \sum_{f=i-\frac{W}{2}}^{i+\frac{W}{2}} N_i \tag{6.1}$$

Since the term $\sum N_i$ is zero-mean Gaussian, it approaches zero for larger W, while the $\frac{1}{W} \sum S_i$ term is simple smoothing. For every frequency bin, we normalize the C_i^* values over a time window so that they range between $[0, 1]$. The result is a 3D contour map, where the locations of higher elevations, i.e., hills, indicate the presence of speech signals. We identify the dominant hills and *zero force* all areas outside them. This is because speech signals always exhibit a large time-frequency footprint, since human voice is not capable of producing sounds that are narrow in frequency and time. Figure 6.15 illustrates the effect of this scheme – the dominant hills are demarcated as the location of speech energy. Evidently, the improvement is conspicuous after this energy localization step.

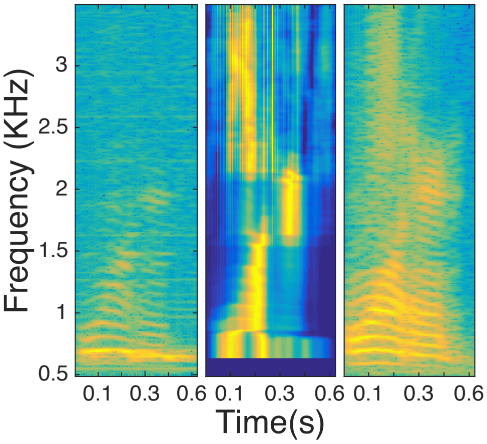


Figure 6.15: Readers are requested to view this figure in color: (left) Raw vibra-motor signal. (center) The output of the *speech energy localization* makes the signal energy visible through a heat-map like contour. (right) The corresponding microphone signal bearing good resemblance to the energy locations.

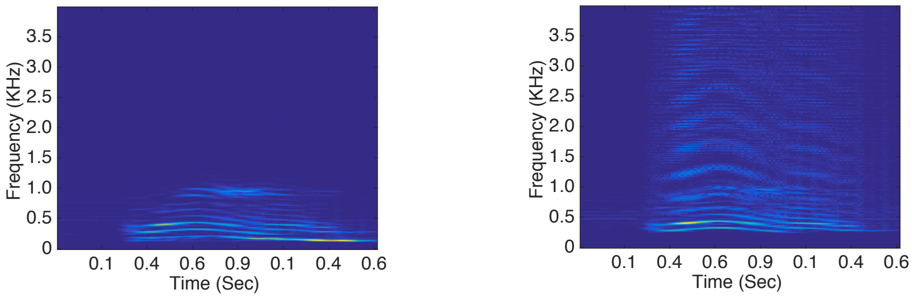
6.5.2 Partial Speech Synthesis

Once the vibra-motor output has been pre-processed, the structure of speech can now be leveraged for signal recovery – we describe our approach next.

(1) Voice Source Expansion:

After the localization step above, we know the location of speech energy (in time-frequency domain), but we do not know the speech signal. In attempting to recover this signal, we exploit the opportunity that the fundamental frequencies in speech actually manifest in higher-frequency harmonics [167, 168]. Therefore, knowledge of the lower-fundamental frequencies can be *expanded* to reconstruct the higher frequencies. Unfortunately, the actual fundamental frequency often gets distorted by the resonant bands.

As a workaround, we use the relatively high SNR signals in the range [250, 2000 Hz] to synthesize the voice source signal at higher frequencies. Synthesis is essentially achieved through careful replication. Specifically, the algorithm copies the coefficient $C_{t,f}$, where t is the time segment and f is the frequency bin of the time-frequency signal, and adds it to $C_{t,kf}$ for all integer k, such that kf is less than the Nyquist frequency. Here integer k indicates the harmonic number for the frequency f. Intuitively, we are copying the harmonics from the reliable range, and replicating them at the higher frequencies. As shown in Figure 6.16, this only synthesizes the *voiced* components (recall the harmonics are only present in the *voiced* signals). For *unvoiced* signals, we blindly fill in the deaf frequencies with copies of the lower-frequency signals.



(a)

(b)

Figure 6.16: Result of source expansion for the *voiced* signal components: (a) Raw vibra-signal and (b) after harmonic replication. Readers are requested to view this figure in color.

(2) Speech Reconstruction:

Recall that the mouth and nasal cavities finally modulate the air vibrations – this can be modeled as weights multiplied to the fundamental frequencies and their harmonics. While we do not know the values of these weights, the location of the energies – computed from the 3D contour hills – is indeed an estimate. We now utilize this location estimate as an *energy mask*. As a first step, we apply an exponential decay function along the frequency axis to model the low intensity of natural speech at the higher frequencies. Then the energy mask is multiplied with this modified signal, emulating an adaptive gain filter. As this also improves the SNR of the unvoiced section of the speech, we apply a deferred spectral subtraction method on these segments to further remove the background noise. Finally, we convert this resultant time-frequency signal to time domain using inverse short time Fourier transform (ISTFT). Figure 6.17 compares the output against the microphone and the raw vibra-motor signal.

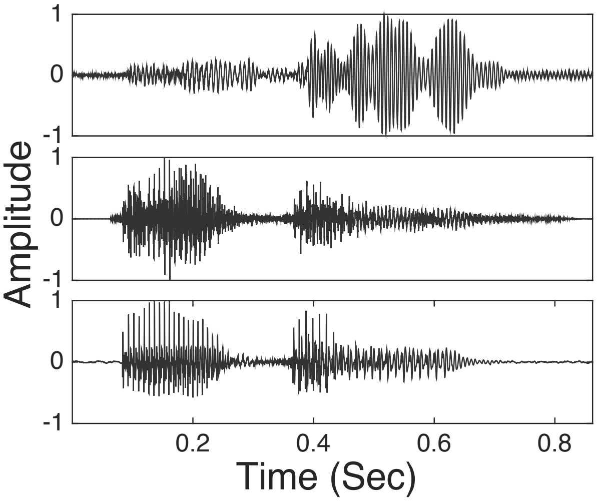


Figure 6.17: Word "often" as manifested in the (top) raw vibra-motor signal, (middle) after *VibraPhone*'s processing, and (bottom) microphone signal.

6.6 Evaluation

Section 6.2.2 described the two experimentation platforms for our system, namely the custom hardware and the Samsung Galaxy smartphone. In both cases, we evaluate *VibraPhone*'s speech intelligibility against the performance of the corresponding microphone. In the custom hardware, the microphone is positioned right next to the vibra-motor, while in the smartphone, their locations are modestly separated. We generate the speech signals using a text-to-speech (TTS) utility available in OS X 10.9, and play them at different volumes through a loudspeaker. The position/volume of the loudspeaker is adjusted such that the sound pressure levels at the vibra-motor and the microphone are equal. The accent and intonation of the TTS utility also does not affect the experiment since both *VibraPhone* and the microphone hear the same TTS speech. The content of the speech is drawn from Google's Trillion Word Corpus [21] – we pick 2000 most frequent words, which is prescribed as a good benchmark [22].

6.6.1 Methodology and Metrics

We perform automatic and manual speech recognition experiments as follows.

