5-1-2018

Secure short-range communications

Timothy J. Pierson
Dartmouth College

Secure short-range communications

Timothy J. Pierson

Technical Report TR2018-845
Dartmouth Computer Science

Abstract

Analysts predict *billions* of everyday objects will soon become "smart" after designers add wireless communication capabilities. Collectively known as the Internet of Things (IoT), these newly communication-enabled devices are envisioned to collect and share data among themselves, with new devices entering and exiting a particular environment frequently. People and the devices they wear or carry may soon encounter dozens, possibly hundreds, of devices each day. Many of these devices will be encountered for the first time. Additionally, some of the information the devices share may have privacy or security implications. Furthermore, many of these devices will have limited or non-existent user interfaces, making manual configuration cumbersome. This situation suggests that devices that have never met, nor shared a secret, but that are in the same physical area, must have a way to securely communicate that requires minimal manual intervention. In this dissertation we present novel approaches to solve these short-range communication issues. Our techniques are simple to use, secure, and consistent with user intent.

We first present a technique called *Wanda* that uses radio strength as a communication channel to securely impart information onto nearby devices. We focus on using Wanda to introduce new devices into an environment, but Wanda could be used to impart any type of information onto wireless devices, regardless of device

type or manufacturer.

Next we describe *SNAP*, a method for a single-antenna wireless device to determine when it is in close physical proximity to another wireless device. Because radio waves are invisible, a user may believe transmissions are coming from a nearby device when in fact the transmissions are coming from a distant adversary attempting to trick the user into accepting a malicious payload. Our approach significantly raises the bar for an adversary attempting such a trick.

Finally, we present a solution called *JamFi* that exploits MIMO antennas and the Inverse-Square Law to securely transfer data between nearby devices while denying more distant adversaries the ability to recover the data. We find JamFi is able to facilitate reliable and secure communication between two devices in close physical proximity, even though they have never met nor shared a key.

Acknowledgements

First and foremost, this work would not have happened without the continued support and guidance of my advisor, Professor David Kotz. I could not have asked for a better advisor and I greatly appreciate all the time, careful attention, and encouragement he selflessly provided over the past several years. I also want to thank my committee, Professors Sean Smith, Xia Zhou, and Marco Gruteser for their patience, invaluable guidance, and encouraging words.

In addition to excellent faculty, I was fortunate to work with an extremely talented and supportive group of students and postdocs during this dissertation. In particular Travis Peters put up with constant interruptions to his work and listened patiently while I expounded crazy ideas. Travis was extremely good at providing honest feedback in a supportive and helpful manner – thank you. I am also grateful to Ron Peterson for his technical expertise and expansive general knowledge that he expertly used to pressure test many nascent ideas. I would also like to say thank you a number of other people who provided support, encouragement and friendship along the way: Andres Molina-Markham, Xiaohui Liang, Reza Rawassizadeh, Cory Cornelius, Shrirang Mare, Aarathi Prasad, Rima Murthy, Tianlong Yun, Varun Mishra, Taylor Hardin, Shengjie Bi, George Boateng, and Ira Ray Jenkins. In the end my work was stronger for your help.

Finally, I consider myself extremely lucky to have a supportive family. My parents James and Paula Pierson taught me the value of an education. My wife Courtney and daughter Katie put up with all the late nights and never once complained. Thank you all for the unwavering support!

Dedication

I dedicate this dissertation to my sister Chris Pierson and to my daughter Katie Pierson.

Chris would have been the first Pierson to complete a Ph.D., but was taken from us far too early. This one is for you little sister!

Katie, I hope you will never stop learning.

Contents

List of Tables

List of Figures

1

Introduction

The concept of the Internet of Things (IoT) has garnered a great deal of press lately, with suggestions that *billions* of everyday objects will soon become "smart" after product designers add wireless communication capabilities to items that were previously not networked [94]. The dream is that networks of these newly connection-enabled devices will give us greater insight into the behavior of complex systems than previously possible. The reality, however, is that the arrival of billions of devices could quickly turn the IoT dream into a nightmare if these new devices are deployed with insecure configurations or are operated without secure communication between devices. In this dissertation we present novel approaches to

short-range communications that are simple to use, secure, and consistent with user intent. Our techniques can be used to impart information such as configuration parameters onto nearby devices and can also be used to ensure secure communications between physically proximate devices.

Before detailing our contributions, we make the following observations about the impending arrival of billions of IoT devices:

- many devices already have IoT connectivity (and billions more are expected);
- devices may handle sensitive information;
- securing sensitive information will be difficult;
- insecure IoT devices are dangerous; and
- existing security methods are inadequate.

1.1 Many devices already have IoT connectivity

As they are typically envisioned, IoT devices are low-powered devices that have one or more sensors or actuators, have limited computational capabilities and user interfaces (UI's), and have short-range radios such as Wi-Fi, Bluetooth, or Zigbee. A simple use case is that these IoT devices will be physically placed in areas of interest, will monitor aspects of the environment using their sensors, then will use their radio to communicate measurements to one or more distant data repositories for aggregation and analysis. In more complex scenarios, multiple devices may work together to influence an outcome. For example, a blood-glucose monitor may work with an insulin pump to maintain a diabetic patient's sugar levels in a desired range [101].

Devices are already shipping with IoT-type connectivity. The market research firm Berg Insight estimates that 5.9 billion products were sold worldwide in 2014

that contained embedded microprocessors and 4.8 billion of those devices had some form of embedded connectivity [12]. Gartner, another market research firm, builds on this growing connectivity theme and projects the IoT market will grow to 26 billion devices in the next few years [78]. Others have projected as many as 100 billion IoT devices will be in service by 2025 [102]. While these estimates vary widely, they point to same conclusion – there will be a lot more connected devices in the near future than there are today. If these forecasts are even close to accurate, then there will soon be more than four connected devices for every person on Earth.

Berg Insight goes on to predict that devices that already have some computational capabilities but currently lack communications gear will be the first to be augmented with short-range Wi-Fi, Bluetooth or Zibgee radios, but by 2020 the majority of new IoT devices will be devices that have historically had no computational capabilities whatsoever. For example, several commercial companies and researchers are adding computing and connectivity to previously non-computational items such as shirts [54], shoes [25], and jewelry [53]. Connecting these previously non-computational devices to the Internet raises security and privacy concerns if the devices, the networks they connect to, or the data repositories where they store data are misconfigured or contain vulnerabilities.

1.2 Devices may handle sensitive information

As IoT devices become ubiquitous, they will likely collect and share sensitive information that may impact a user's security or privacy.

1.2.1 Security

Sometimes IoT devices are envisioned to work together to not only monitor a situation, but also to influence an outcome. For example, as noted above, a blood-glucose

monitor may work with an insulin pump to maintain a diabetic patient's blood-sugar level in a desired range [101]. In this case the blood-glucose monitor would analyze the patient's glucose levels and instruct the insulin pump to discharge a bolus of insulin when necessary. The data exchanged between devices must be secure for patient safety. An adversary that breaks the communication security between the two devices may be able to command a release of dangerous levels of insulin, possibly harming or killing the patient [32].

1.2.2 Privacy

In addition to causing physical harm, IoT networks present numerous privacy risks [98]. For example, as Rose et al. note, if a user's refrigerator reports the foods eaten and a fitness tracker reports activity data, the combination of these data streams paint a much more detailed and private description of the person's overall health [102]. This data aggregation effect can be particularly powerful with respect to IoT devices because many devices produce additional metadata like time stamps and geolocation information. This metadata adds even more specificity about the user and the user may not want that information revealed. In some cases the user may not even be aware the food eaten and activity level data were aggregated and shared.

Sayles et al. highlight another privacy risk with potentially profound consequences. They found HIV-positive patients fear of stigma after disclosure of their condition often led patients to avoid medical services and medications [105]. In this case, if a medical device were to leak private information about an HIV patient's condition, the patient may avoid needed therapy.

1.3 Securing sensitive information will be difficult

Researchers and the popular press have already pointed out that securing billions of devices of devices will be difficult by calling attention to the coming "Internet of *Risky* Things" [66, 80, 107, 116]. These authors point out instances where, even with today's relatively limited numbers of IoT devices, items ranging from automobiles [48] to baby monitors [42] have been compromised by adversaries exploiting incorrect configurations, insecure communications between devices, or other security vulnerabilities. To avoid these compromises devices must be properly initially configured, must be managed so that they stay secure, and must securely communicate with unfamiliar devices.

1.3.1 Initially configuring devices

In many cases IoT devices will not know ahead of time where they will be placed nor the devices with which they will need to communicate. This suggests the devices will need to be individually configured for their local environment at or near deployment time. For example, if a new device must connect with a local Wi-Fi access point (AP), the device must be configured with the particular AP's network name (e.g., SSID) and password. Configuring devices to make these connections has historically been difficult, especially for users who are not tech-savvy [16, 29, 49, 50, 75, 128]. We provide techniques to impart information such as the access point's SSID and password onto new devices. Our methods are fast, easy, secure, and consistent with user intent.

1.3.2 Managing devices

After devices are configured they must be actively managed to remain operational in changing environments. For example, sometimes network access credentials

change, such as when a Wi-Fi AP's password is changed. In that case, all the devices using that password for connectivity must be updated to the current password or the devices will go offline.

While there are other steps required to keep devices operational, such as applying security patches and software updates, in this thesis we are primarily concerned with keeping devices current with the proper configuration information. Like with initial configuration, a solution that would allow users to quickly and consistently manage device configuration would be useful. Our techniques provide a consistent way to impart updated information onto wireless devices, regardless of device type or manufacturer.

1.3.3 Communicating with unfamiliar devices

If billions of wireless devices are deployed, people and the devices they wear or carry may encounter dozens, possibly hundreds, of other devices each day. Sometimes these devices will need to securely communicate. In this thesis we adopt the approach advocated by Balfanz et al. where we consider *demonstrative identification* and *location-limited channels* [10]. Unlike many existing approaches, we do not require pre-authentication between devices.

Demonstrative identification

Demonstrative identification means that a user takes action to identify the specific device with which they intend to interact. The idea is that user does more than simply select a device from a list of (possibly spoofed) devices displayed on a screen. Instead they take action to positively identify a specific device. For example, Balfanz suggested touching a device to positively identify a particular device from many others in the local environment. In this way the user demonstrates identification of the desired device.

We adopt a similar approach. In our work a user moves devices into close physical proximity as a method of demonstrative identification. We assume the user can reliably identify devices they wish to use. Whether the devices have been compromised in some fashion, however, is out of scope for this work.

Location-limited channels

Location-limited channels refers to channels which only operate over a short distance. These location-limited channels are used to exchange secrets between devices in physical proximity. We concur and use aspects of radio wave propagation to ensure secure short-range communications between devices.

Pre-authentication

Pre-authentication is a concept often suggested for device communication [119]. In this case, it is assumed that devices have already been introduced and that they have a means for secure communications going forward. We do not assume devices have already been introduced nor even that they mutually trust a common third party such as a certificate authority. Our techniques can be used to impart information such as a cryptographic key onto a device, can be used to ensure the key came from a nearby device (not a distant imposter), and can facilitate secure short-range communications with nearby devices.

1.4 Insecure IoT devices are dangerous

IoT devices are normally thought of as having little computational power and are sometimes dismissed as not particularly dangerous to a large network. This thinking is misguided when billions of "small" devices are aggregated into large *botnets*. Like the infamous torture, "death by a thousand cuts", billions of computationally

small IoT devices working together, each making a tiny cut, comprise a formidable threat to the entire Internet.

One recent example of a horde of low-powered devices working together to achieve an outsized result is the *Mirai* botnet that first surfaced in 2016 [121]. Mirai scans the Internet for devices using common default user names and passwords. If it encounters a device using one of more than 60 common credentials, Mirai logs in and installs malware on the device. Once a device is compromised by malware, it connects to a Command and Control server for further instructions such as participating in Distributed Denial of Service (DDoS) attacks as part of a larger botnet [123].

Mirai made headlines on September 20, 2016 when it attacked the website of security researcher Brian Krebs with the largest DDoS attack that had ever been seen, with some reports suggesting the site was offline for more than 24 hours (although Krebs' report claims the attack was not successful) [69]. The Mirai botnet subsequently attacked French web hosing and cloud service provider OVH with an even larger attack, generating more than 1 terabyte of traffic per second, largely generated by web cameras with little computation power [43]. Next, Mirai attacked Domain Name Service provider Dyn (now part of Oracle) [27], making several high-profile websites such as Twitter, Netflix and Github unavailable for several hours [55]. These attacks originated from small IoT-type devices such as web cameras, but in aggregate they had an enormous impact by taking down some of the Internet's best known and most used sites.

We now know that Mirai was originally written to give a group of college students an advantage in the online game *Minecraft*, but its method of exploiting misconfigured devices has had a far larger impact [46]. Mirai is still active today and there are now many variants. Mirai and its brethren succeed primarily by checking for misconfigured devices with default user name and passwords [68]. The most

#	*User name*	*Password*
1	root	admin
2	admin	root
3	DUP root	123456
4	ubnt	12345
5	access	ubnt
6	DUP admin	password
7	test	1234
8	oracle	test
9	postgres	qwerty
10	pi	raspberry

Table 1.1: Ten most commonly exploited IoT device default passwords [120].

commonly exploited default passwords reported by security firm Symantec are shown in Table 1.1 [120]. In that table we see that hackers are aware that every Raspberry Pi computer [97] is initially configured with the user name "pi" and password "raspberry" and they write malware that attempts to compromise these devices by guessing those default credentials. If these credentials are not changed before a device is exposed to the Internet, it will be compromised in a short period of time. One study by Nazario found that devices exposed to the Internet were probed for vulnerabilities 800 times per hour and it took on average six minutes for a device with default credentials to be compromised [82].

In addition to various botnets active on the Internet, a search engine for misconfigured Internet-connected devices called Shodan [114] can automatically search for and identify vulnerable devices. This tool can give attackers a ready list of devices to exploit. If a ready-made list of vulnerable devices is not easy enough, Tech Republic reports that networks of compromised devices can be rented to "systematically attack and attempt to take down a company for less than $100" [99].

Clearly these problems, where misconfigured online devices are compromised and used for nefarious purposes, will get worse if the number of Internet-connected devices grows into multiple billions of devices.

1.5 Existing security methods are inadequate

Many IoT-type devices have limited user interfaces and may not have any form of display or keyboard. To overcome this interface limitation, device manufacturers often rely on smartphone or PC-based setup applications for initial configuration and management [34]. Users must find and download the correct setup application for their particular device and if they do not run the setup application, devices frequently use well-known default credentials such as those shown in Table 1.1.

This setup application requirement may be manageable when the number of devices is relatively small (although even that assumption is questionable given the success of Mirai and its siblings), but will quickly break down as the number of devices to be managed grows. If a person needs to find and download a different setup application for each device they manage (e.g., an application for their smart lights, a different application for their smart shirts, another application for their smart shoes, and so on), then when IoT-type devices are common place, users will be overrun with setup applications!

1.6 Our techniques point to a solution

In this thesis we present three approaches that can help with IoT security and privacy: (1) a technique called *Wanda* that can securely impart information such a configuration details onto a nearby target device, (2) a method called *SNAP* where a single-antenna device can determine when it is in close proximity with a transmitter so the single-antenna device can reject spoofed information sent by an adversary located more than about 9 cm away, and (3) a solution called *JamFi* that builds on Wanda, delivering data faster and without the need for the target device to run additional software.

1.6.1 Wanda

Wanda uses radio strength as a communication channel to securely impart information onto nearby devices, even though the nearby devices have never been seen before, nor have any secrets been pre-shared. We focus on using Wanda to introduce new devices into an environment, but Wanda could be used to impart any type of information onto nearby devices, regardless of device type or manufacturer. Unlike many other approaches, Wanda does not require any specialized hardware (or any hardware changes) in the new devices, does not require any pre-shared secrets, does not require a trusted third party, does not require Internet access, and does not require complex algorithms or complicated cryptography libraries. Furthermore, Wanda does not require the devices to be adjacent, or even movable – making it useful for large appliances as well as small mobile devices.

Contributions

We discuss Wanda in Chapter 2 and make the following five contributions:

1. a consistent, fast, easy, and secure method to impart any kind of information onto commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the device;

2. protocols for imparting information onto new devices (such as a Wi-Fi SSID and password), introducing two devices so they can establish a secure and user-intended connection, and imparting cloud identity and credentials into a new device;

3. a prototype implementation and experimental evaluation;

4. a security analysis of the system; and

5. the results of a 36-person user study.

1.6.2 SNAP: SiNgle Antenna Proximity

Because radio waves are invisible, a user may believe transmissions are coming from a nearby device when in fact the transmissions are coming from a distant adversary attempting to trick the user into accepting the adversary's malicious payload. *SNAP* is a method for a single-antenna wireless device to determine when it is in close physical proximity to another wireless device. Our approach significantly raises the bar for an adversary attempting such a trick.

Contributions

We discuss SNAP in Chapter 4 and make the following contributions:

1. a novel method for a single-antenna device to quickly determine when it is in close proximity with a transmitting device;

2. a reference Wi-Fi implementation that performs the same frame decoding steps *any* Wi-Fi device must perform; and

3. an experimental evaluation of the technique using several popular types of antennas.

1.6.3 JamFi

Finally, we present a method called *JamFi* that exploits MIMO antennas and the Inverse-Square Law to securely transfer data between nearby devices while denying more distant adversaries the ability to recover the data. We find JamFi is able to facilitate reliable and secure communication between two devices in close physical proximity, even though they have never met nor shared a key. JamFi is faster than Wanda and does not require the target device to run additional software.

Contributions

We discuss JamFi in Chapter 5 and make the following contributions:

1. a consistent, fast, easy, and secure method to transfer any kind of information between commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the devices;

2. a theoretical analysis of jamming at close range to facilitate data transfer; and

3. an experimental evaluation using several different commercial off-the-shelf Wi-Fi receivers.

1.6.4 Published work

Chapter 2 is a revised and extended version of our INFOCOM paper [88] and technical report [89]. That chapter is also the basis of our MobiSys demonstration [87]. Chapter 4 is currently in preparation for conference submission. We presented an early version of Chapter 5 at the MobiCom S3 workshop [90]. A version of Chapter 5 is currently in preparation for conference submission. Additionally we have applied for patents based on Chapters 2, 4, and 5.

2

Wanda: securely introducing wireless devices

2.1 Introduction

We note in Chapter 1 that configuring and managing billions of devices will be extremely difficult. As an illustration in the healthcare domain, imagine that a general-practice physician tells a patient that he'd like the patient to take home a wireless blood-pressure monitor and use it every day so that the physician can remotely monitor the patient's health. The intention is that the blood-pressure

measurements taken by the patient will end up stored in the patient's Electronic Health Record (EHR) at the physician's clinic. The clinical team can then see the patient's blood pressure on a daily basis and get automated alarms if any abnormal readings are recorded.

At least three problems arise in making scenarios such as at-home blood-pressure monitoring a reality. The first problem is that blood-pressure monitors, like many IoT sensors, do not normally come with long-range communication connections; they have short-range radios such as Wi-Fi, Bluetooth, or Zigbee. The blood-pressure monitor must somehow get connected with other devices in the home such as a Wi-Fi access point (AP) in order to transmit its medical data to the physician's EHR system. Making those connections is difficult for many people, especially considering that different types of devices from different manufacturers often have different methods of making a connection and that the devices themselves often have very limited user interfaces.

A second problem with this blood-pressure scenario is that once a connection is made between the blood-pressure monitor and a device capable of transmitting data long distances, the blood-pressure readings must get to the right patient record in the right physician's EHR system. This implies that the blood-pressure readings must be augmented with additional EHR credentials (e.g., patient ID, password) and destination information (e.g., a Restful API URL).

A third problem arises when devices partner with other nearby devices so they can work together in a peer-to-peer fashion, such as a blood-glucose monitor working with an insulin pump. In these peer-to-peer cases the devices may maintain a connection with a long-range communication device, but may also need a connection with neighboring devices using encryption based on a unique key for a specific pair of devices, rather than a common key shared by all devices. Establishing the encryption can be difficult if the devices have never met before and have

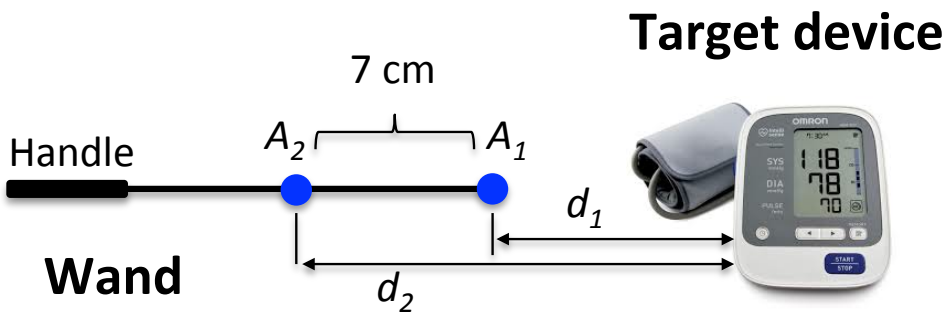

Figure 2.1: Wand with two antennas, A_1 and A_2, separated by 7 cm in our prototype. The distance between antenna A_1 and the target device is d_1. The distance between antenna A_2 and the target device is d_2. The Wand is intended to be pointed directly at the target device, so that $d_2 = d_1 + 7$ cm.

never shared a secret key.

To overcome these three and other difficulties inherent in configuring wireless devices, we present a technique called Wanda. Wanda introduces a small hardware device called the "Wand" that has two antennas separated by one-half wavelength as shown in Figure 2.1 and uses radio strength as a communication channel to simply, securely, and consistent with user intent, impart information onto devices. In this chapter we focus on introducing devices to an environment, but the Wand could be used to impart *any* type of information onto a nearby device. Wanda is more than just a solution for pairing devices or connecting to access points.

Wanda builds on pioneering work done by Cai et al. in Good Neighbor [18] in that the Wand determines when it is in close proximity to another transmitting device by measuring the difference in received signal strength on the Wand's two antennas. Wanda then expands upon Good Neighbor by exploiting wireless signal reciprocity to securely impart information in-band from the Wand onto the nearby target device.

Unlike many other approaches, Wanda does not require any specialized hardware (or any hardware changes) in the new devices, does not require any pre-shared secrets, and does not require complex algorithms or complicated cryptography li-

braries. Furthermore, Wanda does not require the devices to be adjacent, or even movable – useful for large appliances as well as small mobile devices.

Using Wanda could hardly be easier: a person simply points the Wand at a nearby device that requires connectivity and the Wand almost magically imparts connectivity parameters onto the target device. This happens one time and afterward the Wand is not involved in future communications – the Wand itself disappears from the picture.

2.1.1 Assumptions

Throughout this chapter we make the following assumptions about the "target device," which is the device receiving information from the Wand: (1) it has at least one radio antenna that it can use to transmit and receive wireless data, (2) it can measure the signal strength of wireless communication frames, (3) it may be limited computationally, but can run a small piece of software that implements the Wanda protocol, (4) it cannot be relied upon to have additional sensors such as cameras, microphones or accelerometers, and (5) it cannot be altered to add new hardware.

We make the following assumptions about the Wand: (1) it can be trusted to generate a secret key, (2) it has a radio compatible with that of the target devices, and two antennas located approximately one-half wavelength apart, (3) it is easily portable and can be brought next to and pointed at the target device, and (4) it can run the Wanda protocol.

2.1.2 Contributions

Wanda is a novel approach for imparting information onto a target device, even though the target device has never been seen before, nor have any secrets been pre-shared. We make the following five contributions:

1. a consistent, fast, easy, and secure method to impart any kind of information onto commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the device;

2. protocols for imparting information onto new devices (such as a Wi-Fi SSID and password), introducing two devices so they can establish a secure and user-intended connection, and imparting cloud identity and credentials into a new device;

3. a prototype implementation and experimental evaluation;

4. a security analysis of the system; and

5. the results of a 36-person user study.

2.2 Background

Wanda uses radio signal strength to impart information onto devices; in this section we briefly review some basic concepts that are key to Wanda's operation. We start by reviewing the theory behind how a radio signal travels through free space, then examine how obstacles can affect the received signal strength, and finally investigate variation in real-world signal strength by capturing Wi-Fi frames in three different environments. Wanda leverages signal-propagation characteristics described in this section to impart information on target devices and exploits real-world environmental factors to make it virtually impossible for adversaries to eavesdrop on Wanda communications. The material in this section provides the theoretical foundations for why Wanda *should* work, while Section 2.6 shows that Wanda *does* work.

2.2.1 Radio propagation in free space

A radio signal transmitted by an antenna attenuates, or fades, as it travels through the air according to the well known free-space propagation model (sometimes called the Friis transmission equation) [96] given in Equation (2.1):

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2 \tag{2.1}$$

where P_r is the power received in watts, P_t is the power at the surface of the transmitting antenna in watts, G_t and G_r are the gains of the transmitting and receiving antennas, λ is the frequency of the signal, and d is the distance between the transmitting and receiving antennas.

This model assumes the radio waves travel through free space without bouncing off or passing through any obstacles before arriving at a receiving antenna. Although reflections and multipath signals, where the waves bounce off objects, can affect the signal strength measured at a receiver (discussed in more detail below), in general the distance factor d in the denominator of Equation (2.1) tells us that as the distance between the transmitter and receiver increases, the signal strength at the receiver decreases quadratically.

It is sometimes useful to consider signal strength in relation to a known amount of power. In that case, dBm, which expresses power in decibels compared to one milliwatt (mW), is often used. The conversion is given by Equation (2.2) as [96]:

$$\text{dBm} = 10 \log_{10} \left(\frac{P_r}{1 \text{ mW}} \right). \tag{2.2}$$

Using Equation (2.2) we can rewrite Equation (2.1) in dBm for free space as [96]:

$$P_r = P_0 - 10\alpha \log_{10} \left(\frac{d}{d_0} \right) \tag{2.3}$$

where P_r is now the received power in dBm, P_0 is the power in dBm received at a distance of d_0 from the transmitter, d is the distance between the sending and receiving antennas, and α, called the path-loss exponent, represents the reduction in power as the signal travels. In free space α is 2.

In the remainder of this chapter we use P_x to indicate power in dBm predicted by radio signal propagation models, and we use Received Signal Strength Indicator (RSSI) to indicate power measured in dBm by actual hardware.

2.2.2 Obstacles

Equation (2.3) gives a good approximation of signal attenuation in free space, but in the real world obstacles, moving and fixed, can attenuate a signal or cause reflections that create multiple paths between a transmitter and a receiver. The result is that multiple copies of the transmitted signal, each with a different attenuation, delay, and phase shift, arrive at the receiver superimposed upon each other. This superposition can result in either constructive interference where multiple copies of the signal add to each other, or destructive interference where multiple copies of the signal cancel each other. The changes in signal strength caused by obstacles is called *fading*.

There are two types of fading: *slow* and *fast*. *Slow fading* occurs when changes to the signal strength happen slowly over time. Shadowing, where an stationary obstacle such as a building lies between the transmitter and receiver, is an example of slow fading. In this case the alteration to the signal strength is normally constant unless the transmitter or receiver move. *Fast fading* occurs when changes to the signal strength happen quickly such as when a moving obstacle comes near a transmitter and receiver.

We can account for fading by altering Equation (2.3) to add a fading compo-

nent, which gives us the log-normal shadow model [96]:

$$P_r = P_0 - 10\alpha \log\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{2.4}$$

where χ_σ is a Gaussian random variable representing fading. In slow fading environments, χ_σ has zero mean and σ standard deviation. In fast-fading line-of-sight environments χ_σ follows a Rician distribution. In fast-fading, non-line-of-sight environments, it follows a Rayleigh distribution. As noted above, in free space α is 2, but it in real-world dynamic environments α often ranges from 1.2 to about 8 [84].

In a dynamic environment where there are moving objects, the χ_σ in Equation (2.4) can change rapidly, making actual measurements of RSSI highly variable. In a dynamic environment the moving objects are changing their position relative to the transmitter and receiver – which slightly changes the length of the path taken by the portion of the signal reflecting off from those obstacles. The difference in path length, in turn, slightly alters the phase of the received signal. This change in phase can change how the multiple copies of the signal add up to create constructive or destructive interference. Finally, a Doppler shift caused by the moving obstacle slightly changes the frequency of the received signal, and interference has been shown to vary depending on the frequency of the signal [51]. We examine these issues in detail in Chapter 4.

In addition to the environmental variables, the signal strength captured by real hardware is also subject to manufacturing variability as well as thermal noise in the antenna [59]. Wanda exploits the variability from manufacturing and thermal noise, together with variability from obstacles in the environment, to make it difficult for an adversary to eavesdrop on communications between Wanda devices (see Section 2.7 for details).

2.2.3 Real-world observations

To understand the role environment plays in signal propagation, we captured the signal strength of Wi-Fi frames exchanged between a computer and a Wi-Fi AP in three very different (but realistic) locations where Wanda might be used. The first was a quiet home environment where no one was moving, the second was a local coffee shop where a small number of customers were milling about, and the third was a busy computer science lab bustling with student activity. We used an Alfa Networks AWUS036H external Wi-Fi antenna [4] and captured the RSSI returned by the Alfa card in the form of RadioTap [136] headers. These RSSI values were captured using a Python program written with Scapy [13]. In all cases the receiving antenna was stationary while frames were exchanged with the AP.

Figure 2.2 shows the distribution of RSSI measurements returned by capturing 12,000 Wi-Fi frames sent between a Wi-Fi AP and the receiving antenna at each location. Frames were sent every 100 ms (we note that this is 25,000 times longer than the timeframes discussed in Chapter 4). In the home and computer science lab, the distance between the access point and the receiver was approximately 4 meters. In the coffee shop the distance was approximately 8 meters. The differences in distance led to differences in RSSI, and as expected the presence (or absence) of moving obstacles lead to a varying degrees of variability of the RSSI. When frames were captured in the quiet home environment the RSSI readings were tightly grouped and had little variation; we saw increased variability in the coffee shop, and a great deal of variability in the busy lab. Table 2.1 provides details on the mean, standard deviation, and range (number of distinct RSSI values) of the frame RSSIs captured.

Although the variability in RSSI is lower in environments where there is little activity, it is important to note that there is still variability – it is not the case that

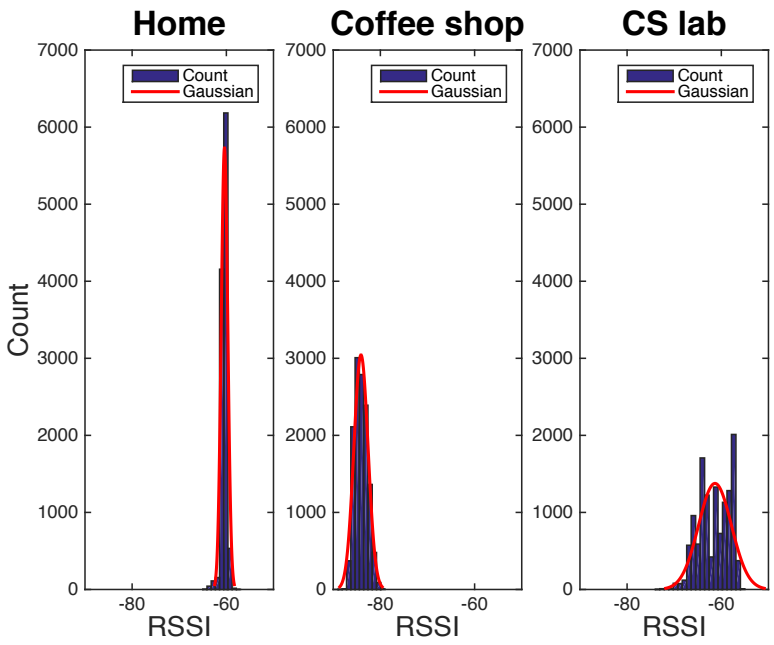

Figure 2.2: Real world RSSI observations. Distribution of 12,000 RSSI readings captured in three different environments. The figures show a histogram of RSSI values measured, and a best-fit Gaussian distribution for the RSSI values. Environments with more moving obstacles had higher variability in RSSI values.

RSSI readings were the same for all frames. We see in the quiet home environment that there were still eight different RSSI values observed. Other researchers have found that even in an underground concrete tunnel where outside signals and the effects of moving obstacles were not present, there was still a variation of at least 2 dBm away from the mean [59].

The equations in this section provide the theoretical basis for Wanda's opera-

Location	*Mean*	*Std Dev*	*Range*
Home	-60	0.69	8
Coffee shop	-84	1.50	10
CS lab	-61	3.48	19

Table 2.1: RSSI mean, standard deviation, and range (number of distinct values) of 12,000 Wi-Fi frames captured at three different locations. The standard deviation and range of RSSI measurements increased as the number of moving obstacles increased, but even the static home environment still exhibited eight different RSSI readings.

tion. We see below that Wanda is able to overcome the unpredictable environmental noise and impart secret information onto a nearby device while making eavesdropping extremely difficult from more than a few centimeters away.

2.3 Approach

Wanda builds on two insights that can be gleaned from the concepts highlighted in Section 2.2. The first insight is that if a device has two antennas, it can determine when it is in close proximity to another device that is transmitting radio signals. The second insight, Wanda's major technical contribution, is that when a device with two antennas determines it is in close proximity to another device, it can use its two antennas to securely impart information onto the other device.

In Wanda, the Wand is a device with two antennas (see Figure 2.1) and it uses those antennas to implement two primitive operations: *detect* and *impart*. This section explains these primitives in detail.

2.3.1 Detect primitive

Wanda uses a two-antenna Wand to determine if it is in close proximity to another device that is transmitting a radio signal. Each antenna in the Wand is capable of independently measuring the power received and providing its own RSSI measurement. Building on Equation (2.4), the power received on each of the two antennas of the Wand will be:

$$\begin{aligned} P_1 &= P_0 - 10\alpha \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma \\ P_2 &= P_0 - 10\alpha \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma \end{aligned} \tag{2.5}$$

where P_0 is the power in dBm measured at a distance of d_0 from the transmitter, P_i is the power in dBm measured at receiving antenna A_i, and d_i is the distance between the transmitter and receiving antenna i.

Armed with the equations in (2.5), we can calculate the difference in signal strength between the two antennas A_1 and A_2 as follows:

$$\begin{aligned} P_1 - P_2 &= P_0 - 10\alpha \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma \\ &\quad - (P_0 - 10\alpha \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma) \\ &= -10\alpha(\log_{10}(\frac{d_1}{d_0}) - \log_{10}(\frac{d_2}{d_0})) \\ &= -10\alpha \log_{10}(\frac{d_1}{d_2}). \end{aligned} \tag{2.6}$$

The antennas on the Wand are physically close together; in our prototype they are 7 cm apart (roughly 1/2 wavelength). Because they are close together and the signal is measured on each antenna at almost exactly the same instant (except for signal travel time to cross the gap between antennas), the environmental factors represented by χ_σ in Equation (2.6) are likely to be similar on each antenna. By taking the difference in signal strength observed on two antennas, sometimes called the RSSI Ratio [20], the environmental factors tend to cancel out. This suggests that some of the randomness of the environment we saw in our real-world observations in Section 2.2 will be minimized in the RSSI Ratio on the Wand.

We note that Equations (2.3) though (2.6) are meant to estimate power in the transmitter's far field, but when the transmitter and receiver are close together, near-field effects can cause changes to the signal. We estimate near-field effects with greater precision in Chapters 4 and 5, but with Wanda we are interested in the *difference* in signal strength between the Wand's two antennas. While estimates of the power received by antennas in the near field diverge from estimates of antennas in the far field, the difference in signal strength as represented by the RSSI Ratio is the same in both fields (see a derivation in Appendix A.1). Additionally we see in Section 2.6 that our experimental results conform well to the equations in this chapter, even when antennas are in the near field.

Next we see that the RSSI Ratio in Equation (2.6) provides an indication of the proximity between the Wand and the target device. When the Wand and the target device are far apart, the distance between antennas A_1 and A_2 is small relative to the distance to the transmitter. In that case the RSSI will be approximately, although not precisely, equal on each receiving antenna. For example, suppose antennas A_1 and A_2 on the Wand are 7 cm apart and are aligned with the transmitting antenna so that A_2 is 7 cm farther away from the transmitting antenna than A_1 (see Figure 2.1). In this case $d_2 = d_1 + 7$ cm. Further suppose the distance between A_1 and the transmitting antenna, d_1 is 30 cm (i.e., more than 4 times the distance between the two antennas). In that case, using Equation (2.6) and assuming $\alpha = 2$ yields a difference, Δ, of:

$$\begin{aligned} d_1 &= 30 \text{ cm} \\ d_2 &= 30 \text{ cm} + 7 \text{ cm} = 37 \text{ cm} \\ \Delta &= -10\alpha \log_{10}(30/37) \approx 1.8 \text{ dBm}. \end{aligned} \tag{2.7}$$

When the Wand is close to the target device, the distance between antennas A_1 and A_2 is large relative to the distance to the transmitter. In that case the difference between received power on the two antennas on the Wand will be large. For example, assume the transmitter in Figure 2.1 is located 1 cm from A_1. In that case the expected RSSI difference is:

$$\begin{aligned} d_1 &= 1 \text{ cm} \\ d_2 &= 1 \text{ cm} + 7 \text{ cm} = 8 \text{ cm} \\ \Delta &= -10\alpha \log(1/8) \approx 18.1 \text{ dBm}. \end{aligned} \tag{2.8}$$

This demonstrates that when the Wand is in close proximity to a transmitting device, the difference in power readings between the Wand's two antennas will be significantly larger than the difference in power readings when the device is far away. In this example there is an expected 10-fold increase in the RSSI Ratio when

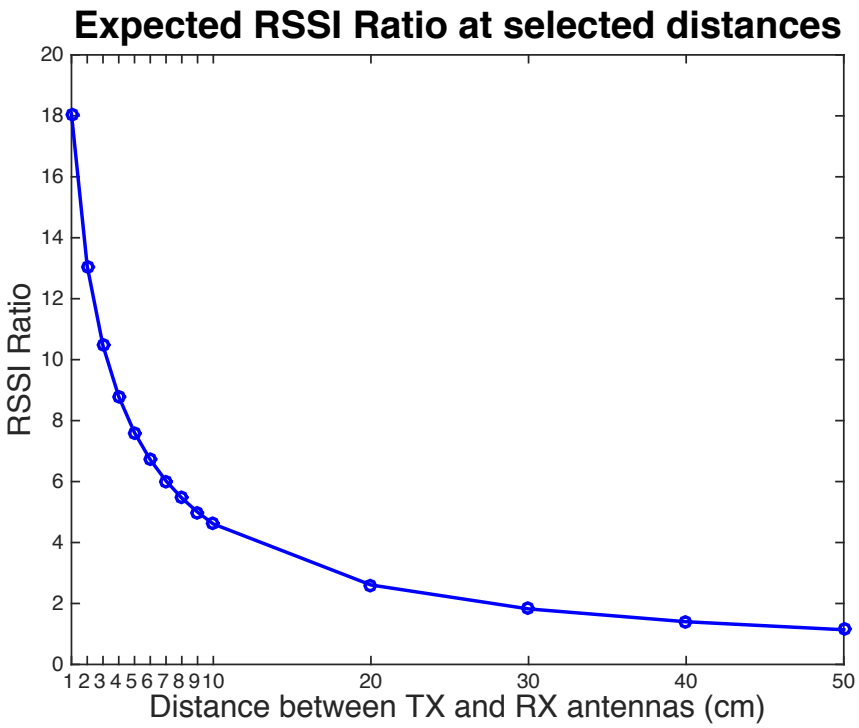

Figure 2.3: Expected difference in RSSI using Equation (2.6) with d_1 ranging from 1 to 50 cm. The difference in RSSI readings increases rapidly as distance decreases.

the Wand moves from 30 cm to 1 cm between the transmitter and A_1. Figure 2.3 shows how the expected power changes as the distance between the device and transmitter changes.

Wanda determines whether the Wand and device are in close proximity by examining the average RSSI Ratio according to the following procedure:

$$\bar{\delta} = \frac{1}{\omega} \sum_{i=1}^{\omega} r_1(i) - r_2(i) \tag{2.9}$$

where i is the i^{th} frame received and $r_1(i)$ is the RSSI for frame i measured on antenna A_1, $r_2(i)$ is the RSSI for the same frame measured on antenna A_2, and ω is a window containing the RSSI of the most recent frames received.

If the average difference $\bar{\delta}$ rises above a predetermined threshold τ, then the Wand declares it is in close proximity to the transmitting device. The Wand waits

to check for proximity until it has received at least ω frames, and re-checks for proximity every $\omega/2$ frames afterward using the last ω RSSI values until it detects it is close to the transmitter or times out.