(1) Automatic Speech Recognition (ASR):
In ASR, a software programmatically converts the time domain speech signal to text. ASR tools typically have three distinct components: (a) an acoustic model, (b) a pronunciation dictionary, and (c) a language model. The *acoustic model* is a trained statistical model (e.g., HMM, neural networks, etc. [169, 170]) that maps segments of the input waveform to a sequence of phonemes. These phonemes are then looked up in the *pronunciation dictionary*, which lists the candidate words (along with their possible pronunciations) based on the matching phoneme sequence. Among these candidates, the most likely output is selected using a grammar or a *language model*.

Our ASR tools is the open-source *Sphinx4* (pre-alpha version) library published by CMU [6, 23]. The acoustic model is sensitive to the recording parameters, including the bandwidth and the features of the microphone. Such parameters do not apply to vibra-motors, so we used a generic acoustic model trained with standard microphone data. This is not ideal for *VibraPhone*, and hence, the reported results are perhaps a slight underestimate of *VibraPhone*'s capabilities.

(2) Manual Speech Recognition (MSR):
We recruited a group of six volunteers from our department building – one native English speaker, one Indian faculty with English as first language, two Indian PhD students, and two Chinese PhD students. We played the vibra-motor and microphone outputs to all the participants simultaneously and collected their responses. In some experiments, volunteers were asked to *guess the word or phrase from the playback*; in other experiments, the volunteers were given a list of phrases and asked to pick the most likely one, including the option of "none of the above". All human responses were accompanied by a subjective clarity score – every volunteer expressed how intelligible the word was, even when he/she could not guess with confidence. Finally, in some experiments, volunteers were asked to guess first, and then guess again based on a group discussion. Such discussions served as a "prior" for speech recognition, and often the group consensus was different from the first individual guess.

(3) Metrics:
Across all experiments, 9 hours of sound was recorded and a total of 20,000 words were tested with ASR at various sound pressure levels (measured in dBSpl). For MSR, a total of 300 words and 40 phrases were played, resulting in more than 2000 total human responses. We report "Accuracy" as the percentage of words/phrases that were correctly guessed, and show its variation across different loudness levels (measured in dBSpl). We report "Perceived Clarity" as a subjective score reported by individuals, even when they did not decode the word with confidence. Finally, we report "Precision", "Recall", and "Fallout" for experi-

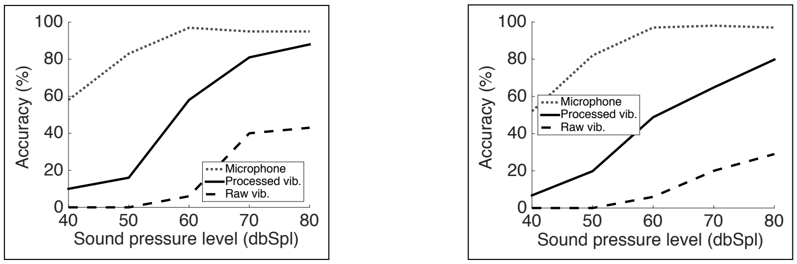
ments in which the users were asked to select from a list. Recall that *precision* intuitively refers to "*what fraction of your guesses were correct*", and *recall* intuitively means "*what fraction of the correct answers did you guess*". We now present the graphs, beginning with ASR.

6.6.2 Performance Results with ASR

(1) Accuracy vs. Loudness:

Custom Hardware: Figure 6.18(a) reports the accuracy with ASR as a function of the sound pressure level (dBSpl), a standard metric proportionally related to the loudness of the sound. *VibraPhone*'s accuracy is around 88% at 80 dBSpl, which is equivalent to the sound pressure experienced by the smartphone's microphone during typical (against the ear) phone call. The microphone's accuracy is obviously better at 95%, while the raw vibra-motor signal performs poorly at 43% (almost half of *VibraPhone*). Importantly, the pre-processing and the synthesis gains are individually small, but since intelligibility is defined as binary metric here, the improvement jumps up when applied together.

Once the loudness decreases at 60 dBSpl – comparable to a normal conversation 1 meter away from the microphone [171] – *VibraPhone*'s accuracy drops to ≈ 60%. At lower sound pressure level, the accuracy drops faster since the vibra-motor's sensitivity is inadequate for "picking up" the air vibrations. However, the accuracy can be improved with training the acoustic model with vibra-motors (recall that with ASR, the training is performed through microphones, which is unfavorable to *VibraPhone*).



(a)

(b)

Figure 6.18: Automatic recognition accuracy as a function of loudness for (a) the custom hardware, and (b) the Samsung smartphone.

Samsung Smartphone: Figure 6.18(b) plots the accuracy with ASR for the smartphone-based platform. *VibraPhone*'s performance is weaker compared to the custom hardware setup, although the difference is marginal – ASR output is still at 80% at 80 dBSpl. Admittedly, we are not exactly sure of the reason for this difference – we conjecture that the smartphone signal processing pipeline may not be exactly tuned to the vibra-motor like we have done in the custom case.

(2) Rank of the Words:

The accuracy results above counts only perfect matches between ASR's output and the actual spoken word. In certain applications, a list of possible words may also be useful, particularly when the quality of the speech is poor. We record the list of all predictions from ASR for each spoken word, played at 50 dBSpl. Figure 6.19 plots the CDF of the rank of the correct word in this list. At this relatively softer 50 dBSpl experiment, only ≈ 20% of the words are ranked at 1, implying exact match. In 41% of the cases, the words were within the top-5 of the list, and *top-10* presents a 58% accuracy.

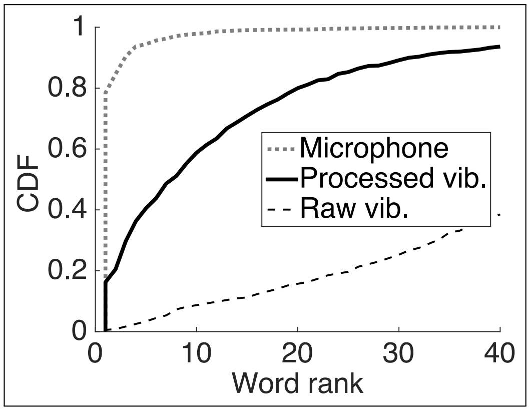
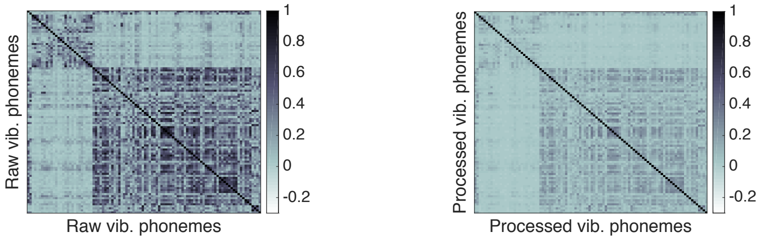


Figure 6.19: The CDF of word rank from ASR's prediction at 50 dBSpl for custom hardware.

(3) Phoneme Similarity:

The acoustic model we used with ASR is not ideal for *VibraPhone* – the impact is pronounced for distorted phonemes. Training ASR's acoustic model with the vibra-motor response is expected to offer improvements, but in the absence of that, we report a subjective overview of the entropy in different phonemes recorded by *VibraPhone*. In other words, we ask whether *autocorrelation* between the same phonemes is high and *cross correlation* across phonemes is low. We extract the STFT coefficients of the 100 phonemes (28 vowels and 72 consonants) from the International Phoneme Alphabet [172, 173] and use these coefficients as the features. We then calculate correlation coefficient of all pairs of phonemes in the list – Figure 6.20 presents the heat map. In case of raw vibra-signal in Figure 6.20(a), the (distorted) phonemes bear substantial similarity between each other, indicated by the multiple dark off-diagonal blocks. The two large darker squares in the figure represents the pulmonic (58 phonemes) and non-pulmonic (14 phonemes) consonant groups [174, 166]. However, with *VibraPhone*, Figure 6.20(b) shows substantial improvements. The autocorrelation is strong across the diagonal of the matrix, while the off-diagonal elements are much less correlated. This extends the possibility that a vibra-motor trained acoustic model could appreciably boost *VibraPhone*'s performance.



(a)

(b)

Figure 6.20: The heat-map shows the correlation of the frequency domain features of the phoneme sounds, recorded with custom vibration motor: (a) before processing and (b) after processing.

6.6.3 Performance Results with MSR

(1) Accuracy vs. Loudness:

Figure 6.21 shows the accuracy with manual speech recognition (MSR) in comparison to automatic (ASR). Unsurprisingly, the accuracy is around 20% more than ASR at higher loudness regimes (60 dBSpl or more) – the individuals guessed the words individually in these experiments. Using consensus from group discussion, the accuracy increases to 88% at 60 dBSpl. When the loudness is stronger, *VibraPhone* is comparable to microphones, both for custom hardware and smartphones.

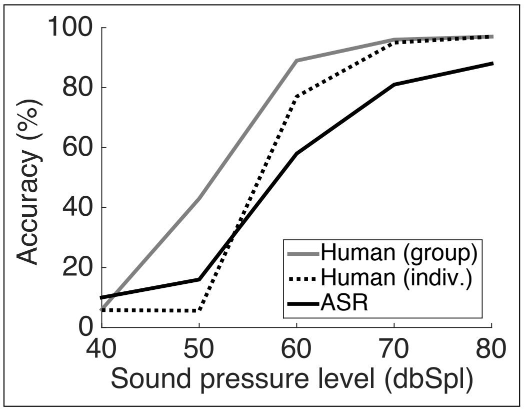


Figure 6.21: This plot compares the accuracy of human decoding with ASR. It shows the performance of the human decoders while working individually and as a collaborative team.

(2) Hot-Phrase Detection:

Figure 6.22 shows manual performance with "hot phrases", i.e., where the volunteer was asked to pick a phrase from the list that best matched the spoken phrase (the volunteer could also select none of the phrases). We provided a list of 10 written phrases before playing the positive and negative samples in arbitrary sequence. Example phrases were "turn left", "happy birthday", "start the computer", etc., and the negative samples were chosen with a comparable number of words and characters.

Figure 6.22(a) reports results from the custom hardware – volunteers almost perfectly identified the phrases and rejected the negative samples. However, when using the smartphone vibra-motor, *VibraPhone* failed to identify some positive samples – Figure 6.22(b) shows the outcome in relatively higher false negative values. Of course, the degradation is relative – the absolute detection performance is still quite high, with accuracy and precision at 0.83 and 0.90, respectively, for the processed vibra-signal.

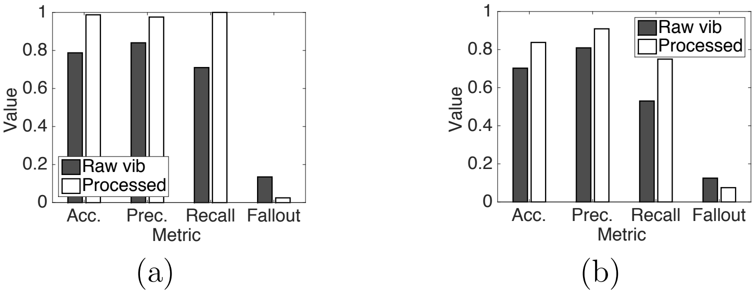


Figure 6.22: The accuracy, precision, recall, and fallout values for manual hot-phrase detection. The recording device is (a) the custom hardware and (b) the smartphone.

(3) Perceived Clarity:

Human volunteers also assigned a "clarity score" on a range of $[0, 10]$ to every word/phrase he/she listened to (a score of 10 indicated a perfectly intelligible word). Figure 6.23 plots the average clarity score of the correctly decoded samples and compares it between the vibration motor and the microphone. The subjective perception of clarity does not change for the microphone for sound pressure levels 50 dBSpl and above. While *VibraPhone*'s clarity is lower than the microphone's clarity in general, the gap reduces at higher loudness levels. At 80 dBSpl, the perceived clarity scores for microphones and *VibraPhone* are 9.1 and 7.6, respectively.

(4) Kinds of Words:

Figure 6.24 shows the top-10 and bottom-10 intelligible words from the ASR experiments. The font size is proportional to the decoding accuracy, indicating that "international" was

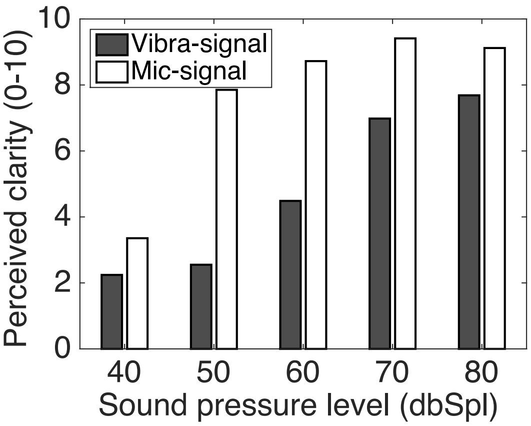
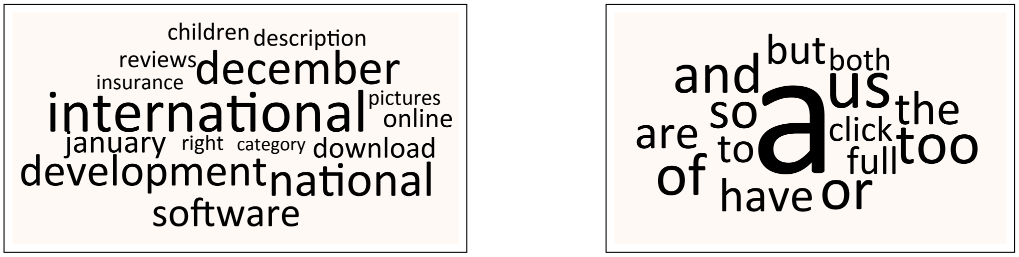


Figure 6.23: The perceived clarity of the correctly decoded speech recorded with microphone and vibration motor.

decoded correctly most frequently, while prepositions like “a”, “and”, “or” were consistently missed. Unsurprisingly, longer words are decoded with higher accuracy because of better interpolation between the partially decoded phonemes. Figure 6.25 quantifies this with ASR and MSR, respectively – words with 5+ characters are mostly multi-syllable, yielding improved recognition.

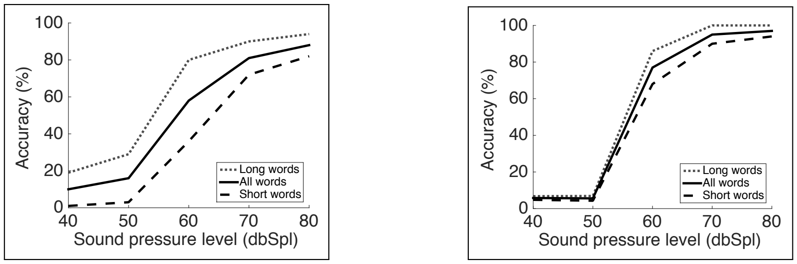


(a) (b)

Figure 6.24: (a) The top-10 words that are correctly decoded by ASR. (b) The top-10 words that are incorrectly decoded by ASR.