In this way, the Wand can determine when it is in close proximity to a transmitting device even if that device has only a single antenna. If the device has multiple antennas, Wanda assumes it will transmit frames using only one of its antennas and will not change transmitting antennas while executing the *detect* primitive.

In keeping with *declarative identification* discussed above, to execute *detect*, a user expresses the intent to start the process by taking an action such as pressing a button on the target device. The target device then begins broadcasting an *AssocReq* frame every 50 ms indicating that it is looking to connect with another device. The Wand uses those broadcast frames to determine whether it is in close proximity to the device using Equation (2.9).

The Wand can provide its user visual or audio feedback to encourage the user to move the Wand closer if needed. The Wand can change a row of LED lights or increase (decrease) the frequency of an audio tone if the spread between RSSI readings on the two antennas is becoming larger (smaller) to indicate if the Wand is getting closer to (farther from) the target device. Additionally, a visual indicator such as a sticker bearing a Wanda logo could be affixed on top of the antenna location on the target device to make *detect* easier. The user would then simply move the Wand close to the sticker and initiate the *detect* process. See Figure 2.4 for an example of how a logo could be affixed to a blood-pressure monitor.

Once the Wand determines that it is in close proximity to the device, it sends an *AssocAck* frame to the target device. The target device receives the *AssocAck*, stops transmitting frames, and begins listening for *Message* frames from the Wand.

Figure 2.4: Blood-pressure monitor with Wanda logo indicating where to place the Wand. Photo by Eli Burakian.

2.3.2 Impart primitive

After devices are in close proximity, the Wand can exploit a property of radio wave propagation called *reciprocity* to impart information onto the nearby target device. Reciprocity says that a signal will experience the same multipath properties (e.g., attenuation, phase shifts, and delays) in both directions of the link [96]. This means that transmitting from the target device to the Wand has the same fading characteristics as transmitting from the Wand to the target device. As we saw above, the Wand should see a large RSSI Ratio when a transmitting device is close to the Wand. Similarly, due to reciprocity, the device should see a large difference in RSSI when the Wand transmits from antenna A_1 versus when it transmits from antenna A_2.

Wanda exploits the expected difference in RSSI on the target device to impart information. The Wand first converts the data to impart onto the device into a

binary string m and then sends m one bit at a time. To send a 1, the Wand sends a *Message* frame using the closest antenna, A_1. To send a 0, it sends a *Message* frame using the farthest antenna, A_2. If the Wand and device are physically close together, the device will see a large difference in RSSI depending on which antenna the Wand used. For example, if we assume as above that the Wand is pointing directly at the device and the distance d_1 between A_1 and the device is 3 cm, then with $\alpha = 2$, the difference in signal strength received on the device between a frame sent by antenna A_1 versus A_2 would be about 10.5 dBm by Equation (2.6).

To test the expected signal strength difference, we measured the RSSI of 1,000 Wi-Fi frames sent with antenna A_1 located 3 cm from the receiver, intermixed with 1,000 frames sent with antenna A_2 located 10 cm from the receiver. Figure 2.5 shows the result. It is clear that there was a large difference in RSSI depending on which antenna sent the frame. In this case, the RSSI values are consistent with Equation (2.6) with the path loss exponent $\alpha \approx 1.6$.

To decode the message m sent by the Wand, the target device simply calculates the average RSSI over all frames received and then compares the RSSI value for each frame with the average RSSI over all received frames. If the RSSI for an individual frame is above the average, the target device declares the frame to be a 1. If the RSSI is below the average, the target device declares the frame to be a 0. More formally:

$$\bar{r} = \frac{1}{n} \sum_{i=0}^{n} r(i) \tag{2.10}$$

$$\hat{m}(i) = \begin{cases} 1 & \text{if } r(i) \geq \bar{r} \\ 0 & \text{if } r(i) < \bar{r} \end{cases}$$

where $r(i)$ is the RSSI measured on the single antenna of the target device for frame i and $\hat{m}(i)$ is the i^{th} bit in the message received. Once complete, the device will have

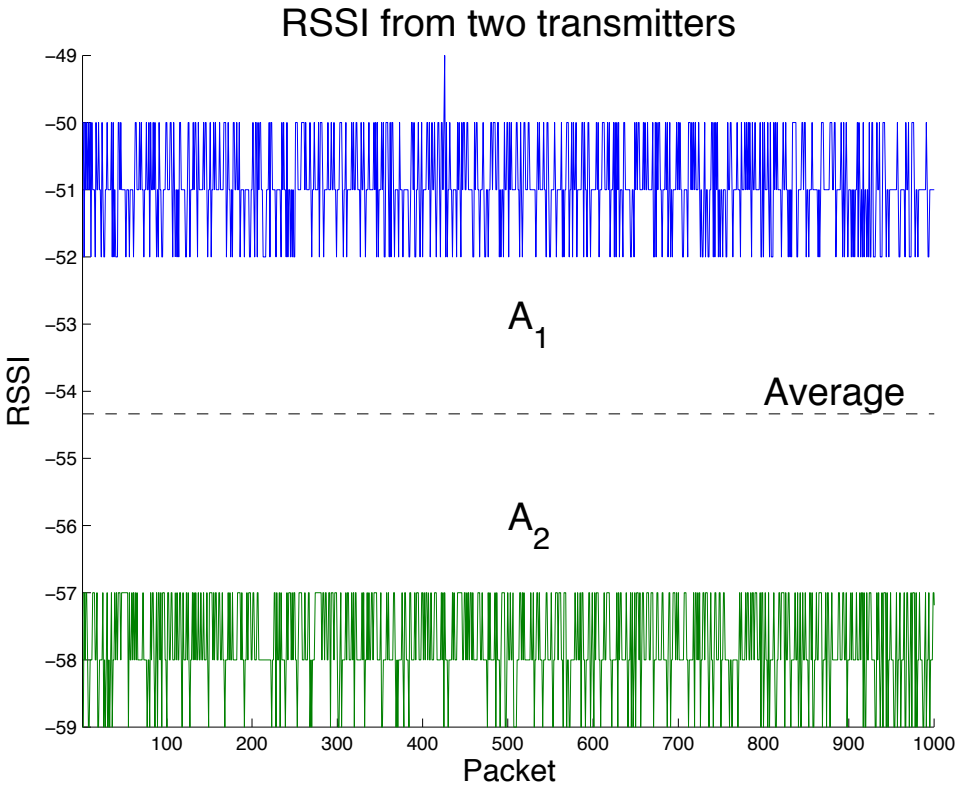

Figure 2.5: Large RSSI difference experimentally observed at close range. There is a large difference in RSSI received from 1,000 Wi-Fi frames sent by antenna A_1 located 3 cm from the receiving antenna, compared with 1,000 Wi-Fi frames sent by antenna A_2 located 10 cm from the receiving antenna.

string $\hat{m}$ representing the string m sent by the Wand.

To ensure the target device is not missing any bits in message m due to dropped frames, each *Message* frame sent by the Wand carries an increasing sequence number in the payload. The target device uses the sequence number of each frame to determine whether it missed any frames. If any frames are missing, the target device requests a resend of only those missing frames; otherwise it sends an empty list to the Wand.

To be clear, the information is transferred using the RSSI alone – the frames themselves sent do not contain portions of the message m. The payload only contains a sequence number so the target device can identify any missing bits.

The Wand sends the entire message without waiting for acknowledgement from the target device. When all message bits have been transmitted, the Wand sends a *Done* frame. The *Done* frame is similar to a *Message* frame, but it also includes a hash of m in the payload. Once the target device receives the *Done* message, it computes the value for each bit using Equation (2.10), creating message $\hat{m}$ on the target.

Finally, the target device hashes $\hat{m}$ and compares it with the hash of m included in the *Done* frame. If the hashes match, the target received all of the frames correctly. If the hashes do not match, the target device tries flipping each bit in $\hat{m}$, one at a time, re-hashes, and compares with the hash sent by the Wand. If a match is still not found, the target device follows a similar pattern but tries flipping two bits. If a match is still not found, the target device signals the Wand to restart by sending a *Restart* frame. If a match is found, the target device transmits a *Success* frame to the Wand.

If the message to be imparted is long, it could be sent in chunks to enable the target device to efficiently flip bits. On the other hand, if messages are short they may be susceptible to an adversary discovering the message by brute-force flipping

bits and hashing. To protect against these potential exploits Wanda can chunk long messages and pad short messages into 128-bit messages.

To demonstrate the *impart* primitive, we converted the message "hello" into binary and sent it to a target device using the *impart* primitive. Figure 2.6 shows the results. The horizontal axis shows the correct bit value for each frame and the vertical axis shows the RSSI measured for each frame. Frames representing bit values of 1 should be received on the target device with an RSSI above the average and frames representing bit values of 0 should be received below the average. Circles indicate frames where the bit value was correctly resolved by the receiver using Equation (2.10) and X's indicate frames where the bit value was incorrectly resolved. We see the message was easily decoded at a distance $d_1 = 3$ cm and had many errors with $d_1 = 30$ cm.

2.4 Protocols

Wanda uses the primitive operations *detect* and *impart* described above to build protocols for configuring new devices. In this section we define three higher-level protocol operations: (1) *Common Key*, where a target device is imparted with parameters that are common to all devices in a local-area network, (2) *Unique Key*, where two devices connect with a key unique to that pair of devices, and (3) *Copy and Paste* where the Wand copies data from one device and pastes it onto another without creating a lasting bond between devices.

2.4.1 Common Key protocol

The *Common Key* protocol is used when a new device must be configured with information common to all devices in a local-area network such as the blood-pressure monitor described above. The blood-pressure monitor must learn the SSID

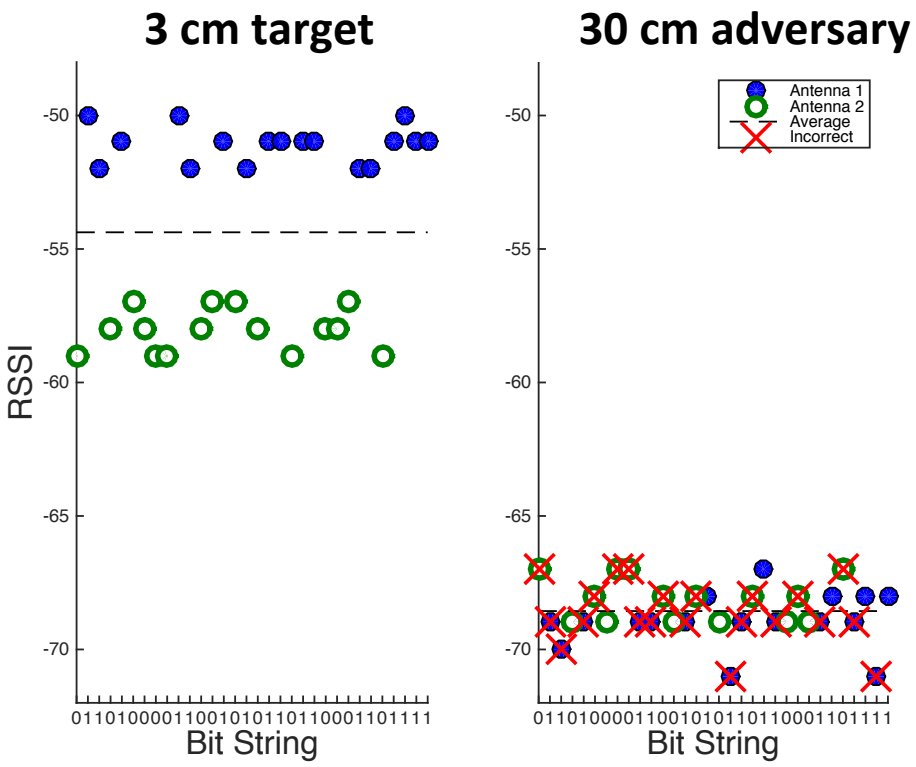

Figure 2.6: Receiving a message at 3 cm and 30 cm ranges. A message m of “hello" was sent at distances of $d_1 = 3$ and 30 cm. Solid blue circles represent frames sent by antenna A_1. Open green circles represent frames sent by antenna A_2. X’s represent bit errors. The message was received with no errors at 3 cm, but had numerous errors at 30 cm.

and password of a Wi-Fi AP. In this case we expect the Wand has earlier learned the SSID and password from the Wi-Fi AP over a wired USB connection. One can imagine the Wand being a 7 cm stick that lives in the USB port of the AP, keeping its batteries charged so it is ready when needed, and using the USB to securely obtain the connectivity parameters from the AP.

The Wand and target device then implement the *Common Key* protocol as follows: the Wand and target device run the *detect* primitive to determine whether they are close together. Once the Wand determines it is in close proximity to the target device, it runs the *impart* primitive to send the SSID and password to the target device. After the target device has confirmed it has properly received the message, flipping bits if necessary as described in the *impart* primitive, the target device connects to the Wi-Fi AP using the SSID and password it received, and the Wand is then not required for future communications.

2.4.2 Unique Key protocol

A slightly more complicated scenario arises when a user wants two devices to establish a connection using a key that is unique to those two devices. In this case the Wand can facilitate the introduction of the devices. The *Unique Key* protocol starts with the Wand generating a random key R. The Wand and Device 1 run *detect* and *impart* to send R to Device 1. The Device 1 includes its IP address (if it has one) in the payload of the *Success* message at the end of *impart* and the Wand notes the IP address as well as the MAC address of the target device from the frame headers. The user then carries the Wand close to Device 2 and the Wand then imparts R plus the MAC and IP address of Device 1 to Device 2 using *detect* and *impart*. Device 2 can now open direct communications with Device 1 by sending a hash of R to Device 1 at the MAC or IP address obtained from the Wand. Device 1 receives the hash from Device 2 and hashes its own copy of R. If the hashes match, then Device 1

bootstraps a MAC- or IP-layer connection with Device 2 using R as an initial key. If the hashes do not match, Device 1 does not attempt the connection.

2.4.3 Copy and Paste protocol

A third Wanda protocol is *Copy and Paste*. In *Copy and Paste* one device has information that the user would like imparted onto another device, although there may be no need for the devices to form a lasting pair as in the *Common Key* or *Unique Key* protocols. An example of where *Copy and Paste* could be useful is the blood-pressure monitor scenario described above. As shown above, the patient can use the *Common Key* protocol to link the blood-pressure monitor to a Wi-Fi AP, and while that solves the problem of getting a long-range communication connection for the short-range blood-pressure monitor, it does not solve the problem of getting the data stored in the patient's Electronic Health Record. For data storage to happen the blood-pressure monitor must know *where* and *how* to send the data. The blood-pressure monitor must know things such as a Restful API URL to send the medical readings, as well as the patient's credentials such as ID and password so the data can be stored in the correct patient record in the EHR.

Copy and Paste is designed to solve this problem. Continuing with the medical example, the patient brings the Wand to the doctor's office and performs the *Copy* phase by using *detect* and *impart* to send a random key R onto a device in the doctor's office. The doctor's office device encrypts the patient's credentials using R as a key and sends the resulting cypher text c to the Wand. The Wand stores the cypher text until the patient returns home. The patient then performs the *Paste* phase by using *detect* and *impart* to send random key R and cypher text c to the blood-pressure monitor. The blood-pressure monitor then decrypts the data and begins sending data to the doctor while the Wand deletes the cypher text. In this way, the *Copy and Paste* protocol copies the data from one device and pastes it onto

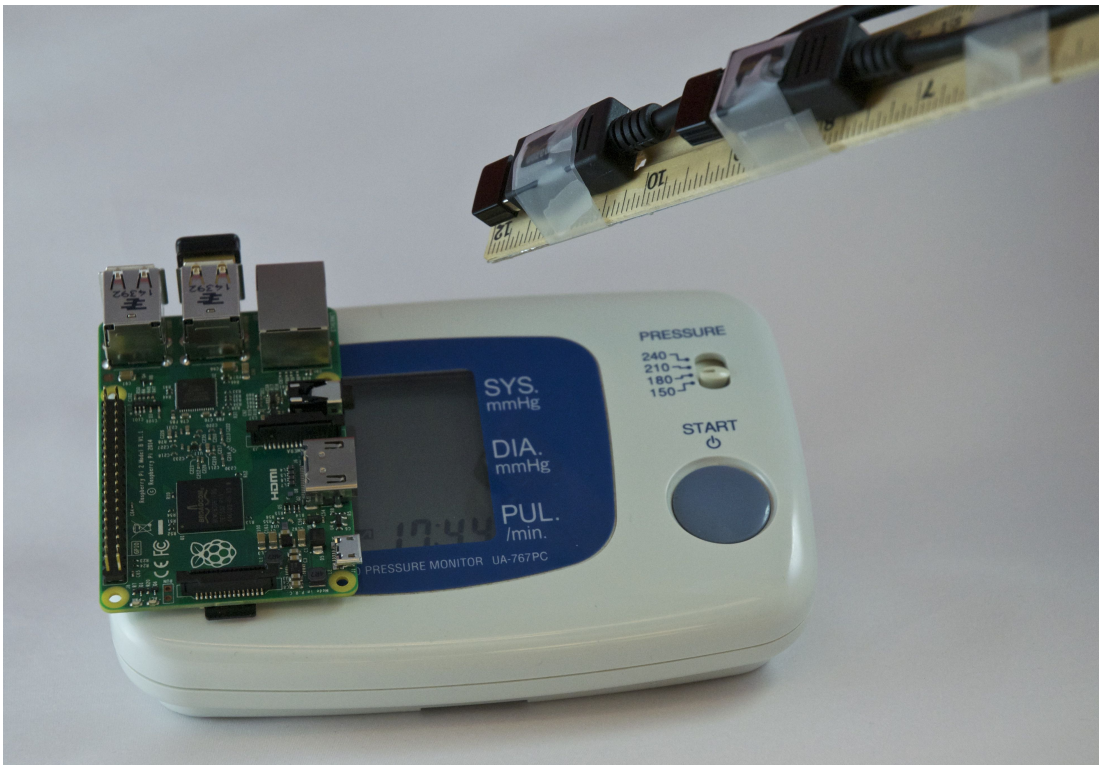

Figure 2.7: Prototype Wand and A&D Medical blood-pressure monitor as target device (some cables removed for clarity). Photo by David Kotz.

another device, even though the devices may be physically far apart.

2.5 Implementation

We implemented a Wand prototype using a Raspberry Pi 2 Model B computer [97] connected to two external Panda Ultra Wireless N USB Wi-Fi adapters [86]. Figure 2.7 shows a photo of the prototype Wand and medical device. A production version would benefit by using one Wi-Fi card that has multiple antennas (commonly found on 802.11n or 802.11ac Wi-Fi devices). This single-radio, dual-antenna approach would ensure consistent energy is transmitted by the two antennas and could help reduce the potential for fingerprinting attacks [24, 61] by generating the radio frequency energy from the same source.

We used an FDA-approved A&D Medical UA-767PC blood-pressure mon-

itor [1] as the target device. Because we were unable to modify the software on FDA approved medical devices, we added an external Raspberry Pi with a single Alfa Networks AWUS036H Wi-Fi antenna [4] and connected to the blood-pressure monitor using a RS-232 over USB connection. This gave us the ability to extract the blood-pressure readings from the blood-pressure monitor using the RS-232 connection and the ability to communicate with the Wand over the single Wi-Fi antenna. Of course the manufacturer of the medical device would be able to alter their software to include the Wanda protocols (Wanda does not require hardware modification as long as the device has Wi-Fi connectivity), but our prototype demonstrates that even an existing device without a radio can be easily retrofitted to the conform to Wanda's protocols. We imagine the retrofit device to be a small dongle instead of our prototype Raspberry Pi-based system.

We then used the prototype Wand to impart two types of information onto the retrofitted blood-pressure monitor located in a busy computer science lab. First we imparted the SSID and password of a local Wi-Fi AP so the device could establish a connection and get to the Internet. Second, we imparted the URL and a username and password for a Restful API representing a web service end point into a medical Electronic Health Record (EHR) in the cloud. The result is that now when someone measures their systolic pressure, diastolic pressure, and pulse, the Raspberry Pi reads those measurements and securely passes them to the simulated EHR.

We used Python and Scapy to create Layer 2 Wi-Fi frames in our prototype. While our prototype used Wi-Fi, the technique could also be adapted for other protocols such as Bluetooth or Zigbee.

Distance	*Detected close*
< 5 cm	100%
5 cm	87%
6 cm	38%
> 6 cm	0%

Table 2.2: Percentage of time when the *detect* primitive declared proximity. The Wand implemented *detect* and successfully discerned proximity with high accuracy at close range while correctly determining it was not in proximity at longer ranges.