(5) Electromagnetic Coupling:

Table 6.1 summarizes the manual speech recognition performance for the electromagnetic coupling test mentioned in Section 6.2.2. In this microbenchmark we remove the equipment (microphone, speaker etc.) from the test environment that can potentially create electromagnetic coupling with the vibration motor. The signal recorded in this microbenchmark does not show any quantitative difference from that of our standard test environment. However, we run a manual speech recognition test on these recordings to identify possible perceptual differences in manual speech recognition. Here the volunteers transcribe the voice of a male non-native speaker recorded with a vibration motor during the microbenchmark test. In



(a) (b)

Figure 6.25: (a) ASR and (b) MSR accuracy for long (> 6 chars) and short (≤ 6 chars) words, as a function of loudness.

this test the volunteers individually listen to the recordings at sound levels according to their personal preferences. The percentage of the incorrect words in the transcription and the perceived quality score given by each user are shown in the Table 6.1. The perceived sound quality is consistent with our previous results at 60 dBSpl, the natural loudness of the speaker's voice at 3 ft from the recording device.

Table 6.1: Manual speech recognition performance for the coupling sensitivity experiment.

User	A	B	C	D	E	F	G	H
Error(%)	8	0	0	8	0	0	17	25
Score	8	8	6	6	4	3	3	3

6.7 Points of Discussion

We discuss a few limitations of the system presented in this chapter, and a few other kinds of applications using *VibraPhone*.

What Is the Best Possible? We have not been able to comment on the best possible performance possible with *VibraPhone*. Such an analysis will certainly need a deeper signal processing treatment, as well as detailed domain knowledge from speech recognition. This work is more of a lower bound on feasibility, drawing on a diverse set of established techniques from literature, and modifying them to suit the needs of this specific problem. We have initiated collaboration with signal processing researchers to push the envelope of this side-channel leak.

Energy: We have side-stepped energy considerations in this chapter. However, we intuitively believe that *VibraPhone* is not likely to be energy hungry (even though the vibramotor consumes considerable energy while pulsating). This is because *VibraPhone* picks up the ambient sounds while it is in the inactive/passive mode, i.e., when it is not serving as an actuator. We plan to characterize the sensitivity and energy profile in future.

Applications: We observed that when vibra-motors are pasted to walls and floors, and music is being played in the adjacent rooms, *VibraPhone* is able to detect these sounds better than the microphone. We also observed that by placing the vibra-motor on the throat, various speech components can be detected, and in some cases, compliments the response of the microphone. Finally, we find that noise properties of vibra-motors and microphones are uncorrelated, enabling the possibility of diversity combining (i.e., they could together behave like a MIMO system, improving the capacity of acoustic channels). All these observations are preliminary, and hence, not reported in this chapter – we plan to investigate them further as a continuation of *VibraPhone*.

6.8 Related Work

Past work on acoustic side-channels and speech recovery are most relevant to this chapter. Given both are reasonably mature areas, we sample a subset of them.

(1) Passive Speech Recording: Gyrophone and AccelWord [152, 153] are perhaps the closest to our work. In Gyrophone [152], authors identify the MEMS sensors' capability to capture sound. The chapter presents a range of signal processing and machine learning techniques to recover traces of ambient sounds from the gyroscope data [175]. AccelWord [153] takes a step forward and uses speech information from the accelerometer [176] to implemented a low-energy voice control application for a limited vocabulary of commands. However, these techniques recover only a low-bandwidth of the spectrum ($<$ 200 Hz), which does not even cover the full range of fundamental frequencies in female speech ($165 - 255$ Hz). Therefore, these techniques mainly focus on extracting the reliable features of sound for consistent pattern classification. In contrast, *VibraPhone* concentrates on recovering a telephone-quality speech (bandwidth 4 kHz [177, 178]) from the vibration motor signal, making the output amenable to manual or automatic decoding. Both Gyrophone and AccelWord are unable to produce (actually not designed for) machine understandable speech.

A family of techniques [179, 180, 181, 182] targets a light/LASER beam on an object exposed to the speech signal and records its vibration by measuring the fluctuation of the reflected beam. Visual microphone [183] is also a similar technique that uses high-speed video of the target object to recover the vibration proportional to the speech signal. Camera-based techniques are devoid of the noisy data that pollute motion sensors/actuators, while they must tackle other difficult challenges in computer vision. A number of solutions have monitored the change in received signal strength (RSS) and phase of the wireless radio signal reflected off the loudspeaker to capture the traces of sound. The projects [184] and [185] demonstrate successful sound recovery using reflected radio signal even when the receiver is not in the line-of-sight of the vibrating object.

(2) Speech Recovery: We borrowed building blocks from the vast literature of speech processing. A body of research [186, 187, 188] explores artificial bandwidth expansion problems primarily to aid high-quality voice transfer over band-limited telephonic channel. Some solutions attempt to identify the phonemes from the low-bandwidth signal and then replace them with high-bandwidth phonemes from a library. These solutions do not solve *VibraPhone*'s problems as majority of them consider 4 kHz signal as the input providing enough diversity for correct phoneme identification. *VibraPhone* attempts to extend the effective bandwidth from 2 kHz to 4 kHz – a challenge because the features up to 2 kHz provide limited exposure to phonemes.

Data imputation techniques [134, 189] attempt to predict erasures in audio signals. When these signals exhibit a consistent statistical model, the erasures can be predicted well, enabling successful imputation. However, vibra-signals often lack such properties, and moreover, the location of erasures cannot be confidently demarcated.

6.9 Chapter Summary

This chapter demonstrates that the vibration motor, present in almost all mobile devices today, can be used as a listening sensor, similar to a microphone. While this is not fundamentally surprising (since vibrating objects should respond to ambient air vibrations), the ease and extent to which the actuator has "picked up" sounds has been somewhat unexpected for us. Importantly, the decoded sounds are not merely vibration patterns that correlates to some spoken words. Rather, they actually contain the phonemes and structure of human voice, thereby requiring no machine learning or pattern recognition to extract them. We show that with basic signal processing techniques, combined with the structure of human speech, the vibra-motor's output can be quite intelligible to most human listeners. Even automatic speech recognizers were able to decode the majority of the detected words and phrases, especially at higher loudness. The application space of such systems remains open, and could range from malware eavesdropping into human phone conversation, to voice-controlled wearables, to better microphones that use the vibra-motor as a second MIMO-antenna. Our ongoing work is in pursuit of a few such applications.

Chapter 7

Conclusion

This dissertation presents methods for sensing, communication, and jamming with inaudible acoustics. We develop signal processing techniques that enable common microphones to sense ultrasonic signals and physical vibrations. These technical primitives serve as the building blocks of new kinds of applications where the signals remain inaudible to humans but become recordable to all microphone-enabled devices. Contributions of this dissertation can be summarized as follows:

(1) Out-of-Band Sensing through Nonlinearity: We leverage channel nonlinearity to develop a signaling-sensing primitive where high-frequency transmitted signals are designed to produce desirable low-frequency components after passing through the channel. Given that microphones exhibit nonlinearity, Chapters 2 and 3 apply this concept to make inaudible ultrasound signals to be recordable my microphones. This leads to new applications in IoT, including an in-air acoustic communication system for beacons. This also uncovers a loophole in voice assistants making them vulnerable to the inaudible voice command attacks. We develop a defense mechanism against such attacks by detecting indelible traces of nonlinearity in the signal through signal forensics.

(2) Touch-based Communication through Modulated Vibration: We explore physical vibration as an alternative modality to communicate with acoustic devices without generating audible sounds. This vibratory communication primitive leads to applications like the secure data transfer through physical contacts. In Chapters 4 and 5, we describe the design of the entire communication stack, including physical layer vibration modulation and sensing techniques, link layer algorithms, and a number of different applications. The prototype of the vibratory communication system is capable of sending data from wearables, like smartwatches, through the human body and enables seamless communication to IoT devices upon physical touch.

(3) Structured Information Reconstruction from Noisy Acoustic Signals: In our work, we design algorithms targeting reconstruction of information which are known to have specific structures. Specifically, in Chapter 5 we develop adaptive denoising method to recover communication data exploiting known structures of the OFDM symbols. Algorithms in Chapter 6 show how harmonic structures of the human voice signal can be exploited to recover speech from weak ambient vibration signals. On the other hand, in Chapter 3, we use the voice signal structure to infer traces of nonlinearity in recorded voice signal leading to a defense technique against inaudible voice command attacks.

Impacts and Future Directions:

In this dissertation, we aim to develop new capabilities in sensing and communication that not only open up new application but also inspire future research. We continuously work toward this vision through collaborations with research groups and reaching out to the community. Various parts of this research have appeared in leading conferences on networking systems (NSDI'15, NSDI'16, NSDI'18) and mobile computing (MobiSys'16, MobiSys'17). The nonlinearity-based sensing technique has received the MobiSys'17 best paper award and it has been selected for the SIGMOBILE'17 research highlights. Principles developed in this dissertation have led to several patented techniques [190, 191, 192] and spurred a diverse set of research both in academia and industry.

We leverage hardware nonlinearity for out-of-band ultrasound recording. However, the acoustic nonlinearity can be generalized to a broader area of research. Nonlinear behavior of air can change the audibility of sound signals leading to techniques in augmented reality. Different liquids and solids exhibit different degrees of nonlinearity for acoustic signals. This can serve as new methods for material identification. On the other hand, vibration of the facial muscles and tissue contains partial information about the spoken words. Therefore, the primitives for structured information recovery from vibration can help speech recognition in a noisy environment.

In closing, we envision a stronger convergence of acoustics in sensing, communication, and computing in tomorrow's cyber-physical systems. Multi-modal sensing-signaling methods are forming the foundation for such systems, ultimately influencing the way in which technology continues to permeate into human society. We believe the inaudible acoustic primitives developed in this dissertation will trigger new opportunities for this foreseeable future.

References

[1] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure peer-to-peer acoustic NFC," in *Proceedings of the ACM SIGCOMM*, 2013.

[2] F. J. Pompei, "Sound from ultrasound: The parametric array as an audible sound source," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[3] N. Roy, H. Hassanieh, and R. R. Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th ACM Annual International Conference on Mobile Systems, Applications, and Services*, 2017.

[4] H. E. Heffner and R. S. Heffner, "Hearing ranges of laboratory animals," *Journal of the American Association for Laboratory Animal Science*, vol. 46, no. 1, pp. 20–22, 2007.

[5] S. Kumar and H. Furuhashi, "Long-range measurement system using ultrasonic range sensor with high-power transmitter array in air," *Ultrasonics*, vol. 74, pp. 186–195, 2017.

[6] "CMU Sphinx," http://cmusphinx.sourceforge.net, [Accessed: Nov. 16, 2018].

[7] W. Hamby, "Ultimate sound pressure level decibel table," http://www.webcitation.org/5rXlLRYsP, 2004, [Accessed: Nov. 16, 2018].

[8] I. M. Jacobs and J. Wozencraft, *Principles of Communication Engineering.* John Wiley & Sons, 1965.

[9] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Springer Science & Business Media, 2012.

[10] F. Xiong, *Digital Modulation Techniques.* Artech House, 2006.

[11] R. G. Lyons, *Understanding Digital Signal Processing.* Pearson Education India, 2004.

[12] S. A. Tretter, *Communication System Design Using DSP Algorithms: With Laboratory Experiments for the TMS320C6713TM DSK.* Springer Science & Business Media, 2008.

[13] D. Mercy, "A review of automatic gain control theory," *Radio and Electronic Engineer*, vol. 51, no. 11.12, pp. 579–590, 1981.

[14] J. P. A. Pérez, S. C. Pueyo, and B. C. López, “AGC fundamentals,” in *Automatic Gain Control.* Springer, 2011, pp. 13–28.

[15] D. Whitlow, “Design and operation of automatic gain control loops for receivers in modern communications systems,” *Microwave Journal*, vol. 46, no. 5, pp. 254–269, 2003.

[16] T. Nakamura, “Piezoelectric speaker,” June 3 1986, US Patent 4,593,160.

[17] “Zedboard,” http://zedboard.org, [Accessed: Nov. 16, 2018].

[18] “Chirp technology,” http://www.chirp.io, [Accessed: Nov. 16, 2018].

[19] “Zoosh technology,” http://www.bdti.com/insidedsp/2011/07/28/naratte, [Accessed: Nov. 16, 2018].

[20] P. A. Iannucci, R. Netravali, A. K. Goyal, and H. Balakrishnan, “Room-area networks,” in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, 2015.

[21] “Top 10000 words from Google’s trillion word corpus,” https://github.com/first20hours/google-10000-english, [Accessed: Nov. 16, 2018].

[22] P. Nation and R. Waring, “Vocabulary size, text coverage and word lists,” *Vocabulary: Description, Acquisition and Pedagogy*, vol. 14, pp. 6–19, 1997.

[23] D. Huggins-daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, “PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.

[24] “High-power Bluetooth speaker: 38 watt,” http://www.fugoo.com/fugoo-tough-xl/, [Accessed: Nov. 16, 2018].

[25] “High-power Bluetooth speaker: 12 watt,” https://www.cnet.com/products/jbl-pulse/specs/, [Accessed: Nov. 16, 2018].

[26] P. J. Westervelt, “Scattering of sound by sound,” *The Journal of the Acoustical Society of America*, vol. 29, no. 2, pp. 199–203, 1957.

[27] P. J. Westervelt, “The theory of steady forces caused by sound waves,” *The Journal of the Acoustical Society of America*, vol. 23, no. 3, pp. 312–315, 1951.

[28] C. Fox and O. Akervold, “Parametric acoustic arrays,” *The Journal of the Acoustical Society of America*, vol. 53, no. 1, pp. 382–382, 1973.

[29] L. Bjørnø, “Parametric acoustic arrays,” in *Aspects of Signal Processing.* Springer, 1977, pp. 33–59.

[30] J. Yang, K.-S. Tan, W.-S. Gan, M.-H. Er, and Y.-H. Yan, “Beamwidth control in parametric acoustic array,” *Japanese Journal of Applied Physics*, vol. 44, no. 9R, p. 6817, 2005.

[31] "Holosonics webpage," https://holosonics.com, [Accessed: Nov. 16, 2018].

[32] "Soundlazer webpage," http://www.soundlazer.com, [Accessed: Nov. 16, 2018].

[33] "Soundlazer kickstarter," https://www.kickstarter.com/projects/richardhaberkern/soundlazer, [Accessed: Nov. 16, 2018].