2.6 Results

We evaluated both the *detect* and *impart* phases of Wanda. For the evaluation we used the same software as our prototype, but for easier control and monitoring of our experiments we used a MacBook Pro instead of a Raspberry Pi. All experiments were conducted in a busy computer science lab.

2.6.1 Detect tests

We conducted 1,000 trials of the *detect* primitive where the distance d_1 between the Wand's A_1 antenna and the device's antenna ranged between 1 and 50 cm. Trials were conducted at 1 cm intervals from 1 to 10 cm, then at 10 cm intervals from 10 to 50 cm for a total of 14 distances with 1,000 trials each. The percentage of trials where the Wand detected proximity is shown in Table 2.2 using a window size $\omega = 20$ and a threshold value $\tau = 6.2$. We chose this value for τ because the equations in Section 2.3 estimate that *detect* will declare the devices in close proximity when d_1 is less than 6 cm. We found that at distances less than 5 cm, proximity was detected 100% of the time. At 5 cm proximity was detected 87% of the time, and at 6 cm proximity was detected 38% of the time. At distances longer than 6 cm proximity was not detected. These results suggest that *detect* was able to correctly determine when it is in close proximity to the device with high probability.

2.6.2 Impart tests

We tested Wanda's ability to correctly impart data by first confirming the RSSI differences behaved as expected, then sent 1,000 messages from the Wand to the target device at various distances and counted bit errors to determine the Wand's effective range. Finally we measured how fast the Wand could impart information on target devices.

RSSI differences

To confirm that a single-antenna device is able to correctly receive a message when using the *impart* primitive, we tested whether it would consistently measure a significant difference in RSSI based on the Wand's transmitting antenna (A_1 or A_2) as predicted by the equations in Section 2.3. In these tests the Wand sent 1,000 Wi-Fi data frames from each of its two antennas, alternating between antenna A_1 and A_2, where the distance d_1 between antenna A_1 and the target device ranged from 1 to 50 cm and the distance d_2 was 7 cm larger than d_1. For these experiments, each *Message* frame contained a sequence number as specified in the *impart* primitive, as well as an indication of which antenna sent the frame to avoid confusion over which antenna actually sent the frame.

The target device recorded the RSSI of each frame and calculated an RSSI difference for each of the 1,000 pairs of frames it received. The results are shown in Figure 2.8 along with the RSSI difference predicted by Equation (2.6). The plot shows that the values observed mirror the predicted values when $\alpha = 1.6$.

Bit errors

Next we measured how well the Wand was able to impart information on another device. We ran 1,000 trials where the Wand sent a 128-bit random message to a

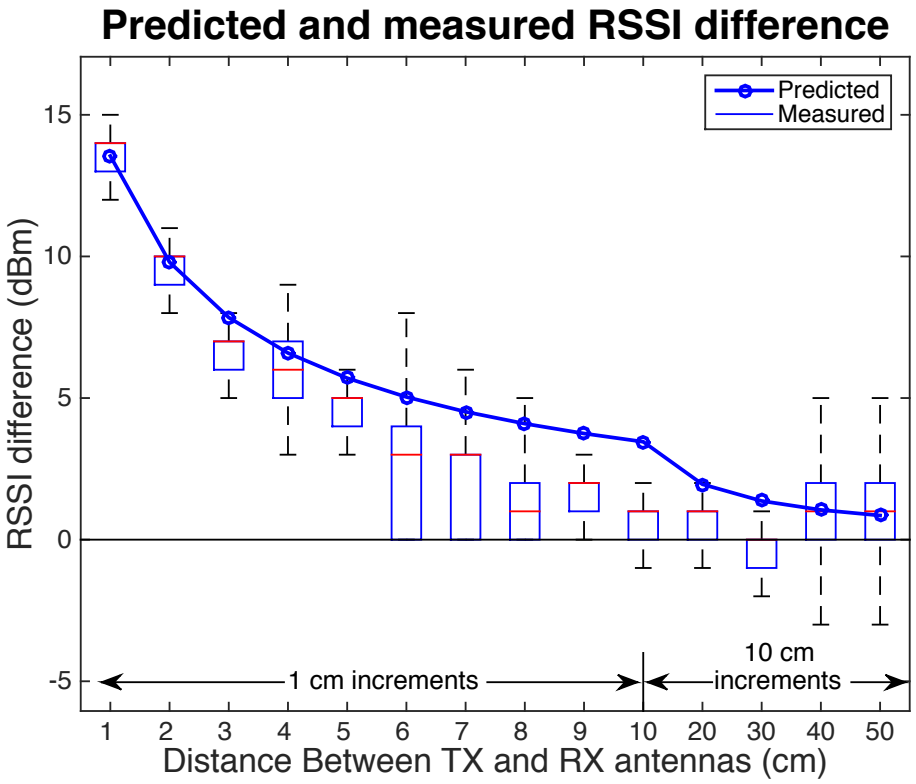

Figure 2.8: Observed RSSI differences on a single-antenna device from 1,000 pairs of frames sent by the Wand alternating between antennas. The box represents the 75th and 25th percentiles of the observed RSSI differences, the red line is the median, and the whiskers represent the range of differences. The predicted RSSI difference according to Equation (2.6) is shown with $\alpha = 1.6$.

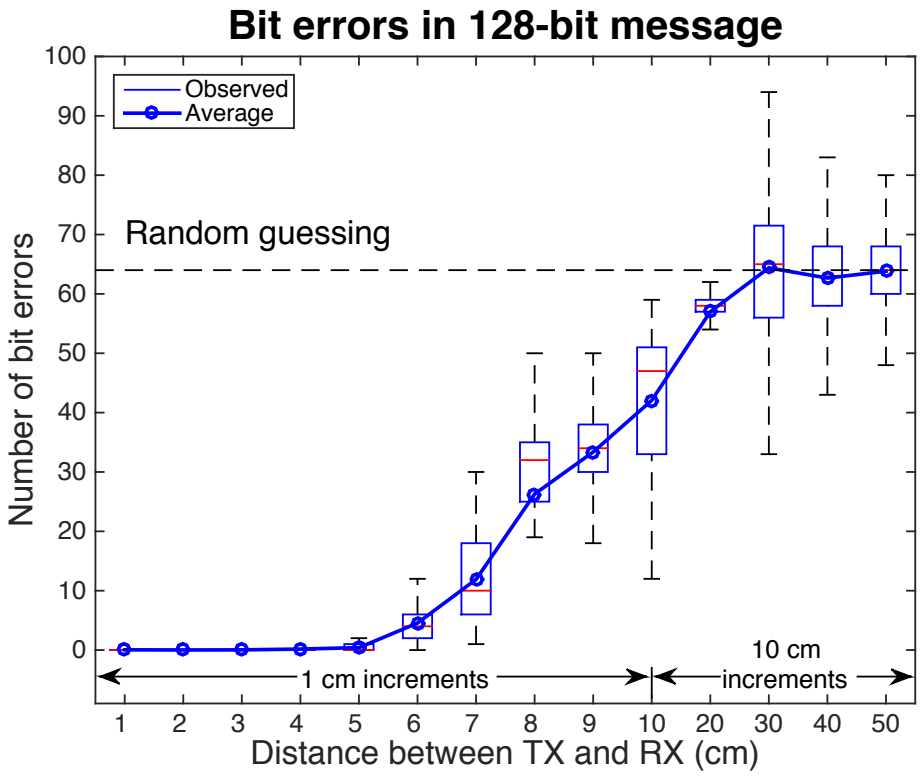

Figure 2.9: Bit errors decoding a 128-bit message. The box represents the 75th and 25th percentile, the red line is the median, and the whiskers represent the range of bit errors.

single-antenna target device, and then counted the number of mismatched bits. Figure 2.9 shows that very few bit errors occurred at close range, but the number of errors increased significantly as distance between the Wand and the receiver, d_1, increased. Because each message contained 128 bits, random guessing should yield 64 correct bits. In our experiments this began to happen at a distance of about 30 cm.

Some of these errors can be corrected with the bit-flip technique described above where the target device flips bits in its derived message $\hat{m}$ and re-hashes. Figure 2.10 shows the percentage of successful message transfers at distance from 1 to 50 cm, correcting bits when needed, by flipping zero to three bits. From this graph we see that messages were transferred with a high probability of success when the Wand was less than 6 cm from the device. Due to the variability in RSSI

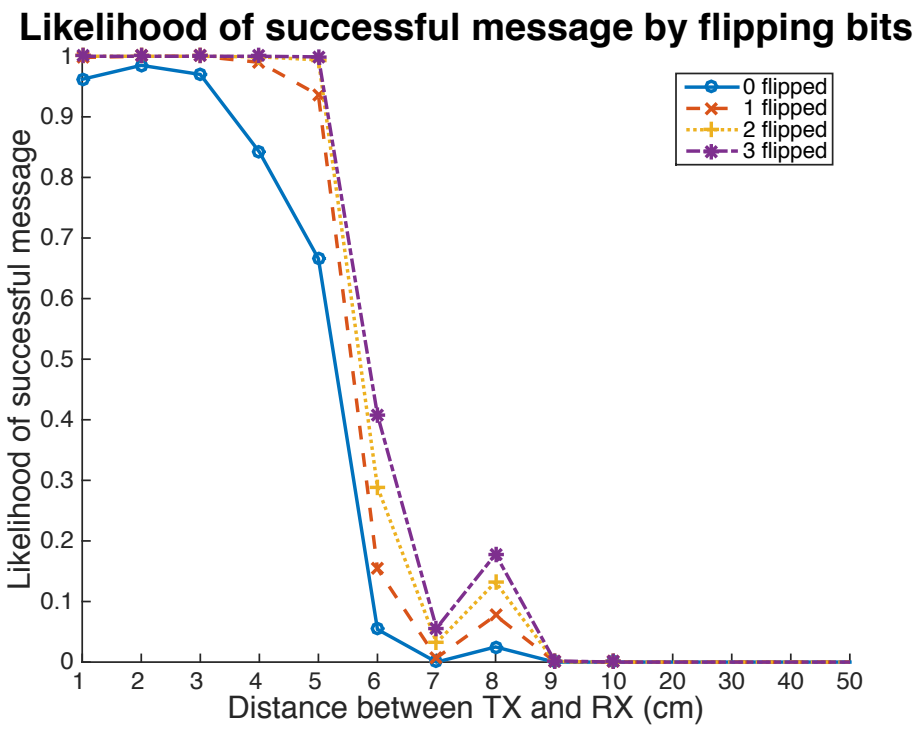

Figure 2.10: Likelihood of successful message by flipping up to three bits. At distances less than 6 cm messages were received with high probability.

Ratio, however, the bit-flipping technique is not effective at long range. This suits a legitimate user well because the devices are close together, but makes a distant attacker's task difficult.

Timing

We also measured the speed at which the Wand was able to impart a message. The average time to send 128 bits was 0.454 seconds, which translates to just over 280 bits per second. We note that our implementation was written in Python. An implementation in C might have seen even faster throughput, although for many applications transferring a message in under half a second is acceptable. Additionally, long messages can be sent by first imparting a key and then using that key to encrypt normal frames carrying data in their payload.

2.7 Security

In prior sections we show that Wanda works well; in this section we evaluate its security against passive adversaries attempting to eavesdrop on communications between the Wand and the target device, and active adversaries attempting to inject malicious information onto the target device or Wand. We assume an adversary has complete knowledge of the Wanda protocol and can use that knowledge to try to exploit the system.

We assume the adversary:

- is able to receive, tamper with, or inject frames into the communications between the Wand and target device;
- is able to modulate its transmit power;
- may have multiple antennas and be positioned at multiple locations;
- does not try to jam the communications channel, creating a denial of service;
- does not have physical access to tamper with the Wand or target device; and
- is located more than 30 cm away from the target device and Wand while they are communicating.

2.7.1 Eavesdropping

Because the bits in the message m sent by the Wand are encoded only in the Wand's choice of transmitting antenna, an adversary must determine which antenna sent a frame in order to decode the information transferred. There are four main ways this could be done by an adversary: (1) receive frames from only one Wand antenna, (2) use the environment to differentiate between antennas, and (3) analyze the

RSSI to differentiate between antennas and (4) use signal processing and MIMO antennas.

Receive frames from only one Wand antenna

If it were possible for an adversary to receive frames sent by only one of the Wand's antennas – not both – the adversary would be able to determine which antenna sent all of the bits in a message. The adversary would simply list the frame sequence numbers it receives and infer those frames represent a bit with a value of 1. For the sequence numbers the adversary does not receive, it can assume those frames came from the other antenna on the Wand and infer those represent a bit value of 0. After all the frames are sent, if the adversary does not drop any frames, the adversary will either be correct on all bits (the monitored antenna was actually sending 1s), or wrong on all bits (the monitored antenna was actually sending 0s) in which case the adversary simply flips all bits. The hashed message contained in the *Done* frame would let the adversary determine which value is correct.

The adversary's dilemma is that both antennas on the Wand are close together and radiate energy that travels outward in a spherical shape. This makes receiving signals from only one antenna very difficult. An adversary could try to use a highly directional antenna and attempt to create a narrow main lobe pointed precisely at one of the antennas on the Wand. Given that the antennas on the Wand are only 7 cm apart, this approach is unlikely to work if the adversary is located a reasonable distance away because the main lobe expands with distance and should encompass both of the Wand's antennas. For example, as shown in Figure 2.11, an adversary located 1 m away and bore-sighted on one of the Wand's antennas would need to have a one-half beam width of $\alpha = \tan^{-1}(7/100) \approx 4$ degrees to avoid receiving a signal from the Wand's second antenna.

An adversary might use an antenna array in an attempt to create a narrow

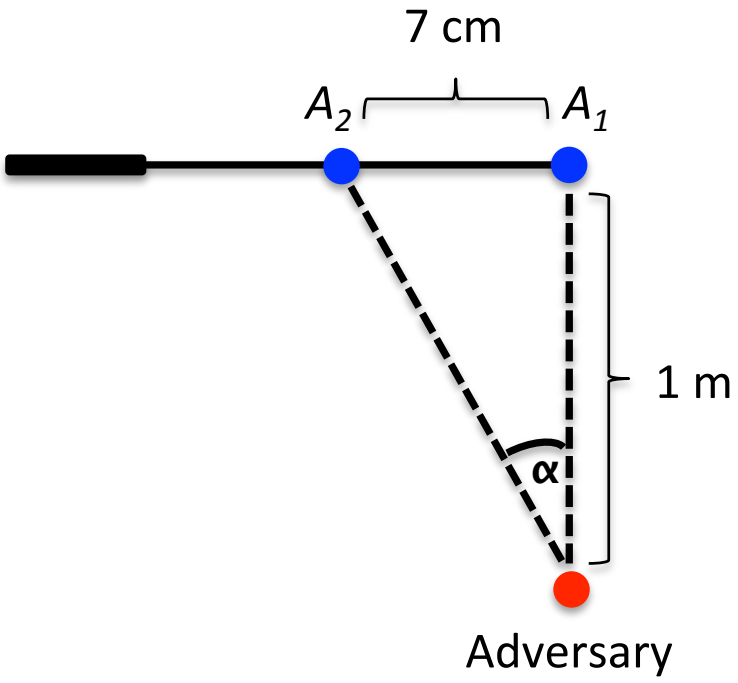

Figure 2.11: Adversary beam width. Given the antennas on the Wand are separated by 7 cm, an adversary located 1 m away and bore-sighted on one of the Wand's antennas, would need a one-half beam width of $\alpha = \tan^{-1}(7/100) \approx 4$ degrees to avoid receiving a signal from the Wand's second antenna.

beam and only receive the signal from one of the Wand's two antennas. Using software provided by Balanis [9] and assuming the antennas are spaced 0.25λ apart, Figure 2.12 shows the resulting normalized radiation pattern for a 10-antenna array. This radiation pattern is the same for both transmitting and receiving, and demonstrates that the array can create a narrow beam focused along the z-axis. In this case, even with a large number of antennas sharpening the beam, the Half-Power Beam Width (e.g., the width were the receive power is one-half that of the maximum and commonly considered to be the resolution at which an antenna array can separate two sources [9]) can be shown to be 38.64 degrees – far wider than beam width required for an adversary to receive from a single Wand antenna if the adversary is located 1 m away. If the adversary is located more than 1 m, it would need an even smaller beam width than 4 degrees.

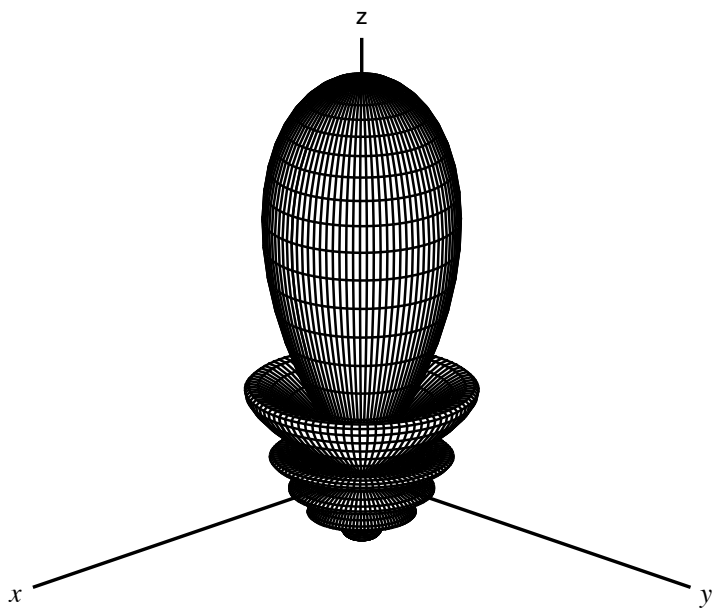

Figure 2.12: Antenna array radiation pattern. A 10-antenna array with antenna spaced at 0.25λ creates a focused beam. This array can be shown to have a Half-Power Beam Width of 38.64 degrees [9].

Use the environment to differentiate between antennas

An adversary might also attempt to determine which antenna sent a frame by detecting differences in the signal due to environmental effects. Because the characteristics of the received signal depend on the specific paths taken as the signal travels from the transmitter to the receiver, and signals from different transmit antennas might take different paths to an adversary, the adversary might be able to determine which antenna sent each frame. The chances of this attack succeeding, however, are vanishingly small. Cai et al. calculated the odds of an adversary succeeding with this type of attack from a random location to be 10^{-15} [18]. They go on to suggest that, in theory, an adversary might choose an ideal location by carefully measuring locations, geometries, and surface properties of all objects in the environment. While this precise measurement is practically impossible, never-

theless even that attack could be mitigated by incorporating a frequency-hopping scheme where each frame is sent on a different Wi-Fi frequency.

Analyze the RSSI to differentiate between antennas

Wanda uses a simple algorithm on the target device to determine which antenna sent a frame based on the RSSI, but we assume an adversary can use more sophisticated techniques. While we cannot anticipate every possible technique, we expect from Equation (2.6) that the difference in RSSI when the Wand uses antenna A_1 vs. when it uses antenna A_2 will be small when the Wand is not close to the adversary. Additionally, the environmental noise described in Section 2.2 increases as distance increases. Figure 2.13 illustrates these differences for 1,000 frames sent by antenna A_1 and 1,000 frames sent by antenna A_2 at $d_1 = 3$ cm and $d_1 = 50$ cm. As expected, the RSSIs of frames from the same transmit antenna form a Gaussian with a distinct mean (due to distance) and standard deviation (due to noise).

If an adversary were somehow armed with perfect knowledge of the Gaussians of each antenna on the Wand, they might be able to determine which antenna sent a frame. When a frame arrives, the adversary could measure the RSSI and determine from which distribution that sample is drawn, that is, which antenna is most likely responsible for sending the frame. The distributions are constantly changing due to changing environmental factors, however, making this assumption of a priori knowledge of the Gaussians unrealistic.