[34] M. Yoneyama, J.-i. Fujimoto, Y. Kawamo, and S. Sasabe, "The audio spotlight: An application of nonlinear interaction of sound waves to a new type of loudspeaker design," *The Journal of the Acoustical Society of America*, vol. 73, no. 5, pp. 1532–1536, 1983.

[35] "Woody Norris TED talk," https://www.ted.com/speakers/woody_norris, [Accessed: Nov. 16, 2018].

[36] E. Norris, "Parametric transducer and related methods," May 6 2014, US Patent 8,718,297.

[37] M. L. Lenhardt, R. Skellett, P. Wang, and A. M. Clarke, "Human ultrasonic speech perception," *Science*, vol. 253, no. 5015, pp. 82–85, 1991.

[38] B. H. Deatherage, L. A. Jeffress, and H. C. Blodgett, "A note on the audibility of intense ultrasonic sound," *The Journal of the Acoustical Society of America*, vol. 26, no. 4, pp. 582–582, 1954.

[39] R. A. Dobie and M. L. Wiederhold, "Ultrasonic hearing," *Science*, vol. 255, no. 5051, pp. 1584–1585, 1992.

[40] Y. Okamoto, S. Nakagawa, K. Fujimoto, and M. Tonoike, "Intelligibility of bone-conducted ultrasonic speech," *Hearing Research*, vol. 208, no. 1, pp. 107–113, 2005.

[41] S. Nakagawa, Y. Okamoto, and Y.-i. Fujisaka, "Development of a bone-conducted ultrasonic hearing aid for the profoundly sensorineural deaf," *Transactions of Japanese Society for Medical and Biological Engineering*, vol. 44, no. 1, pp. 184–189, 2006.

[42] S. Kim, J. Hwang, T. Kang, S. Kang, and S. Sohn, "Generation of audible sound with ultrasonic signals through the human body," in *16th IEEE International Symposium on Consumer Electronics (ISCE)*, 2012.

[43] M. H. Sherif, *Protocols for Secure Electronic Commerce.* CRC press, 2016.

[44] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *IEEE Symposium on Security and Privacy (SP)*, 2013.

[45] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: Real-time hidden acoustic signal capture with smartphones," in *Proceedings of the 22nd ACM Annual International Conference on Mobile Computing and Networking*, 2016.

[46] L. Zhang, C. Bo, J. Hou, X.-Y. Li, Y. Wang, K. Liu, and Y. Liu, “Kaleido: You can watch it but cannot record it,” in *Proceedings of the 21st ACM Annual International Conference on Mobile Computing and Networking*, 2015.

[47] R. Goubran and R. Botros, “Adaptive sound masking system and method,” June 5 2003, US Patent 20,030,103,632.

[48] A. M. McCalmont, “Voice privacy system with amplitude masking,” Mar. 25 1980, US Patent 4,195,202.

[49] “Sound masking device,” http://www.oeler.com/sound-masking-systems/, [Accessed: Nov. 16, 2018].

[50] “Sound masking solutions,” https://www.speechprivacysystems.com, [Accessed: Nov. 16, 2018].

[51] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, “Inaudible voice commands: The long-range attack and defense,” in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018.

[52] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, “DolphinAtack: Inaudible voice commands,” *arXiv preprint arXiv:1708.09537*, 2017.

[53] L. Song and P. Mittal, “Inaudible voice commands,” *arXiv:1708.07238*, 2017.

[54] “Inaudible voice commands demo,” https://www.youtube.com/watch?v=wF-DuVkQNQQ&feature=youtu.be, [Accessed: Nov. 16, 2018].

[55] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, “Hidden voice commands,” in *USENIX Security Symposium*, 2016.

[56] K.-L. Lee and R. Mayer, “Low-distortion switched-capacitor filter design techniques,” *IEEE Journal of Solid-State Circuits*, vol. 20, no. 6, pp. 1103–1113, 1985.

[57] G. G. G. González and I. M. S. V. Nässi, “Measurements for modelling of wideband nonlinear power amplifiers for wireless communications,” M.S. thesis, Department of Electrical and Communications Engineering, Helsinki University of Technology, 2004.

[58] A. Dobrucki, “Nonlinear distortions in electroacoustic devices,” *Archives of Acoustics*, vol. 36, no. 2, pp. 437–460, 2011.

[59] D. Self, *Audio Power Amplifier Design Handbook.* Taylor & Francis, 2006.

[60] M. J. Hawksford, “Distortion correction in audio power amplifiers,” *Journal of the Audio Engineering Society*, vol. 29, no. 1/2, pp. 27–30, 1981.

[61] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, “DolphinAttack: Inaudible voice commands,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2017.

[62] H. Ye and S. Young, "High quality voice morphing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[63] "Lyrebird," https://lyrebird.ai, [Accessed: Nov. 16, 2018].

[64] I. R. Titze and D. W. Martin, "Principles of voice production," *The Journal of the Acoustical Society of America*, vol. 104, no. 3, pp. 1148–1148, 1998.

[65] "Keysight waveform generator." http://literature.cdn.keysight.com/litweb/pdf/5991-0692EN.pdf, [Accessed: Nov. 16, 2018].

[66] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: Exploiting the gap between human and machine speech recognition," in *9th USENIX Workshop on Offensive Technologies (WOOT)*, 2015.

[67] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, 2014.

[68] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012.

[69] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones," in *Proceedings of the 13th ACM Annual International Conference on Mobile Systems, Applications, and Services*, 2015.

[70] Z. Sun, A. Purohit, R. Bose, and P. Zhang, "Spartacus: Spatially-aware interaction for mobile devices through energy-efficient audio sensing," in *Proceeding of the 11th ACM Annual International Conference on Mobile Systems, Applications, and Services*, 2013.

[71] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel, "Doplink: Using the doppler effect for multi-device interaction," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013.

[72] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: Using the doppler effect to sense gestures," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2012.

[73] Q. Lin, L. Yang, and Y. Liu, "TagScreen: Synchronizing social televisions through hidden sound markers," in *IEEE Conference on Computer Communications INFOCOM*, 2017.

[74] P. Jayaram, H. Ranganatha, and H. Anupama, "Information hiding using audio steganography–a survey," *The International Journal of Multimedia & Its Applications (IJMA) Vol*, vol. 3, pp. 86–96, 2011.

[75] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in *International Workshop on Information Hiding*, 1996.

[76] I. Constandache, S. Agarwal, I. Tashev, and R. R. Choudhury, "Daredevil: Indoor location using sound," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 2, pp. 9–19, 2014.

[77] R. Nandakumar, A. Takakuwa, T. Kohno, and S. Gollakota, "CovertBand: Activity information leakage using music," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2017.

[78] D. Jakobsson and M. Larsson, "Modelling and compensation of nonlinear loudspeaker," M.S. thesis, Chalmers University of Technology, 2010.

[79] W. J. Klippel, "Active reduction of nonlinear loudspeaker distortion," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings.* Institute of Noise Control Engineering, 1999.

[80] F. X. Gao and W. M. Snelgrove, "Adaptive linearization of a loudspeaker," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991.

[81] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: Communicating through physical vibration," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2015.

[82] H. D. Funke, "Acoustic body bus medical device communication system," May 19 1992, US Patent 5,113,859.

[83] A. Dhananjay, A. Sharma, M. Paik, J. Chen, T. K. Kuppusamy, J. Li, and L. Subramanian, "Hermes: Data transmission over unknown voice channels," in *Proceedings of the 16th ACM Annual International Conference on Mobile Computing and Networking*, 2010.