Even if an adversary somehow did have perfect knowledge of the Gaussian distributions that characterize frames sent by each antenna on the Wand, the adversary will still suffer from a large number of errors when observing from long distances. Figure 2.14 shows that, even if armed with perfect knowledge of the frame distributions, an adversary only a short distance away would still make nearly 50% bit errors predicting which antenna sent a frame using the Gaussian

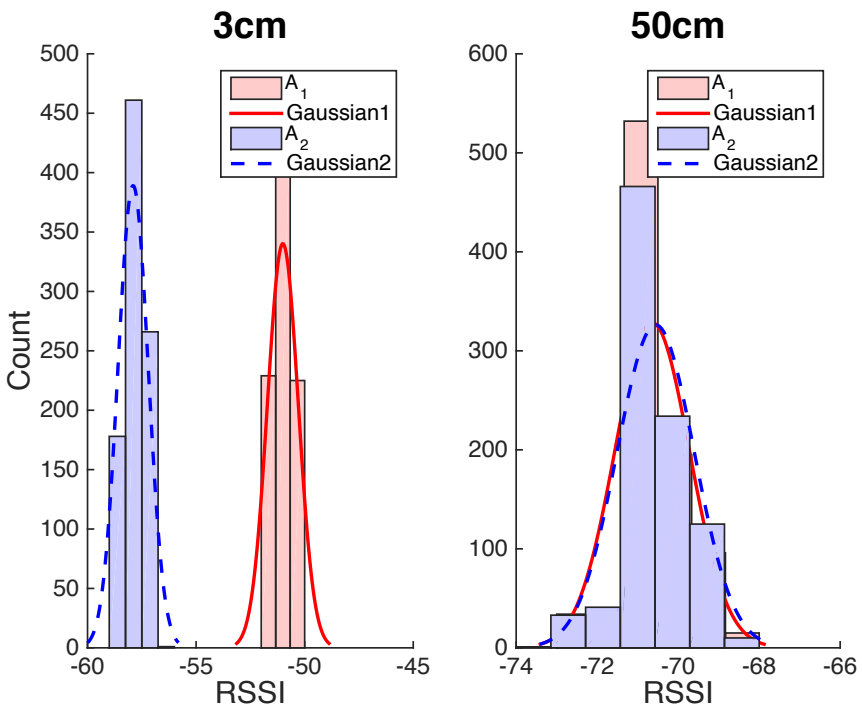

Figure 2.13: RSSI distribution of 1,000 frames sent where d_1 = 3 cm and 50 cm. At close range there was a distinct difference between antennas whereas at longer distances the gaussian distributions of frame RSSIs heavily overlapped.

distributions. We conducted those experiments with a prototype built with two radios (rather than one radio), cheap antennas (not specifically selected for a spherical radio dispersion pattern), and without precise antenna alignment (see Figure 2.7); a commercial Wand (with a single radio and two antennas selected and aligned carefully) would be even harder to attack in this manner.

Analyze the signal phase to differentiate between antennas

The phase of a received signal is dependent on the distance between the transmitter and receiver. A sophisticated adversary may try to use Channel State Information (discussed in Chapter 3) to measure the difference in signal phase to determine which of the Wand's antennas sent a frame. The antennas on the Wand are located close together, suggesting the difference in distance between an adversary and each of the Wand's antennas will be small and phase of the received signal will likely be

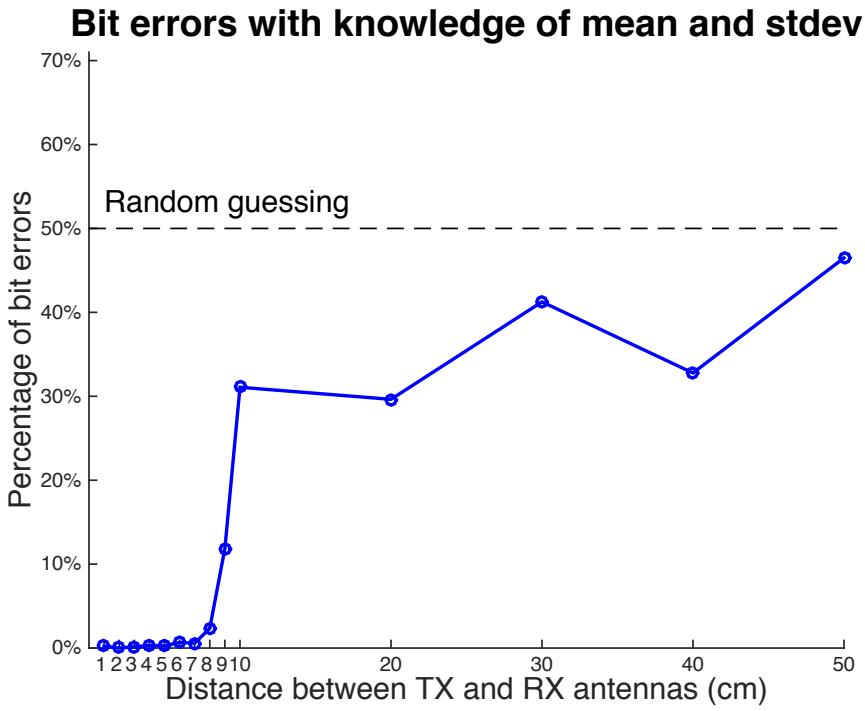

Figure 2.14: Percentage of bit errors if adversary had perfect knowledge of RSSI distributions by antenna. Even with the unrealistic assumption of perfect knowledge, an adversary would still make numerous errors.

similar for both antennas.

It may be the case, however, that an adversary in a particular location is able to reliably discern a difference in phase from signals sent by each the Wand's antennas. A simple countermeasure would be add a different random phase offset in the range $[0, 2\pi)$ radians to each frame the Wand transmits. The target device uses the signal strength of each frame to determine which antenna sent the frame, and does not consider the phase of the signal to reconstruct the Wand's message. The added phase would confound a phase-observing adversary while not affecting a signal-strength-observing target device.

2.7.2 Frame injection

An active adversary may attempt to inject information onto the target device by tricking the target device into believing it is communicating with the Wand while

the Wand is not actually present. Wanda defends against the attack by asking the user to declare intent to start the protocol on the target device by taking an action such as pushing a button on the target device. This ensures that when the Wand is not present, the target device will not begin running the Wanda protocols. In that case, if an adversary were to try to communicate with the target device, the target device would not respond.

Alternatively, an adversary could try to override the information sent by the Wand while the Wand is communicating with the target device. To override the Wand, an adversary might modulate its transmission power; increasing power to send a 1 and decreasing power to send a 0. The target device, which may have only a single antenna, has no way of knowing if these modulated signals are coming from a nearby Wand or from a distant adversary because the RSSI of the frames would appear to the target device in the same way frames appear from the Wand. To prevent this attack, the Wand can monitor for rogue *Message* frames that it did not send. If it detects rogue frames, the Wand can send a *Stop* frame to the target device to halt the process. In Chapter 4 we discuss a method where the single-antenna target device can determine proximity with a transmitter without the Wand's assistance.

The Wand can protect *itself* from storing malicious data (as in the *Copy and Paste* protocol), by ensuring any received frames have a large RSSI ratio. This test would ensure the data came from a nearby target device, and not a distant adversary attempting to exploit the Wand.

2.8 Related Work

There has been a great deal of research on how people actually configure and manage computing devices. Our interest is focused on how people will introduce

and manage wireless devices such as the blood-pressure monitor described above with limited UI's (e.g., devices with a small number of buttons, not keyboards, and with limited or non-existent displays) into existing networks. This leads to two primary questions studied in the research literature: (1) *who* configures and manages devices, and (2) *how* are new devices actually configured to work with other existing devices.

2.8.1 Who configures and manages devices

There are three main approaches identified in the literature that can be used to configure and manage a new device such as a blood-pressure monitor: (1) use a professional technician, (2) find a local expert, or (3) do it yourself. We briefly discuss the pros and cons of each of these three approaches.

Use a professional technician

One approach to adding a new device such as a Wi-Fi-enabled blood-pressure monitor into a person's home would be for the device manufacturer, or another service entity, to dispatch a professional technician to physically visit the home to install and configure the new device. This approach makes sense for some applications, such as a heart monitor to track a patient that has just undergone cardiac surgery [67]. In this case, the relative rarity of cardiac surgery combined with the dire consequences that could arise from missing the onset of a life-threatening condition make installation by a trained professional desirable.

As the number and variety of deployed IoT devices grows, however, this approach will be economically unattractive. If the predictions of the growth rate of IoT devices are correct (or even partially correct) and billions of connected devices are coming, then relying on professional technicians to configure new devices will be like calling a professional to change a light bulb – impractical. Poole et al.

noted that as residential computing infrastructures have become more complex, even before the wide-spread roll out of IoT devices, professional technical support services have not sufficiently matured alongside the technologies [93].

Recognizing the highly variable equipment and dispersed geographical nature that exists in real-world deployments, researchers have proposed standardized environments such as HomeLab [17] to help deal with the complexity. These proposals, however, are typically aimed at the research community rather than the population at large.

While useful in some instances, practical issues make relying on a professional technician undesirable for configuring and managing many devices.

Find a local expert

Another approach to configuring a new device is to assume the presence of a local tech-savvy expert (sometimes a grandchild) who can configure a new device for the less savvy user. Other researchers have found that there is often a person who is a "technology driver" [65, 77, 91] and takes responsibility for configuring devices on family or friend's networks.

The local expert, however, may not be willing or available to help configure a new device at any given time. One group of researchers developed an algorithm to predict if the local expert is inclined to provide help, citing factors such as the amount of the expert's available free time, and how important the person needing help is to the expert [92]. Often the answer is no, the expert will not help. This causes an issue, of course, if the device user must rely on the local expert to configure a new device. Additionally, the local tech expert may not have experience with a particular type of device that a user would like configured or the user may hesitate to bother the expert yet again to configure the latest device.

Privacy is another factor that further compounds reliance on a local expert.

If someone other than the device user configures a new device, say a tech-savvy neighbor, it is possible the device's data may be visible to the person performing the configuration. This may not be critical in some environments, but in others, such as where a device could reveal a medical condition that carries a social stigma, maintaining privacy could be extremely important to the user. As noted previously, for example, Sayles et al. found HIV-positive patients fear of stigma after disclosure of their condition often led patients to avoid medical services and medications [105]. Even in the less extreme blood-pressure monitor example, where a person's blood pressure may not be as sensitive an issue as HIV, the tech-savvy neighbor may learn the SSID and password of the Wi-Fi AP while configuring the blood-pressure monitor and may then be able to sniff any traffic on the network, including banking, email, or other sensitive matters (although they may need to force devices to re-join the network to observe all four EAPOL handshake packets) [132].

Do it yourself

Finally, a user may elect to configure a new device themselves. As noted above, configuring new devices has historically been difficult and prone to error. Making secure connections between IoT devices such as a Wi-Fi-enabled blood-pressure monitor is difficult for a number of reasons:

Limited UIs

IoT devices tend to have limited user interfaces – they may not have any form of display or keyboard. This makes it impossible to use traditional approaches to pairing devices such as Bluetooth's Simple Secure Pairing [15] where one device displays a numeric code that is entered into a second device. In fact, IoT devices cannot be counted on to have any particular form of inputs such as cameras or accelerometers or even USB ports.

Confusing setup applications

To skirt UI limitations, many IoT devices use a setup application running on a smartphone or laptop to configure the device and establish connectivity. This need for a setup application creates its own set of issues. Users must find the correct setup application for their particular phone or operating system, and find the right app for the specific model of device to be configured, must download it, then must figure out how to connect the setup application with the the new device.

As an example, Table 2.3 shows the setup steps required to configure the popular Fitbit Aria smart weight scale [33]. The Aria has been praised for its easy setup process, but in step 8 the user is required to select their Wi-Fi network from a list of nearby networks, and then must enter the network's password. Selecting the proper network SSID from all nearby networks is trivial if the setup is being done in a remote farmhouse where there are no other networks nearby, but would be significantly more challenging in a crowded city apartment building where most apartments have a Wi-Fi network (which of the 12 "linksys" networks I can see is mine?). Our experiments in the User Study section suggest many users do not know their SSID or password, causing even the Aria's relatively straightforward setup process to fail.

Inconsistent methods

Even if a user does master a setup application from one vendor, it does not mean that they have mastered the setup for all IoT devices. For example, GoPro cameras [45] take a similar approach to the Aria weight scale in that both the camera and the scale create their own Wi-Fi network, but the two devices differ significantly in setup application UI and steps required. The skills learned setting up the scale do not necessarily transfer to setting up the camera.

Finally, if the projected proliferation of IoT devices happens, each of us may interact with dozens of IoT devices on a daily basis. Without a consistent means

of configuring devices, we may have to deal with dozens of setup applications (an app for the washing machine, a different app for the refrigerator, a different app for the lighting system, and so forth). Each of these setup applications will likely have differing UI's and steps users must take. This situation will likely be bewildering and unmanageable for many users. Also, each application widens the potential attack surface that malicious actors may be able to exploit!

2.8.2 How are new devices actually configured

Researchers have proposed many technical solutions to the problem of *how* to securely configure new devices. While the proposed approaches vary widely, they can be categorized into two main groups: out-of-band (OOB) and in-band communications. In OOB solutions a secret key is exchanged between devices over a secondary communications channel that is impervious to observation and interference by an adversary; the devices then bootstrap a secure connection over the primary channel using the information exchanged over the secondary channel. In-band approaches differ in that they only use the primary communication channel to establish a secure connection. In this section we examine some of the proposed solutions and highlight some of their differences with Wanda.

Out-Of-Band

Systems employing an OOB approach use a secondary channel to exchange secret information (e.g., a cryptographic key) that is used to secure the primary channel's communication. While many methods have been proposed, they all convey secret information over secondary channels such as: wired, visual, audio, gesture, or out-of-band radios such as RFID or NFC. These approaches, however, assume the presence of hardware that may not be present on some devices and may also require

Step	*Description*
1.	Go to http://www.fitbit.com/setup.
2.	Click the Get Started button below Aria.
3.	Double-click the .dmg file in your Downloads folder. NOTE: If the file does not appear on your desktop, search for "Fitbit Wi-Fi Scale Setup" using the Spotlight search feature.
4.	Click Get Started.
5.	a. If you don't already have a Fitbit.com account, enter your email address and a password and click Sign up. b. If you have a Fitbit.com account, select Log in to your account and enter your Fitbit.com credentials, then click Log in.
6.	Enter or confirm your personal information and click Next. This information personalizes your Fitbit experience. By default this personal information is only visible to your friends, but you can adjust your privacy settings from your Fitbit.com account to control what information is shared with others.
7.	Name your scale and input your initials, then click Next.
8.	The setup software will attempt to detect your Wi-Fi network. If the software does not detect your Wi-Fi network, it will show all visible networks in range. **Select your network from the list.** If your network is password-protected, enter your password in the box to the right and click Connect. NOTE: If your network is hidden or does not appear in this list, select Add Network. Enter your network's name and password and click Connect to continue.
9.	If you have not already put your scale into Setup Mode, the onscreen instructions will direct you to do so by removing a battery for 10 seconds and then reinserting it.
10.	Click Connect to start searching for your scale. When your computer finds the scale, the setup software will show a success screen. The software will automatically transfer the wireless network credentials to the scale, and your scale's display will show a checkmark.
11.	Click Done, place the scale on a hard surface, and weigh yourself to start tracking your weight.
12.	Following a weight measurement, a checkmark on the scale's display will indicate a successful sync with your Fitbit.com Dashboard. NOTE: If you have trouble setting up your Aria using the Fitbit Wi-Fi Scale Setup software, please try the web-based setup method at http://fitbit.com/scale/setup/start.

Table 2.3: Steps to configure a Fitbit Aria scale using Mac OS X. Users must go through several steps and must know their Wi-Fi SSID and password to complete the configuration (from www.fitbit.com/setup/aria; emphasis on step 8 is ours).

complex processing that exceeds the capabilities of embedded devices.

The wired approaches assume the presence of a cable that is able to connect devices [62, 119]. Many devices today, however, do not have a compatible wired interface or are mobile, making a cable run impractical.

The visual channel assumes the presence of cameras to capture an image such as QR codes [16], 2D bar codes [76], or blinking LEDs [104, 103], and may also require displays on some of the devices to show changing codes [108]. The visual channel also normally requires a great deal of computational capability on the devices to analyze the images captured.

The audio channel requires microphones to listen to environmental noises or human voices [41]. Additionally, the audio channel may require speakers to purposely generate tones [118].

Some gesture approaches require coordinated button presses [117] while others require accelerometers [74]. Aside from the fact that many devices do not have accelerometers or the computational horsepower to run the required algorithms, some sensitive devices may not tolerate being jostled, and some are simply too big to shake.

Approaches requiring secondary radios such as RFID or NFC have also been proposed. These radios, however, have not yet been widely deployed on actual devices. Medical devices in particular tend not to have RFID or NFC capabilities. Unlike smart phones, medical devices tend to have only one radio.

Wanda differs significantly from these all of these approaches in that it does not assume the presence of specialized hardware other than the existing wireless radio, nor does it require advanced processing power. Furthermore, Wanda requires little human effort and the Wand's mobility allows it to be used when devices that are not physically adjacent or would be inconvenient to move (such as a treadmill and a Wi-Fi AP).

In-Band

Researchers have also suggested techniques that do not require a OOB channel, but instead exploit characteristics of the in-band radio channel. These techniques are typically more closely aligned with Wanda than OOB techniques. Although Gollakota et al. developed an in-band method to defend against Man-In-The-Middle attacks [39], their approach relies on changes to the Wi-Fi specification.

Most in-band approaches, however, use characteristics of the radio channel to develop a secret key independently on two devices. To develop the secret key, each device typically goes through several phases. The first phase is *bit extraction*, where each device monitors a common radio channel simultaneously and extracts bits from extreme signal fluctuations to form a string of bits. Next, *reconciliation* attempts to ensure both devices have extracted the same bit string. This normally involves several rounds exchanging information about portions of the bit string, such as checksums, in the clear. Finally, a *privacy amplification* phase reduces the size of the bit string to form a secret key that is known to the participating devices and unknown (with high probability) to an adversary [11]. Several works use a variant of this extraction-reconciliation-amplification approach. ProxiMate [73], for example, uses FM radio or TV signals to generate its bit string, but requires additional radios to tune those signals. BANA [112] and ASK-BAN [113] use on and off-body signal propagation for bit extraction. Zeng et al. use multiple antennas for bit extraction, followed by reconciliation and amplification [137].

The extraction-reconciliation-amplification approach has several shortcomings. First, it is quite slow, often taking 30 seconds or more to make connections. Wanda is fast, taking less than half a second on average to send a 128-bit message. Another problem is that Wi-Fi, in many practical environments where device connections might be made, lacks the necessary entropy to extract a secure bit string [59]. Wanda does not rely on random environmental fluctuations to generate common bits on

two devices; it imparts the bits onto a target device based on the antenna chosen by the Wand.

Wanda does share common elements with two papers. In *Good Neighbor* [18] the authors use a technique similar to Wanda to determine if a sending device with a single antenna is in close proximity to a receiving device with two antennas. *Good Neighbor*, however, runs significantly slower on average than Wanda and requires the use of public key cryptography to transfer data between devices whereas Wanda does not. Wanda could be useful for devices with limited computational and storage capabilities. Additionally, *Good Neighbor* requires two devices to move in close proximity with each other. Wanda uses a mobile Wand that allows devices to be physically distant from each other. This would be useful if the devices are not easily moved, such as a refrigerator connecting with a Wi-Fi AP.

Another recent approach called *SeAK* [60] uses two antennas to develop a secret key, but each device independently develops a key based on the RSSI of exchanged frames. In Wanda, the Wand knows the secret information and imparts it onto the other device without the need for the Wand to develop the same key as the target device.

2.9 User Study

After IRB approval, we conducted a 36-person user study to evaluate the ease of use and effectiveness of our Wanda prototype.

2.9.1 Participants

There were 16 males and 20 females in our subject population, ranging in age from 24 to 76. Subjects were randomly recruited from people passing by in public places such as a library. While we did not screen for it, all subjects had Wi-Fi in their

homes.

Before beginning the experiment, subjects who agreed to participate were asked for demographic information, which consisted of their age, gender, and experience using wireless technology. Subjects rated their experience with wireless devices on a scale of 1 to 5 where 1 indicated limited or no experience with wireless technology and 5 indicated expert or "power user." We later grouped the subjects into categories based on their self-reported experience. Those subjects who rated themselves a 1 or 2 we categorized as *Novices*, those who rated themselves as a 3 were classified as *Intermediates*, and 4 or 5 were *Experts*. Our study population consisted of 12 members from each category for a total of 36 subjects.

2.9.2 Task

After collecting the demographic information, researchers told the subjects that the purpose of the study was to evaluate a new technique for configuring wireless devices. One of the major goals of Wanda is to impart configuration information onto any type of wireless target device in a consistent manner. This means that we cannot assume the target device has any specific kind of input method such as a USB port. To simulate a device with minimum inputs, we simply used a 3 cm x 10 cm x 3 cm cardboard box with a single Alfa Networks AWUS036H Wi-Fi antenna [4] hidden inside. The Alfa Networks antenna was connected to a MacBook Pro running the target device portion of the Wanda software.