[84] D. Brady and J. C. Preisig, "Underwater acoustic communications," *Wireless Communications: Signal Processing Perspectives*, vol. 8, pp. 330–379, 1998.

[85] I. Hwang, J. Cho, and S. Oh, "Privacy-aware communication for smartphones using vibration," in *IEEE 18th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2012.

[86] T. Yonezawa, H. Nakahara, and H. Tokuda, "Vib-connect: A device collaboration interface using vibration," in *IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2011.

[87] C. Mulliner, "Vulnerability analysis and attacks on NFC-enabled mobile phones," in *IEEE International Conference on Availability, Reliability and Security (ARES)*, 2009.

[88] E. Haselsteiner and K. Breitfuß, "Security in near field communication (NFC)," in *Workshop on RFID Security*, 2006.

[89] T. W. Brown, T. Diakos, and J. A. Briffa, "Evaluating the eavesdropping range of varying magnetic field strengths in NFC standards," in *IEEE 7th European Conference on Antennas and Propagation (EuCAP)*, 2013.

[90] T. P. Diakos, J. A. Briffa, T. W. Brown, and S. Wesemeyer, "Eavesdropping near-field contactless payments: A quantitative analysis," *The Journal of Engineering*, vol. 1, no. 1, 2013.

[91] H. Eun, H. Lee, and H. Oh, "Conditional privacy preserving security protocol for NFC applications," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 153–160, 2013.

[92] M. Giri and D. Singh, "Theoretical analysis of user authentication systems," *International Journal of Innovative Research and Development*, vol. 2, no. 12, 2013.

[93] V. Kaajakari, *Practical MEMS: Design of Microsystems, Accelerometers, Gyroscopes, RF MEMS, Optical MEMS, and Microfluidic Systems.* Small Gear Publishing, 2009.

[94] R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control.* Prentice Hall, 2001.

[95] J. R. Brauer, *Magnetic Actuators and Sensors.* John Wiley & Sons, 2006.

[96] "Pulse width modulation," http://homepage.cem.itesm.mx/carbajal/Microcontrollers/ASSIGNMENTS/readings/ARTICLES/barr01_pwm.pdf, [Accessed: Nov. 16, 2018].

[97] "A guide to accelerometer specifications," https://www.edn.com/electronics-news/4379981/A-Guide-to-Accelerometer-Specifications, [Accessed: Nov. 16, 2018].

[98] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2014.

[99] "Adxl345 datasheet," https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf, [Accessed: Nov. 16, 2018].

[100] "Arduino," http://arduino.cc, [Accessed: Nov. 16, 2018].

[101] "Atmel microcontroller," https://en.wikipedia.org/wiki/ATmega328, [Accessed: Nov. 16, 2018].

[102] P. Semiconductors, "The I2C-bus specification," *Philips Semiconductors*, vol. 9397, no. 750, p. 00954, 2000.

[103] "Cyanogenmod," https://en.wikipedia.org/wiki/CyanogenMod, [Accessed: Nov. 16, 2018].

[104] S. Widnall, "Lecture L19-vibration, normal modes, natural frequencies, instability," *Dynamics*, 2009.

[105] J. Hendershot, "Causes and sources of audible noise in electrical motors," in *Proceedings of the 22nd Incremental Motion Control Systems and Devices Symposium*, 1993.

[106] “Audible noise reduction,” https://www.yumpu.com/en/document/view/10372296/audible-noise-reduction-danfoss, [Accessed: Nov. 16, 2018].

[107] “Reducing audible noise in vibration motor,” http://www.precisionmicrodrives.com/tech-blog/2013/08/15/reducing-audible-noise-in-vibration-motors, [Accessed: Nov. 16, 2018].

[108] “High performance audio with Android,” http://createdigitalmusic.com/2012/07/android-high-performance-audio-in-4-1-and-what-it-means-plus-libpd-goodness-today/, [Accessed: Nov. 16, 2018].

[109] “Samsung Galaxy-S4 teardown,” https://www.ifixit.com/Teardown/Samsung+Galaxy+S4+Teardown/13947, [Accessed: Nov. 16, 2018].

[110] “Loudness comparison,” http://www.gcaudio.com/resources/howtos/loudness.html, [Accessed: Nov. 16, 2018].

[111] A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman, “The visual microphone: Passive recovery of sound from video,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 33, no. 4, pp. 79:1–79:10, 2014.

[112] D. Morris, H. Z. Tan, F. Barbagli, T. Chang, and K. Salisbury, “Haptic feedback enhances force skill learning.” in *WHC*, 2007.

[113] D. Feygin, M. Keehner, and F. Tendick, “Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill,” in *10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS)*, 2002.

[114] K. Huang, E.-L. Do, and T. Starner, “PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills,” in *12th IEEE International Symposium on Wearable Computers (ISWC)*, 2008.

[115] M. Niwa, Y. Yanagida, H. Noma, K. Hosaka, and Y. Kume, “Vibrotactile apparent movement by DC motors and voice-coil tactors,” in *Proceedings of the 14th International Conference on Artificial Reality and Telexistence (ICAT)*, 2004.

[116] Y.-J. Cho, T. Yand, and D.-S. Kwon, “A new miniature smart actuator based on piezoelectric material and solenoid for mobile devices,” in *The 5th International Conference on the Advanced Mechatronics (ICAM)*, 2010.

[117] P. Marquardt, A. Verma, H. Carter, and P. Traynor, “(sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 2011.

[118] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, “Tapprints: Your finger taps have fingerprints,” in *Proceedings of the 10th ACM International Conference on Mobile Systems, Applications, and Services*, 2012.

[119] A. Studer, T. Passaro, and L. Bauer, "Don't bump, shake on it: The exploitation of a popular accelerometer-based smart phone exchange and its secure replacement," in *Proceedings of the 27th ACM Annual Computer Security Applications Conference*, 2011.

[120] "BUMP technologies," http://blog.bu.mp, [Accessed: Nov. 16, 2018].

[121] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[122] N. Saxena and J. H. Watt, "Authentication technologies for the blind or visually impaired," in *Proceedings of the USENIX Workshop on Hot Topics in Security (HotSec)*, 2009.

[123] C. Castelluccia and P. Mutaf, "Shake them up!: A movement-based pairing protocol for CPU-constrained devices," in *Proceedings of the 3rd ACM International Conference on Mobile Systems, Applications, and Services*, 2005.

[124] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp*, 2001.

[125] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *IEEE Symposium on Security and Privacy (SP)*, 2008.

[126] J. Lester, B. Hannaford, and G. Borriello, "'Are you with me?'–Using accelerometers to determine if two devices are carried by the same person," in *Pervasive computing.* Springer, 2004, pp. 33–50.

[127] T. Halevi and N. Saxena, "On pairing constrained wireless devices based on secrecy of auxiliary channels: The case of acoustic eavesdropping," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010.

[128] N. Roy and R. R. Choudhury, "Ripple II: Faster communication through physical vibration," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.

[129] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: Communicating through physical vibration," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2015.

[130] W. S. Burdic, *Underwater Acoustic System Analysis.* Prentice Hall, 1991.

[131] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[132] R. F. Coates, *Underwater Acoustic Systems.* Halsted Press, 1989.

[133] A. D. Waite and A. Waite, *Sonar for Practising Engineers.* Wiley London, 2002, vol. 3.

[134] P. Smaragdis, B. Raj, and M. Shashanka, “Missing data imputation for spectral audio signals,” in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2009.