The box size was chosen because it was nearly identical to several commercial blood-pressure monitors such as the Fora D30 [35] and because it was large enough so that the data transfer would fail if the user did not move the Wand close enough to the correct location on the target device. The box was adorned with a Wanda logo (see Figure 2.15) affixed near the Alfa Networks antenna to indicate where the user should move the Wand. We envision that future production devices may

Figure 2.15: Target device is a simulated blood-pressure monitor. Note the Wanda logo on the side of the device. Photo by Timothy J. Pierson.

come pre-marked with the antenna location, but even if they are not marked, the Wand can guide the user to the correct location using the technique described in Section 2.3.

Subjects were asked to imagine that the target device was a Wi-Fi-enabled blood-pressure monitor that their doctor had asked them to take home and use everyday. In the imagined scenario, the simulated blood-pressure monitor would transmit blood-pressure readings to the subject's doctor through a Wi-Fi connection. The subject's task was then to configure the simulated blood-pressure monitor to connect to a Wi-Fi AP.

2.9.3 Procedure

After explaining the task, the researcher then showed the subject the prototype Wand and told that them that the Wand could wirelessly impart the network name

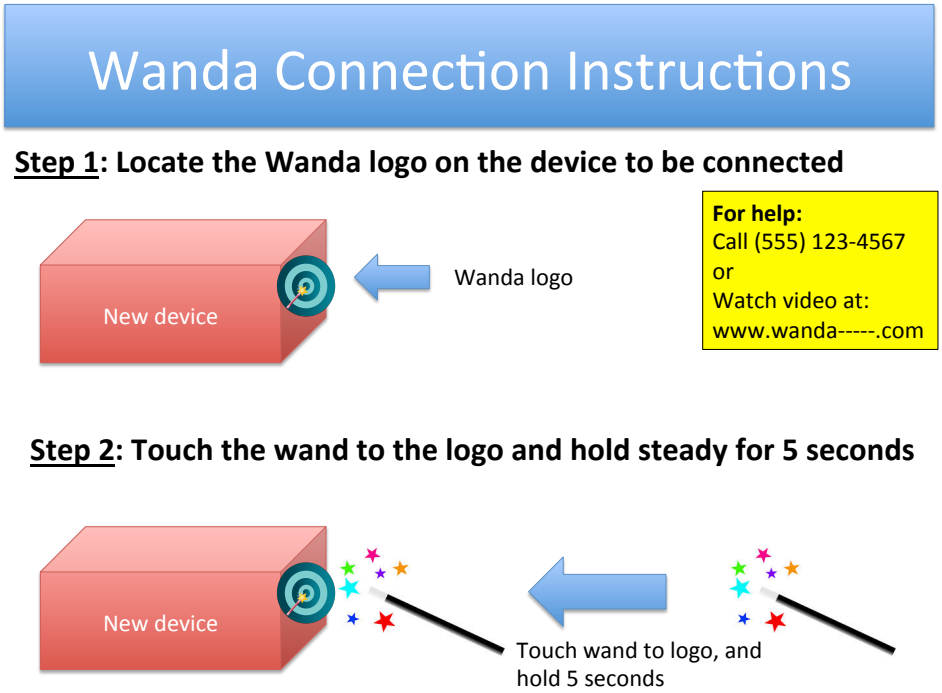

Figure 2.16: Instructions provided to subjects. Note that the instructions included a phone number to call for help and a link to a video, but no subjects elected to use those options.

and password of a Wi-Fi network onto the simulated blood-pressure monitor. If the subject asked how the Wand got the network name and password, the researcher explained that in one arrangement the Wand was envisioned to be part of the AP, connecting to the AP over a wired USB connection so the Wand could get the SSID and password securely over the wire, without fear of a hacker intercepting the credentials.

Once the researcher explained that the purpose of the Wand, subjects were shown the instructions in Figure 2.16. The researcher asked the subject to read the instructions and when they felt like they understood how to use the Wand, to let the researcher know. After showing the instructions to the subject, the researcher refused to answer any questions about how to use the Wand.

The instructions have two steps: (1) Locate the Wanda logo on the device to be configured (e.g., the simulated blood-pressure monitor in this case), and (2) Touch

the Wand to the logo and hold steady for five seconds. The first step was included so the subject would know where to place the Wand so that the Wand could run the *detect* and *impart* primitive operations. The second step of the instructions was provided so that the subject understood they should actually move the Wand in close proximity to the target device.

The subjects were not asked to initiate the Wanda protocol on the target device. To provide security against an adversary who might try to impart his own information onto a target by modulating his transmit power to simulate the Wand's two antennas, Wanda calls for the protocol to begin only after the user takes an action on the target device such as pressing a button. Because the target device could have any number of different interfaces to begin the protocol (e.g., on one type of device the user presses a button, on another type the user selects from a list of options displayed on a screen, on another type the user shakes the device), we elected to forgo this step for consistency.

After the subject indicated they were ready to begin, the researcher told the subject that he would like the subject to try using the Wand five times to configure the simulated blood-pressure monitor. The researcher then placed both the Wand and the simulated blood-pressure monitor on a desk in front of the subject and started the Wanda software running on the Wand and the simulated blood-pressure monitor. The Wand would then play a sound file instructing the subject to, "Move the Wand close to the target device." If the subject moved the Wand close enough for the Wand to detect proximity with the simulated blood-pressure monitor, the Wand played a sound file that said, "Transferring data." If they did not move within close proximity within four seconds, the Wand played a sound file that said, "Touch the Wand to the Wanda logo and hold steady." This repeated every four seconds until the Wand was close or 100 seconds elapsed, in which case the Wand timed out and declared the configuration a failure. If proximity was detected and the

data transfer completed (this took approximately 0.5 seconds), the Wand played a sound file that said either, "Successfully configured device", if the impart succeeded and the target device received all bits correctly, or "Configuration failed, please try again", if the impart failed and the target device did not receive all bits correctly.

A trial succeed if the target was able to decode a SSID of "Linksys" and an eight-character password (128 bits in total). If the simulated blood-pressure monitor did not receive the SSID and password correctly, or if 100 seconds elapsed, the trial was considered a failure. After each trial, the researcher returned the Wand and target device to the desk.

2.9.4 Results

We evaluated Wanda on *reliability, speed,* and *ease of use* for each of the subject categories – Novice, Intermediate, or Expert.

Reliability

To be considered reliable, Wanda must correctly impart data onto the target device with high probably. Each of the 36 study participants used the Wand five times for a total of 180 trials. Of those 180 trials, 174 succeeded for a 96.6% success rate. Figure 2.17 shows the result for each of the five trials for each participant where the participants are sorted by their self-reported familiarity with wireless technology. None of the 12 Experts failed in any of their 60 attempts to configure the target device. Three of the Novices and three of the Intermediates each failed on one of their attempts, although no subject failed more than one time, and five of the six failures were on the first attempt. If we do not consider the subject's first attempt to use the Wand, their success rate improved to 99.4%.

Failures occurred when the subject moved the Wand away from the target

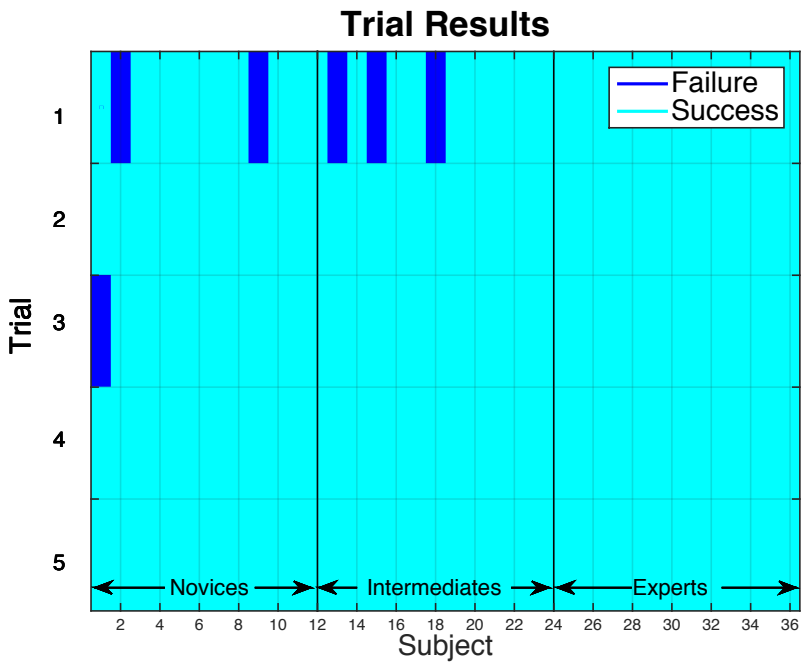

Figure 2.17: Outcome of five trials for each study participant. Success indicates the target device correctly received the SSID and password imparted by the Wand. Failure indicates the target device did not correctly receive the data.

device while the *impart* primitive was running. This movement away made it impossible for the target device to determine which antenna sent the frames at the end of message string m. This is by design to prevent an attacker located more than approximately 30 cm away from learning the data imparted by the Wand. Without prompting from the researcher, each subject correctly used the Wand on subsequent attempts after the initial failure.

Using Wanda, even inexperienced Novices and Intermediates were ultimately able to configure a device with the same level of accuracy as the Experts. To confirm there was no difference between the subjects based on their self-reported familiarity with wireless technology, we performed an ANOVA analysis of the trials. A successful trial was recorded as a 1 and an unsuccessful trial was recorded as a 0. The p-value was 0.215, suggesting no statistical difference between the groups. After eliminating the first trial the p-value climbs to 0.371, further reinforcing the

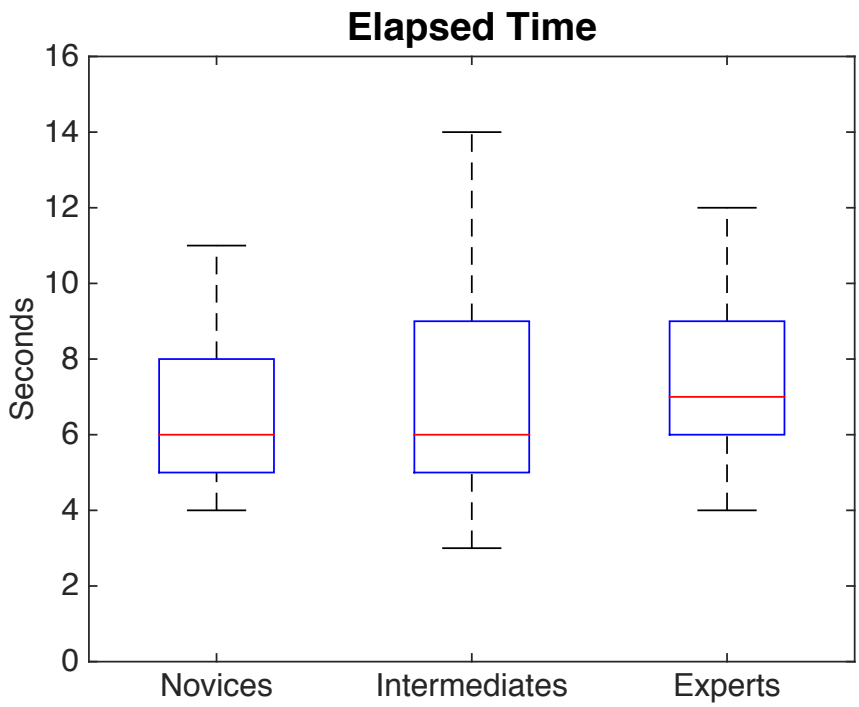

Figure 2.18: Elapsed time. Time in seconds between starting the Wanda protocol and success or failure. There were 60 trials in each group. The box represents the 75th and 25th percentiles, the red line is the median, and the whiskers represent the range of times.

notion that all groups are equally skilled using Wanda.

Given the high probably of successfully configuring the target device, we conclude that **Wanda is reliable**, even for inexperienced users.

Speed

To evaluate the speed of Wanda we measured the elapsed time from when the subject was asked to begin a trial until the time when the target device reported success or failure. The results are shown in Figure 2.18.

We see that the participants were able to complete the task in under 7 seconds on average and all trials took less than 14 seconds. From Figure 2.18 it appears that Novices were on average slightly faster than Intermediates and Experts and had less variability in timing than the other groups. Normally we would expect Novices to take *longer* and have *more* variability configuring a device to connect to a Wi-Fi

AP than Intermediates or Experts. Our goal for Wanda was to make configuring devices as fast for Novices as it is for Experts. To evaluate whether there was any statistical difference between the groups, we performed an ANOVA analysis using the timing data. The p-value between groups was 0.087, suggesting no statistically significant difference between the groups.

For comparison, three expert users followed the directions given in Table 2.3 to configure the Fitbit Aria smart scale and it took them on average over 24 minutes to complete the configuration the first time. The first trial, however, required the users to create accounts and download software. A second trial, using the accounts and downloaded software from the first trial, still took the three experts nearly 6 minutes on average to complete.

We conclude that **Wanda is fast** in comparison with traditional setup application such as the Fitbit Aria and is fast regardless of the subject's familiarity with configuring wireless devices.

Ease of use

We conducted a structured interview after the usage trials to get a deeper understanding of each subject's experience using wireless devices prior to and after the user study. While we did not explicitly screen for it, all subjects had Wi-Fi in their homes and had used it previously. Most of them, 32 out of 36, had previously connected devices to their Wi-Fi AP, but four had not. Interestingly, 20 of the 32 subjects who attempted to connect a device found it too difficult to do by themselves and had to call someone for help. Table 2.4 shows who was called.

From Table 2.4 we see that family members are most commonly called upon for help, but corporate IT departments are also sharing a good deal of the burden for setting up personal devices for our study subjects. Other IT support such as the subject's cable TV provider also fielded a number of help requests. Device

Number of calls	*Entity called*
6	Spouse/relative
5	IT support (work)
4	IT support (other)
3	Device manufacturer
2	Friend

Table 2.4: Calls for help. 20 of 32 subjects who attempted to configure a wireless device without using Wanda needed to call for help. This table shows the number of calls to help entities.

manufacturers (e.g., Cisco) and non-family friends were called less frequently. While none of our study participants reported using a professional installer, clearly the group was reliant on others for help.

One of the major problems many of the subjects reported was remembering their Wi-Fi AP's network name and password: 18 said they knew their network's name and password, 13 said they did not know the password because it was complex (and written down somewhere), and 5 (about 14%) did not know the password at all and did not know how to find it. With Wanda the user does not need to know the network credentials – the Wand learns them over a wire and then imparts them onto devices. Because the user does not need to memorize (or write down) the network credentials, stronger, more complex passwords can be used with the same ease as simple, easy to crack passwords. Wanda could be extended to allow each device have its own unique password to access the network such as in 802.1x networks typically found in corporate environments and managed by technology professionals, or in proposed research initiatives such as MultiNet [16].

We asked the subjects to rate how confident they are that they could configure a wireless device before and after using Wanda. Before using Wanda, the subjects rated themselves an average 3.4 on a scale from 1 to 5 where 1 indicated they had no confidence they could configure a device and 5 indicated they were very confident they could configure a device. After using Wanda, all subjects except one reported

they were a 5 on the same scale. The subject who did not report a 5 said he was 4.5 due to one of the trials failing. ANOVA analysis yielded a p-value of 1.4×10^{-10}, suggesting it is extremely unlikely the subject's confidence levels did not improve.

The subjects generally enjoyed using Wanda. We asked them for general comments on Wanda's approach versus methods they had previously used to configure devices and their comments are listed in Appendix A.2. While most of the comments are extremely positive, subject number 12 was concerned that this approach would be less safe than entering her password herself, noting that she had previous experienced identity theft. None of the other participants expressed security concerns.

Finally, the subjects all understood how to use the Wand simply by reading the instructions in Figure 2.16. The instructions contained a phone number to call for help (which rang at the researcher's desk) and link to a video. None of the subjects elected to call the help line or watch the video.

Based on these results, we conclude **Wanda is easy to use**.

3

Physical Layer Concepts

So far in this thesis we have focused on RSSI as the primary characteristic of a radio signal. RSSI, however, is a relatively coarse description of a signal, providing limited information. Much of the cutting-edge research on radio communications these days uses more detailed information about the physical layer (sometimes referred to as PHY layer). The terms used in the physical-layer literature, however, can be confusing for researchers accustomed to working at layers above the physical layer. In this section we briefly review some of the basic physical-layer concepts that we build upon in following chapters. Readers familiar with PHY-layer concepts can safely skip this chapter.

We begin with a discussion of how complex numbers provide a convenient way to mathematically describe and manipulate radio signals. Next we review how real hardware uses complex numbers to create sophisticated modulation schemes such as Quadrature Amplitude Modulation (QAM). Because our focus in is primarily oriented around Wi-Fi, we discuss how Wi-Fi uses these modulation schemes together with Orthogonal Frequency Division Modulation (OFDM) to create fast and reliable wireless data transfer. Finally, we revisit how a signal changes as it propagates through space while traveling from transmitter to receiver and how those changes can be estimated by a receiver using Channel State Information (CSI).

3.1 Complex number review

Complex numbers are a convenient way for a single quantity to represent vectors in a two-dimensional space. Complex numbers are the sum of two components: a real component and an imaginary component. The real component is simply an ordinary number, and the imaginary component is also an ordinary number but is multiplied by $\sqrt{-1}$. Complex numbers are represented in rectangular form as $a + bj$, where a is the real component, b is the imaginary component, and j is $\sqrt{-1}$. Mathematicians often use i to represent $\sqrt{-1}$, while engineers typically use j. In this thesis we use j.

Complex numbers can be viewed in the *complex plane* where the horizontal axis maps the real component (often denoted as Re, or I for the "in-phase" component described in Section 3.3.1) and the vertical axis maps the imaginary component (often denoted as Im, or Q for the "quadrature" component described in Section 3.3.1) of a complex number as shown in Figure 3.1.

A complex number can be separated into its real and imaginary components using two operators: $\Re(x)$ returns the real component, and $\Im(x)$ returns the imagi-

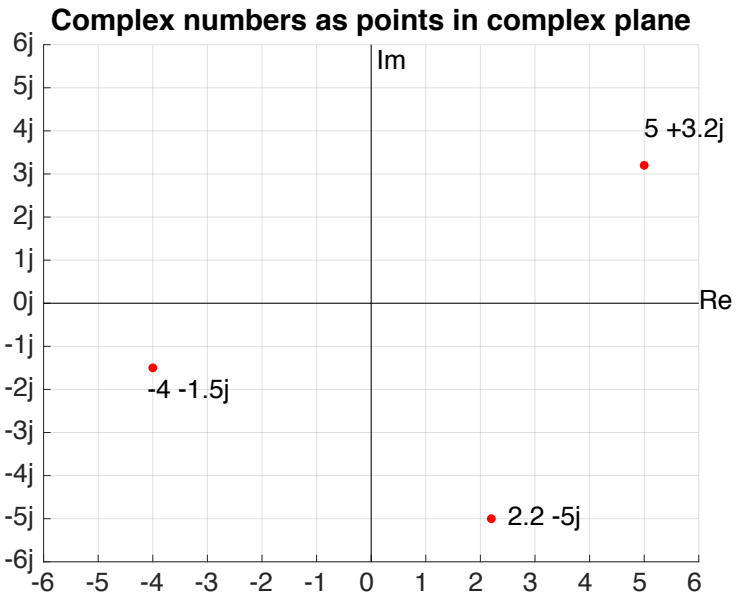

Figure 3.1: Numbers plotted in the complex plane. Several complex numbers in the form $a + bj$ are plotted in complex plane.

nary component of complex number x without the j. For example, if $x = 5 + 3.2j$, then $\Re(x) = 5$ and $\Im(x) = 3.2$.

In addition to the rectangular notation described above, complex numbers can also be described in polar notation. In polar notation, points in the complex plane are described in terms of phase angle θ and a magnitude M instead of coordinates a and b. Magnitude M represents the distance of a point from the origin, while phase θ is an arc from the positive real axis in a counterclockwise direction as shown in Figure 3.2.

Because rectangular notation and polar notation both describe the same point, we can convert between them. Equation (3.1) converts from rectangular to polar notation and Equation (3.2) converts from polar to rectangular notation.

From rectangular to polar:

$$M = \sqrt{(\Re(x)^2 + \Im(x)^2} \tag{3.1}$$

$$\theta = arctan[\Im(x)/\Re(x)].$$

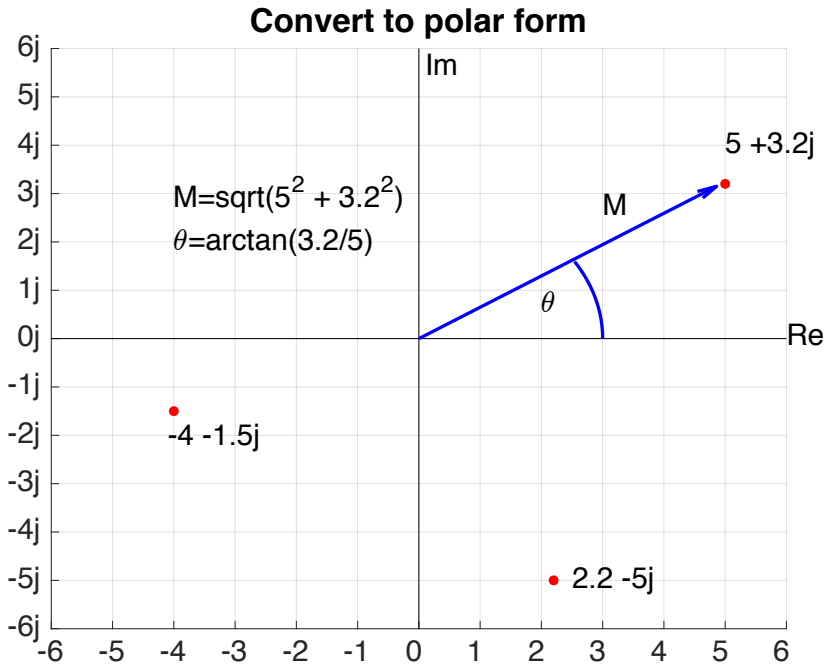

Figure 3.2: Polar form. Converting rectangular point 5 + 3.2j to polar notation with magnitude $M = \sqrt{5^2 + 3.2^2}$ and phase $\theta = arctan(3.2/5)$.

From polar to rectangular:

$$\Re(x) = M\cos(\theta) \tag{3.2}$$

$$\Im(x) = M\sin(\theta).$$

Applying Equation (3.1) to the point $x = 5 + 3.2j$ yields polar coordinates $M = \sqrt{5^2 + 3.2^2}$ and $\theta = arctan(3.2/5)$. The key idea is that the same point can be represented in either rectangular or polar form in the complex plane. In rectangular form information is carried in variables a and b, but the entire complex number is the expression $a + bj$. Similarly, in polar notation the information is contained in M and θ, but the entire complex number is the expression $M(\cos(\theta) + j\sin(\theta))$.

The fact that the same point can be represented in two ways yields the following equality:

$$a + bj = M(\cos(\theta) + j\sin(\theta)). \tag{3.3}$$

Additionally, we know from Euler that [115]:

$$e^{jx} = \cos(x) + j\sin(x). \tag{3.4}$$

Substituting the Euler's formula in Equation (3.4) into Equation (3.3), we see that:

$$a + bj = Me^{j\theta}. \tag{3.5}$$

Describing a point in the form $Me^{j\theta}$ is especially useful for describing a radio signal. As discussed below, we often use two features to describe a signal: M to denote the magnitude (amplitude) of a signal, and θ to describe its phase. Complex numbers provide a compact representation of both aspects the signal in a single value and a convenient way to mathematically manipulate a signal.

3.2 Representing a sinusoid as a complex number

Communication systems often use a high-frequency carrier signal (sometimes called the *bandpass* signal) to "carry" a lower frequency information signal (sometimes called the *baseband* signal). In this case the carrier frequency is modulated (changed) according to the lower frequency information signal. Except for extremely basic modulation schemes such as on/off keying, the carrier signal itself provides no information – the information is encoded by modulating the carrier signal in some manner. If the modulation is done in a predetermined fashion, a receiver can demodulate (recover) the information signal encoded onto the carrier by a transmitter.

As suggested above, a radio signal can be described in terms of two values: magnitude and phase. The following formula is the standard way to represent a

sinusoid radio signal and uses magnitude and phase to describe the signal over time in polar notation [127]:

$$V(t) = M\cos(2\pi f_c t + \phi) \tag{3.6}$$

where $V(t)$ is the voltage of a signal at time t, M is the magnitude or amplitude of the signal, f_c is the frequency of the carrier signal,[1] and ϕ is the phase offset of the signal.

Given magnitude and phase information from Equation (3.6), the state of the signal in the complex plane at an instantaneous point in time can be plotted as discussed above. Furthermore, coordinates of the signal in rectangular form can be computed using Equation (3.2). Figure 3.3 shows the conversion to rectangular coordinates for a point with magnitude M and phase θ. Basic trigonometry confirms the rectangular coordinates of the point are (I, Q), where $I = M\cos(\phi)$ and $Q = M\sin(\phi)$.

While somewhat counterintuitive, describing the signal in terms of point (I, Q) provides an easy way to modulate a signal. Next we discuss how actual hardware uses these coordinates to modulate a carrier frequency's phase and amplitude, imparting the information signal onto the carrier.

3.3 Modulating a sinusoid signal with actual hardware

An unmodulated sinusoid carries no information; for a signal to convey information, it will need to be modulated in one or more ways. Recall from Equation (3.6) that a sinusoid can be described as $M\cos(2\pi f_c t + \phi)$. This equation suggests three possibilities to impart information onto the carrier signal – vary amplitude M by

[1] $2\pi f_c$ is sometimes called the *natural frequency* often denoted as ω, and represents the rate of change of the signal in radians per second.

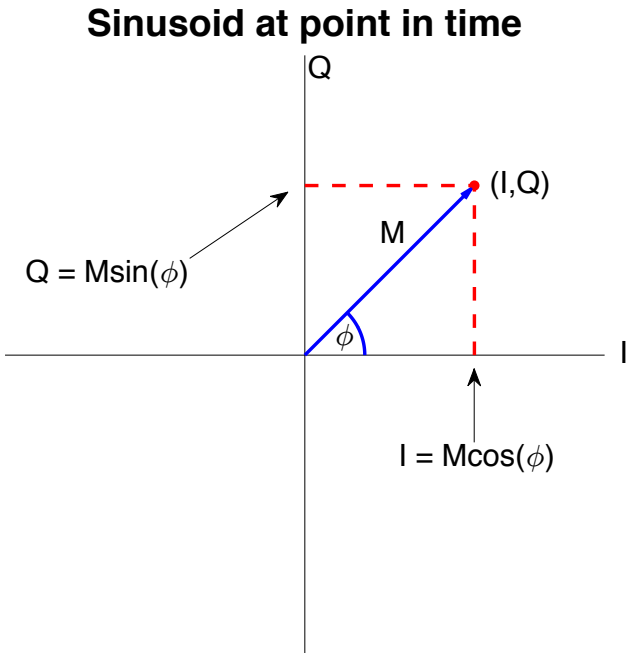

Figure 3.3: Plotting phase and magnitude with I and Q. Given phase and magnitude information, the state of a signal can be plotted in the complex plane in rectangular form as a point (I, Q).

changing the transmit power, vary frequency f_c by changing the cycles per second of the signal, or vary phase by changing the phase offset ϕ of the signal. Sometimes more than one of these factors are modulated simultaneously.

Mathematically, these factors can be modeled by converting the values that were previously constants in Equation (3.6) into time-varying vectors:

- $M(t)$ for amplitude
- $f(t)$ for frequency
- $\phi(t)$ for phase.

In this section we examine how the magnitude and phase of a carrier frequency can easily be modulated by changing I and Q.

3.3.1 I and Q

While there are three possibilities for modulating a signal, it is difficult with physical hardware to precisely modulate the phase of signal in accordance with an input signal. We can, however, accomplish the same thing and avoid directly manipulating the carrier's phase by modulating the I and Q values discussed above. To understand why, we turn to the following trigonometric identity [36]:

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta). \tag{3.7}$$

Applying this trigonometric identity to Equation (3.6), where a signal is described by $M\cos(2\pi f_c t + \phi)$, multiplying Equation (3.7) by M, and substituting $2\pi f_c t$ for α and ϕ for β yields:

$$M\cos(2\pi f_c t + \phi) = M\cos(2\pi f_c t)\cos(\phi) - M\sin(2\pi f_c t)\sin(\phi). \tag{3.8}$$

We saw from Equation (3.2) and Figure 3.3 that $I = M\cos(\phi)$ and $Q = M\sin(\phi)$. Substituting I and Q into Equation (3.8) yields:

$$M\cos(2\pi f_c t + \phi) = I\cos(2\pi f_c t) - Q\sin(2\pi f_c t). \tag{3.9}$$

Equation (3.9) shows that by changing I and Q, we can steer the coordinates of the modulated signal (e.g., the left side of Equation (3.9)) in the complex plane to any desired location. Then by using Equation (3.1) we can convert back to polar notation, yielding the new magnitude and phase of the modulated signal.

Furthermore, building on Equation (3.9), we see that it is possible to modulate a single carrier frequency with both the I and Q components. A sine wave and a cosine wave of the same frequency have a 90-degree ($\pi/2$ radians) phase offset

between them (e.g., $\cos(2\pi f_c t - \pi/2) = \sin(2\pi f_c t)$). Signals with a 90-degree offset are said to be in *quadrature* with each other. By convention, the cosine wave in Equation (3.9) is said to be the *in-phase* or I component of the signal, while the sine wave is said to be the *quadrature* or Q component. We can leverage the quadrature nature of the sine and cosine wave in Equation 3.9 by substituting a cosine wave for the sine wave as follows:

$$M\cos(2\pi f_c t + \phi) = I\cos(2\pi f_c t) - Q\cos(2\pi f_c t - \pi/2). \tag{3.10}$$

With Equation 3.10 we see that a single input frequency with no phase shift can be modulated by the I component (the in-phase component) and a 90-degree phase shifted version of the same input frequency can be modulated by the Q component (the quadrature component). Because of its quadrature nature, this modulation technique is called *Quadrature Amplitude Modulation* (QAM).

Figure 3.4 shows a simple hardware diagram of a transmitter that implements Equation (3.10), illustrating how a single carrier frequency can be modulated by both I and Q. In this diagram the circles with an 'X' represent mixers – devices that perform frequency multiplication and either upconvert from the baseband signal to the carrier frequency or downconvert from the carrier frequency to the baseband signal [81]. In Figure 3.4 we see the I/Q modulator mixing the I signal with the unaltered RF carrier wave, and mixing the Q signal with a quadrature (90-degree offset) version of the same RF carrier. The Q signal is subtracted from the I signal as in Equation (3.10) to produce the final RF modulated waveform.

At the receiver the process is reversed. The incoming RF signal is mixed with the output of a local oscillator with a zero phase shift to recover the I component of the signal, and the RF is separately mixed with a 90-degree offset of the local oscillator output to recover the Q portion. We examine the receiver closely in

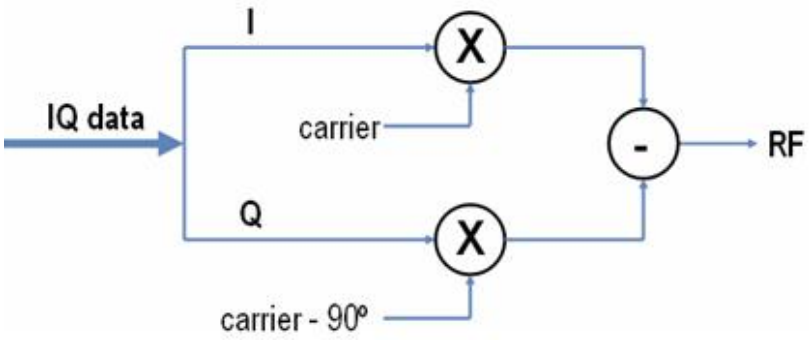

Figure 3.4: Simplified hardware diagram of an I/Q modulator [81].

Chapter 4.

In summary, Quadrature Amplitude Modulation modulates an in-phase version of a RF carrier with the I information signal and a 90-degree phase shifted version of the same RF carrier with the Q information signal. This allows QAM to shift the output signal to any (I, Q) point, and thus any desired magnitude and phase.

3.4 BPSK, QPSK, and QAM modulation

To illustrate how I and Q can be used to implement real-world modulation schemes, consider the case where I and Q can take the value of 0 or 1, that is: $I, Q \in \{0, 1\}$. In this case, if $I = 1$ and $Q = 0$, then by Equation (3.10) the output is simply the cosine wave (phase equal to 0). If $I = 0$ and $Q = 1$, then the output is a sine wave (phase shifted 90 degrees). If both I and Q are 1, then the output is the combination of both phase shifts (0 and 90 degrees), resulting in a new signal with a phase shift of 45 degrees.

3.4.1 BPSK

Binary Phase Shift Keying (BPSK) exploits this concept. In BPSK, Q is kept constant at 0, and I is modulated as either +1 or -1. This creates a phase shift when I

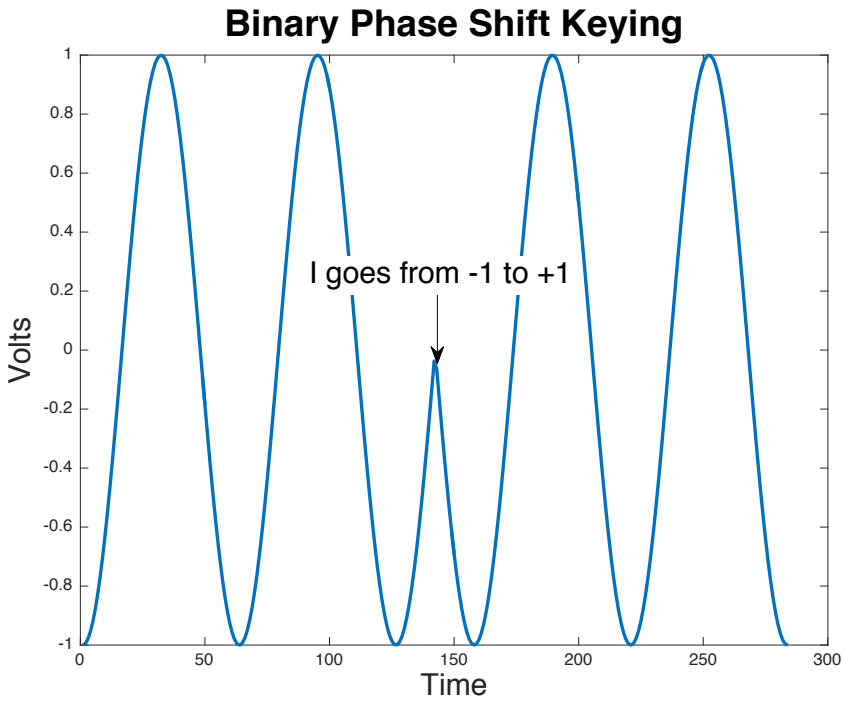

Figure 3.5: Binary Phase Shift Keying (BPSK). BPSK keeps Q constant and modulates I, producing a change in phase when the value of I changes [124].

modulates as shown in Figure 3.5. The resulting signal has either a phase of 0 degrees or a phase of 180 degrees, depending on whether I was +1 or -1. Figure 3.6 shows the results graphically in a plot called a *constellation diagram*. Table 3.1 shows the resulting phase from $I \in \{-1, +1\}$ and $Q = 0$. The receiver can use the signal's phase to decode a bit as a 1 if the phase shift is 0 and to decode a bit as 0 if the phase shift is 180 degrees.

I	Q	*Phase*
+1	0	0
-1	0	180

Table 3.1: BPSK phases in degrees. In BPSK $I \in \{+1, -1\}$ and $Q = 0$, resulting in two possible phases.

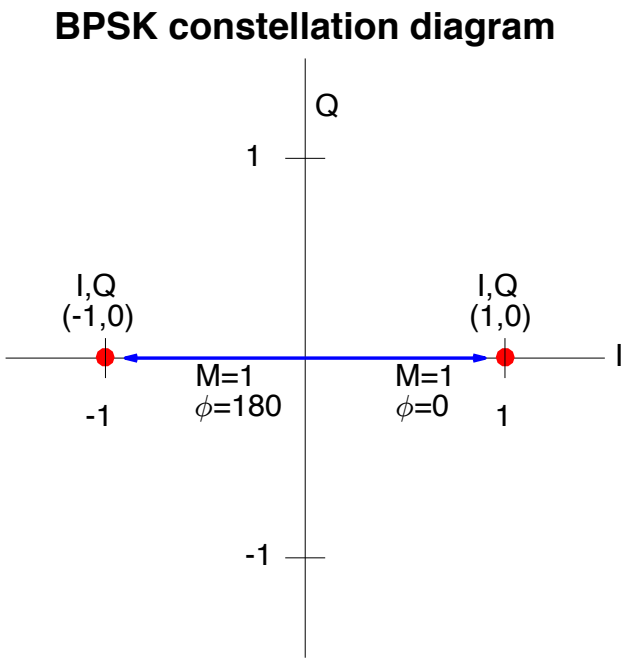

Figure 3.6: BPSK constellation diagram showing the two possible phases where $I \in \{-1, +1\}$ and $Q = 0$.

3.4.2 QPSK

Quadrature Phase Shift Keying (QPSK, sometimes referred to as 4-QAM) builds on BPSK by allowing both I and Q to take values of either +1 or -1. Table 3.2 shows the phase values the signal can take on and Figure 3.7 shows the resulting constellation diagram. Given that values of I and Q are either +1 or -1, there are four distinct values each transmission can assume. Each distinct value is called a *symbol* and each symbol can represent two bits in QPSK. Because each symbol can represent two bits, QPSK can transmit data twice as fast as BPSK.

I	Q	*Phase*
+1	+1	45
-1	+1	135
-1	-1	225
+1	-1	315

Table 3.2: [QPSK phases in degrees. In QPSK $I, Q \in \{+1, -1\}$, resulting in four possible phases.

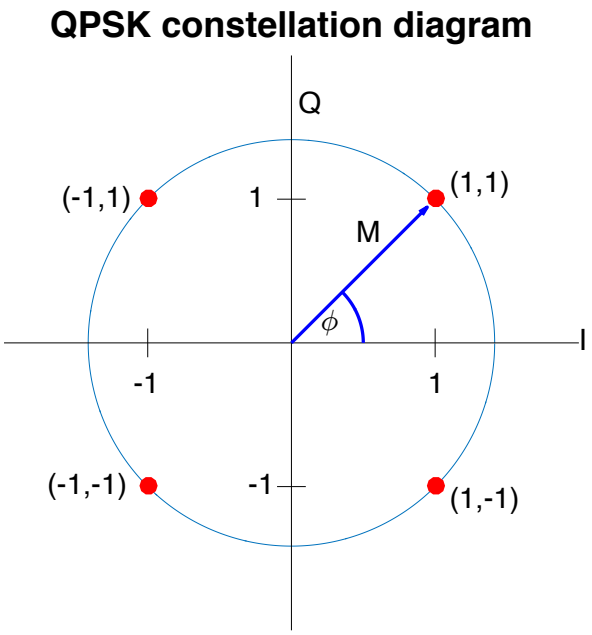

Figure 3.7: QPSK constellation diagram showing the four possible phases where $I, Q \in \{-1, +1\}$.

3.4.3 QAM

More complex QAM modulations simply have more states that I and Q can take on. For example, in 16-QAM, I and Q can each take on one of four discrete values {-3, -1, +1, +3}, resulting in 16 possible combinations of amplitude and phase. These 16 positions effectively allow 16-QAM to transmit 4 bits per symbol as shown in the constellation diagram in Figure 3.8. This allows 16-QAM to transmit data at twice the rate of QPSK and four times the rate of BPSK.

Other modulations schemes allow for more values for I and Q – 64-QAM has 64 symbols and transmits 6 bits per symbol, and 256-QAM has 256 symbols, transmitting 8 bits per symbol. Also, because I and Q can assume the same values (e.g., either +1 or -1 in QPSK, or one of four values in 16-QAM, and so on), the resulting constellation diagrams are square.