[135] “Ripple webpage,” http://synrg.csl.illinois.edu/ripple/, [Accessed: Nov. 16, 2018].

[136] J. Adkins, G. Flaspohler, and P. Dutta, “Ving: Bootstrapping the desktop area network with a vibratory ping,” in *Proceedings of the 2nd ACM International Workshop on Hot Topics in Wireless*, 2015.

[137] M. Krawczyk and T. Gerkmann, “STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1931–1940, 2014.

[138] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[139] J. K. Roberge, *Operational Amplifiers: Theory and Practice.* John Wiley & Sons, 1975.

[140] M. Debbah, “Short introduction to OFDM,” *White Paper, Mobile Communications Group, Institut Eurecom*, 2004.

[141] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

[142] A. Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in *Audio Engineering Society Convention 108*, 2000.

[143] M. Engels and F. Petré, *Broadband Fixed Wireless Access: A System Perspective.* Springer Science & Business Media, 2006.

[144] M. Rumney et al., *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges.* John Wiley & Sons, 2013.

[145] I. Galili, D. Kaplan, and Y. Lehavi, “Teaching Faraday’s law of electromagnetic induction in an introductory physics course,” *American Journal of Physics*, vol. 74, no. 4, pp. 337–343, 2006.

[146] P. Tanner, J. Loebach, J. Cook, and H. Hallen, “A pulsed jumping ring apparatus for demonstration of Lenz’s law,” *American Journal of Physics*, vol. 69, no. 8, pp. 911–916, 2001.

[147] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 409–420, 2007.

[148] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. R. Miller, "Maranello: Practical partial packet recovery for 802.11." in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[149] "Chirp," http://www.chirp.io, [Accessed: Nov. 16, 2018].

[150] "Tagtile report," https://techcrunch.com/2012/04/13/facebook-ups-the-mobile-ante-again-buys-mobile-loyalty-rewards-startup-tagtile/, [Accessed: Nov. 16, 2018].

[151] N. Roy and R. Roy Choudhury, "Listening through a vibration motor," in *Proceedings of the 14th ACM Annual International Conference on Mobile Systems, Applications, and Services*, 2016.

[152] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *Proceedings of the 23rd USENIX Security Symposium (SEC)*, 2014.

[153] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra, "AccelWord: Energy efficient hotword detection through accelerometer," in *Proceedings of the 13th ACM Annual International Conference on Mobile Systems, Applications, and Services*, 2015.

[154] D. Keele Jr, "Low-frequency loudspeaker assessment by nearfield sound-pressure measurement," *Journal of the Audio Engineering Society*, vol. 22, no. 3, pp. 154–162, 1974.

[155] "Sound pressure level," https://en.wikipedia.org/wiki/Sound_pressure, [Accessed: Nov. 16, 2018].

[156] B. Taylor, *Guide for the Use of the International System of Units (SI): The Metric System.* DIANE Publishing, 1995.

[157] "Vibraphone project webpage," http://synrg.csl.illinois.edu/vibraphone/, [Accessed: Nov. 16, 2018].

[158] G. Fant, *Acoustic Theory of Speech Production: With Calculations Based on X-Ray Studies of Russian Articulations.* Walter de Gruyter, 1971, vol. 2.

[159] I. R. Titze, *Principles of Voice Production.* National Center for Voice and Speech, 2000.

[160] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989.

[161] O. Lapteva, *Speaker Perception and Recognition: An Integrative Framework for Computational Speech Processing.* Kassel University Press GmbH, 2011.

[162] K. Ogata, *System Dynamics.* Prentice Hall New Jersey, 1998, vol. 3.

[163] J. Hillenbrand and G. M. Sessler, “High-sensitivity piezoelectric microphones based on stacked cellular polymer films (L),” *The Journal of the Acoustical Society of America*, vol. 116, no. 6, pp. 3267–3270, 2004.

[164] J. Eargle, *The Microphone Book: From Mono to Stereo to Surround - A Guide to Microphone Design and Application.* CRC Press, 2012.

[165] P. Fausti and A. Farina, “Acoustic measurements in opera houses: comparison between different techniques and equipment,” *Journal of Sound and Vibration*, vol. 232, no. 1, pp. 213–229, 2000.

[166] B. Hayes, *Introductory Phonology.* John Wiley & Sons, 2011, vol. 32.

[167] Y. Qi and R. E. Hillman, “Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals,” *The Journal of the Acoustical Society of America*, vol. 102, no. 1, pp. 537–543, 1997.

[168] D. R. Feinberg, B. C. Jones, A. C. Little, D. M. Burt, and D. I. Perrett, “Manipulations of fundamental and formant frequencies influence the attractiveness of human male voices,” *Animal Behaviour*, vol. 69, no. 3, pp. 561–568, 2005.

[169] M. J. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[170] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[171] “Sound pressure level chart,” http://www.sengpielaudio.com/TableOfSoundPressureLevels.htm, [Accessed: Nov. 16, 2018].

[172] I. P. Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

[173] A. Brown, “International phonetic alphabet,” in *The Encyclopedia of Applied Linguistics.* Wiley Online Library, 2013.

[174] P. Ladefoged, “The revised international phonetic alphabet,” *Language*, pp. 550–552, 1990.

[175] J. B. Scarborough, *The Gyroscope: Theory and Application.* Interscience Publishers, 1958.

[176] C. D. Johnson, “Accelerometer principles,” in *Process Control Instrumentation Technology.* Pearson Education Limited, 2009.

[177] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, no. 8, pp. 1707–1719, 2003.

[178] P. J. Moreno and R. M. Stern, "Sources of degradation of speech recognition in the telephone network," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1994.

[179] R. P. Muscatell, "Laser microphone," Oct. 23 1984, US Patent 4,479,265.

[180] G. Smeets, "Laser interference microphone for ultrasonics and nonlinear acoustics," *The Journal of the Acoustical Society of America*, vol. 61, no. 3, pp. 872–875, 1977.

[181] C.-C. Wang, S. Trivedi, F. Jin, V. Swaminathan, P. Rodriguez, and N. S. Prasad, "High sensitivity pulsed laser vibrometer and its application as a laser microphone," *Applied Physics Letters*, vol. 94, no. 5, p. 051112, 2009.

[182] J. R. Speciale, "Pulsed laser microphone," Oct. 9 2001, US Patent 6,301,034.

[183] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: Passive recovery of sound from video," *ACM Trans. Graph*, vol. 33, no. 4, p. 79, 2014.

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

[185] W. McGrath, "Technique and device for through-the-wall audio surveillance," Mar. 30 2005, US Patent App. 11/095,122.

[186] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[187] L. Laaksonen, J. Kontio, and P. Alku, "Artificial bandwidth expansion method to improve intelligibility and quality of AMR-coded narrowband speech." in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[188] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 873–881, 2007.

[189] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2067–2080, 2011.

[190] N. Roy, R. R. Choudhury, and M. K. Gowda, "Communicating through physical vibration," Mar. 28 2017, US Patent 9,608,848.

[191] R. R. Choudhury, H. Wang, N. Roy, S. Sen, M. Youssef, A. Elgohary, and M. Farid, "Unsupervised indoor localization and heading directions estimation," Aug. 8 2017, US Patent 9,730,029.

[192] R. R. Choudhury and N. Roy, "Vibrational devices as sound sensors," 2018, US Patent Application 20180336274. [Online]. Available: http://www.freepatentsonline.com/y2018/0336274.html