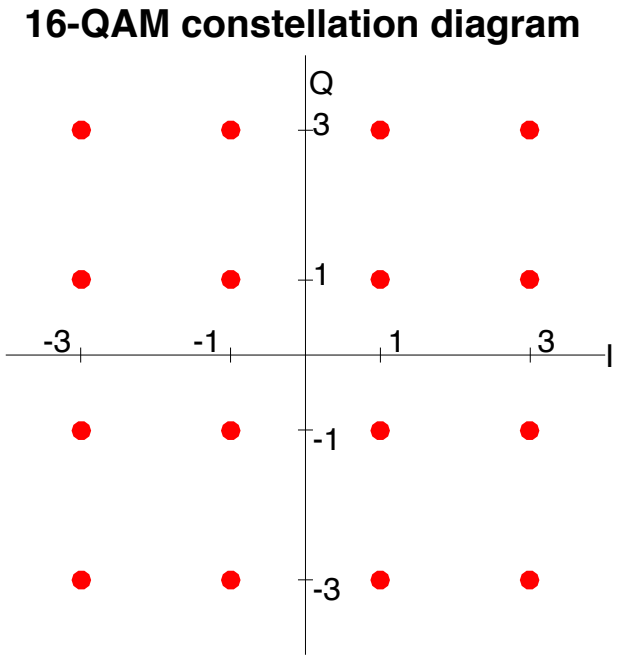

Figure 3.8: 16-QAM constellation diagram showing the 16 possible phases and amplitudes where $I, Q \in \{-3, -1, +1, +3\}$. Because I and Q can each take on one of four different values, there are 16 possible combinations and each symbol in 16-QAM represents four bits.

3.4.4 Demodulation

When a receiver demodulates a symbol, it effectively plots the position of the symbol's I and Q components on a constellation diagram and selects the closest reference point on the constellation diagram as the most probable value the sender transmitted. Noise on the channel, however, can cause alter the demodulated I and Q values and cause the receiver to misinterpret the sender's symbol. As the modulation scheme increases in complexity, the number of values I and Q can take on increases, increasing the chance for error. In practice, complex modulation schemes are chosen when the channel is relatively noise-free to increase throughput, and simpler schemes are chosen in noisy conditions to increase reliability.

We examine demodulation closely in Chapter 4 where we build our own Wi-Fi receiver using a Software Defined Radio (SDR).

3.5 Wi-Fi OFDM

So far we have discussed modulating signals in general. In this section we focus on how Wi-Fi in particular uses the modulation techniques described above, together with *Orthogonal Frequency Division Multiplexing* (OFDM), to provide fast and reliable information transfer. At the physical layer, Wi-Fi versions 802.11a/g/n/ac use OFDM to send portions of a data over several orthogonal (non-interfering) frequencies simultaneously.[2] Wi-Fi's implementation of OFDM creates 14 channels in the 2.4 GHz band and 24 channels in the 5 GHz band.[3] Each channel is 20 MHz wide[4] and is divided into 64 subcarriers that are separated by 312.5 KHz, although only 48 of the 64 subcarriers are used to carry data in 802.11a/g (the other subcarriers are pilot and guard subcarriers) and 52 carry data in 802.11n/ac. Each subcarrier can be thought of as its own narrowband data channel and is modulated independently of the other subcarriers. This means that Wi-Fi can send 48 or 52 symbols in parallel with each symbol modulated with either BPSK, QPSK, 16-QAM, 64-QAM, or 256-QAM, and with each symbol on its own subcarrier. Collectively this collection of data symbols is called an *OFDM symbol* and lasts 4 microseconds according to the Wi-Fi specification [57]. The choice of modulation scheme depends on the hardware capabilities of the transmitter and receiver, and the noisiness of the channel. As noted above, the more complex modulation schemes allow more bits per symbol, which equates to faster data throughput, but can lead to more errors when the Signal-to-Noise Ratio (SNR) is low.

In addition to providing faster throughput by parallel transmission of data on

[2]The older 802.11b uses Direct Sequence Spread Spectrum (DSSS) instead of OFDM. In this thesis we focus on OFDM techniques.

[3]802.11a only uses the 5 GHz band, 802.11g uses the 2.4 GHz band, 802.11n/ac can use either the 2.4 GHz or the 5 GHz band. Some countries restrict usage of some channels due to concerns of interference with other systems such as weather radar.

[4]802.11n/ac can use wider channels, but in this thesis we focus on standard 20 MHz wide channels.

separate subcarriers, the multiple OFDM frequencies help combat the fast-fading effects of *multipath* described in Section 2.2. Under multipath conditions, a signal reflects from obstacles in the environment and multiple copies of the signal arrive at the receiver with different delays, amplitudes, and phases. Sometimes the multiple copies of the signal combine destructively. Small changes in the position of the transmitter or receiver, or other moving objects in the environment, can significantly alter the received strength of a signal, sometimes severely fading a signal that had been strong only a short while before. If a signal is highly faded, it is likely the data transmitted will be received with errors.

OFDM attempts to avoid errors due to fading while still maintaining high reliability by sending redundant copies of the data on different frequencies simultaneously. Because it is less likely that *all* frequencies will be highly faded at the same time compared with sending data over only one frequency, OFDM's frequency multiplexing tries to ensure reliable data transfer. By using multiple frequencies and sending data in parallel, OFDM tries to maintain high speed.

Wi-Fi attempts to combat fading by sending duplicate bits on different subcarriers. The process is shown in Figure 3.9. In the first part of the process data bits are sent through a convolution encoder to introduce redundancy. For example, a 1/2 convolution code provides two output bits for every input bit. The resulting bits can then optionally be "punctured" to remove selected bits, reducing the level of redundancy, but increasing throughput. The resulting bits are then interleaved to ensure redundant bits are sent on different subcarriers.

The resulting bits are then fed into a Modulator as shown in Figure 3.10 [38]. The Modulator groups bits into symbols according to the desired modulation, BPSK, QPSK, or QAM, as described in Section 3.4. Each symbol is comprised of a distinct phase and amplitude and depending on the modulation scheme chosen, may represent multiple bits. The output of the Modulator is a stream of

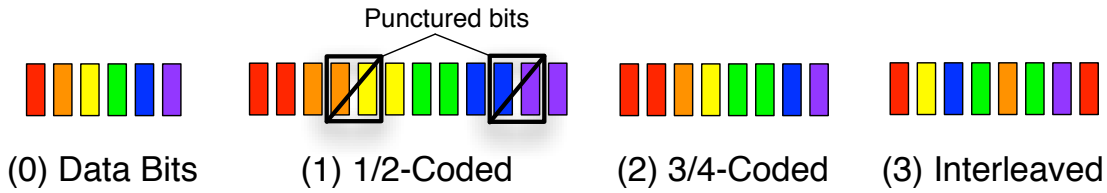

Figure 3.9: OFDM encoding process. Data bits (0) are duplicated to provide redundancy and protection against fading, but reducing throughput, and then optionally punctured (1) by dropping certain bits for reduced redundancy but higher throughput (coding rate here is 3/4). The remaining bits (2) are interleaved (3) to spread the redundancy across subcarriers and protect against frequency-selective fades. Reproduced from Halperin et al. [51].

complex-valued symbols $X[0] \ldots X[N-1]$ representing these distinct phase and amplitude values. Stream $X[0] \ldots X[N-1]$ is then converted to parallel by the Serial-To-Parallel Converter with each symbol matched to one of the N subcarrier frequencies. This means subcarrier i will be modulated with the phase and amplitude specified by $X[i]$, where $i = 0 \ldots N-1$. Collectively the output of the Serial-to-Parallel Converter is the frequency-domain representation of an OFDM symbol. The frequency-domain representation is converted to the time domain with an Inverse Discrete Fourier Transform (IDFT), most commonly implemented with an Inverse Fast Fourier Transform (IFFT). The time-domain output of the IFFT, $x[n]$, is given by [38]:

$$x[n] = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} X[i] e^{j2\pi ni/N} \qquad 0 \leq n \leq N-1. \tag{3.11}$$

Next a *Cyclic Prefix (CP)* consisting of the last 16 samples in x is prepended to x. The CP is not used by the receiver, but is included to help prevent inter-symbol interference (ISI). The time-domain signal $x[n]$ is then converted to back to serial by the Parallel-to-Serial Converter and is converted from digital to analog by the Digital to Analog (D/A) converter, resulting in $x(t)$. Finally, $x(t)$ is upconverted with the carrier wave of frequency f_0 and transmitted as $s(t)$.

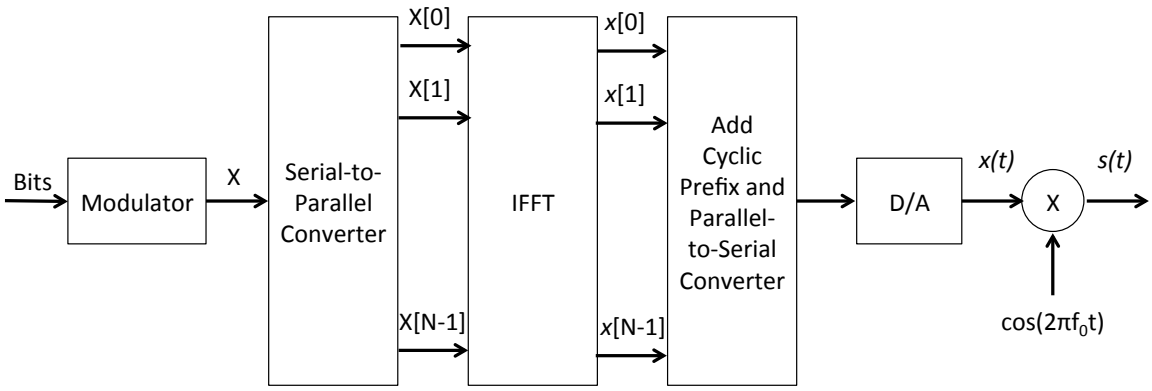

Figure 3.10: OFDM transmit process. Data bits from Figure 3.9 are fed into a Modulator which groups bits into complex symbols X[0] ... X[N-1], one symbol per subcarrier. These symbols are converted from serial to parallel and the resulting frequency-domain signal is converted into the time domain by an IFFT operation. Next a Cyclic Prefix is added, and the parallel time-domain signal is converted back to serial. The serial time-domain signal is converted to analog and finally upconverted with a carrier wave of frequency f_0 resulting in transmitted signal $s(t)$. Reproduced from Goldsmith [38].

The Wi-Fi receiver then reverses these steps to recover the 48 or 52 transmitted symbols. We discuss this process in detail in Chapter 4.

3.6 SISO, SIMO, MISO, and MIMO

So far we have implicitly assumed that there is one transmit antenna and one receive antenna. This configuration is often called *Single Input Single Output* (SISO). Modern systems, however, frequently use multiple antennas to transmit and receive signals. Wanda, for instance, uses two antennas on the Wand to receive signals from the single-antenna target device to determine proximity in the *detect* portion of the Wanda protocols. In this case the arrangement is called *Single Input Multiple Output* (SIMO) because the signal transmitted by the single-antenna target is picked up by multiple (in this case two) antennas on the Wand. During the *impart* phase of the Wanda protocols, Wanda switches its method of operation and the Wand uses its two antennas to transmit signals to the single-antenna target device. This type of configuration where multiple transmit antennas send a signal to a single receiver

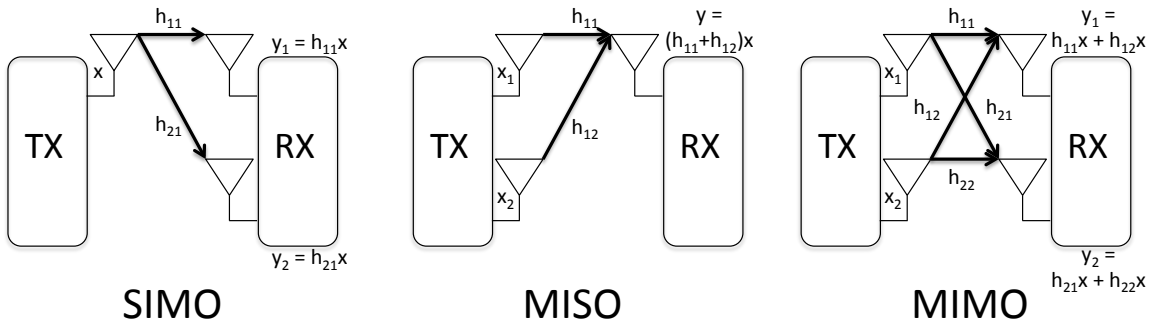

Figure 3.11: SISO, SIMO, MISO and MIMO. Various antenna configurations with channel state represented by h_{rt} for the path between transmit antenna t and receive antenna r.

is called *Multiple Input Single Output* (MISO). Finally, some systems have multiple antennas on both the transmitter and receiver. These systems are called *Multiple Input Multiple Output* (MIMO). SIMO, MISO, and MIMO configurations are shown in Figure 3.11.

3.7 Channel State Information

Now that we have reviewed how to represent, transmit, and receive signals, we turn our attention to what happens to the signal while in transit between the transmitter and receiver. As noted above, a signal can be changed at the receiver by multipath effects and moving obstacles. Each antenna on a receiver acquires a copy of the transmitted signal modified by the channel between the transmitter and itself. The state of the channel (e.g., the nature of the changes the channel makes to the transmitted signal) may change quickly, but can be estimated on the receiver (and in some cases that information can be fed back to the transmitter to allow the transmitter to alter the signal before transmission to purposefully cause the multipath signals to add up constructively at the receiver). These estimates of the channel's state are called *Channel State Information* (CSI). The drivers on some commercial NICs can be modified by software tools to provide their estimate of the

channel to user applications [52].

In Wi-Fi, the channel state for all symbols in a frame can be estimated during the frame's preamble. The preamble includes a number of "training" bits that precede the data in the frame and are sent at a pre-determined amplitude and phase. The receiver can recognize the preamble bits and compare the magnitude and phase of the bits it receives with the pre-determined magnitude and phase of the training bits. This allows the receiver to estimate the changes to the signal, in both amplitude and phase, caused by the channel. CSI tools can provide their estimate of the channel state for each data subcarrier in an OFDM channel. Because the channel is estimated during a frame's preamble, there is only one CSI measurement for each Wi-Fi frame, even though a Wi-Fi frame may contain many symbols.

Radio transmissions are often modeled according to the following time-invariant formula [127]:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{3.12}$$

where $\mathbf{y}$ is the symbol received at a receiver, represented by a complex number comprised of the magnitude and phase of the received signal, $\mathbf{H}$ is a matrix of complex numbers representing CSI, the changes to the signal caused by the state of the channel, $\mathbf{x}$ is the transmitted symbol, represented as a complex number of phase and magnitude, and $\mathbf{w}$ is white Gaussian noise. If there are n_r receive antennas and n_t transmit antennas, then $\mathbf{H}$ is a $n_r \times n_t$ matrix of complex numbers: $\mathbf{H} \in \mathbb{C}^{n_r \times n_t}$.

It is important to note that the magnitude and phases in complex numbers contained in matrix $\mathbf{H}$ represent the *changes* to signal as it travels through the air. They *do not* represent the magnitude and phases imparted on the carrier frequency during modulation to form symbols. The symbol modulations are represented in vector $\mathbf{x}$. Vector $\mathbf{y}$ represents symbol $\mathbf{x}$ transformed by the channel and noise.

The line-of-sight channel between transmit antenna t and receive antenna r as

shown in Figure 3.11 is denoted as a complex number h_{rt} [127]:

$$h_{rt} = a_{rt} e^{-j2\pi f \tau_{rt}} \tag{3.13}$$

where a_{rt} is the attenuation of the signal on the path between transmit antenna t and receive antenna r, f is the signal frequency, and τ_{rt} is the time of flight between transmit antenna t and receive antenna r.

Each of the different paths from transmitter to receiver will impart a different attenuation, delay, and phase shift on the signal and can be modeled in H as follows [127]:

$$h_{rt} = \sum_{p=1}^{P} a_p e^{-j2\pi f \tau_p} \tag{3.14}$$

where P is the total number of paths from transmitter to receiver, including the line-of-sight and all multipath routes, a_p represents the attenuation of the signal on path p, f is the signal's frequency, and τ_p is the propagation delay (i.e., time-of-flight) for the signal on path p.

We can also formulate h in terms of the distances and wavelength of the signal. We know from physics that distance d is equal to rate times time, and that a radio signal travels at the speed of light c, so distance is equal to $c \times \tau$. We also know that the frequency f of a signal is equal to c/λ, where λ is the wavelength of the frequency. We can substitute into Equation (3.14) to formulate the channel in terms

of distance and signal wavelength:

Given:

$$d_p = c \times \tau_p$$
$$f = \frac{c}{\lambda}$$

Then:

$$\tau_p = \frac{d_p}{c} \tag{3.15}$$

$$2\pi f \tau_p = 2\pi(\frac{c}{\lambda})(\frac{d_p}{c})$$
$$= 2\pi d_p/\lambda$$

$$h_{rt} = \sum_{p=1}^{P} a_p e^{-j2\pi d_p/\lambda}.$$

A CSI tool provided by Halperin [52] that works on the Intel 5300 Wi-Fi card [58] provides estimates of **H** as a $n_r \times n_t \times 30$ matrix of complex numbers where each entry in **H** is given by Equation (3.15) in the form $a + bj$. Here n_r is the number of receive antennas, n_t is the number of transmit antennas, and 30 is the number of subcarriers estimated. Equation (3.1) converts from rectangular form to polar form to produce magnitude $M = \sqrt{a^2 + b^2}$ and phase $\theta = arctan(b/a)$ for each transmit and receive antenna pairs on the 30 different subcarriers.

4

SNAP: SiNgle Antenna Proximity

4.1 Introduction

We saw in Chapter 2 that Wanda is a practical way for a two-antenna Wand to securely impart data onto a nearby target device. The two-antenna Wand is able to determine proximity with a nearby target device by examining the RSSI of a signal transmitted by the target. If the RSSI received on the Wand's antenna A_1 is sufficiently stronger than the signal received on antenna A_2, then the Wand declares proximity. The single-antenna target device, however, cannot use the two-antenna method and has no way of verifying its proximity to the Wand. Without a means of

separating nearby devices from distant devices, the single-antenna target device could potentially be tricked into accepting data from a distant adversary that is modulating its transmit power as discussed in Section 2.7.

To protect against a distant adversary attempting to impart data onto the target device, in Chapter 2 we suggest that a human must intentionally initiate the Wanda protocol by taking action such as pressing a button on the target device. The Wand then listens for rogue Wanda frames. If rogue frames are detected, the Wand sends a *Stop* message to the target device to it inform of the presence of the adversary. Ideally, however, the target device would have its own way to recognize frames sent by a distant device. That way if the Wanda protocol were initiated on the target (perhaps accidentally) and the Wand were not present to monitor for rogue frames, the target would be able to reject the distant adversary's frames on its own, without the Wand's help.

In this chapter we present a technique called SNAP: SiNgle Antenna Proximity. SNAP is a novel method for a single-antenna target device to quickly determine when it is in close proximity to a transmitting antenna. SNAP leverages the repeating nature Wi-Fi's preamble and the characteristics of a transmitting antenna's *near field* (i.e., the region physically close to the antenna) to detect proximity with a transmitter. When the target device is physically close to a transmitter, near-field effects will cause the repeated portions of the preamble to differ in phase and amplitude, whereas when the target device is far from the transmitter, the repeated portions of the preamble will be received with a consistent phase and amplitude. We use these observations to determine when the single-antenna target device is in close proximity to a transmitter.

4.1.1 Contributions

We make the following contributions in this chapter:

- a novel method for a single-antenna device to quickly determine when it is in close proximity with a transmitting device;
- a reference Wi-Fi implementation that performs the same frame decoding steps *any* Wi-Fi device must perform; and
- an experimental evaluation of the technique using several popular types of antennas.

4.2 Wi-Fi Preamble

In this section we briefly describe the Wi-Fi preamble, focusing on the repeating portions of the Long Training Field (LTF). We show that when the target is not in the transmitter's near field, even though the channel changes the transmitted signal, the repeated portions are changed consistently and are received with matching phase and amplitude. In Section 4.3 we show that when the target is in the transmitter's near field, this consistency does not hold, allowing SNAP to determine proximity with the transmitter.

4.2.1 PHY layer preamble format

Every OFDM Wi-Fi frame begins with a physical (PHY) layer preamble to aid in synchronizing the transmitter and receiver. In this thesis we focus on 20 MHz wide channels described in 802.11a/g/n/ac, but the technique could easily be extended for wider channels available in 802.11n/ac. The format of the PHY layer preamble is shown in Figure 4.1 and consists of a Short Training Field (STF) followed by a Long Training Field. The STF consists of 10 identical short training symbols (denoted t_1 through t_{10} in Figure 4.1) where each STF symbol is sampled 16 times, for a total of 160 samples. The STF is used by the receiver for frame detection,

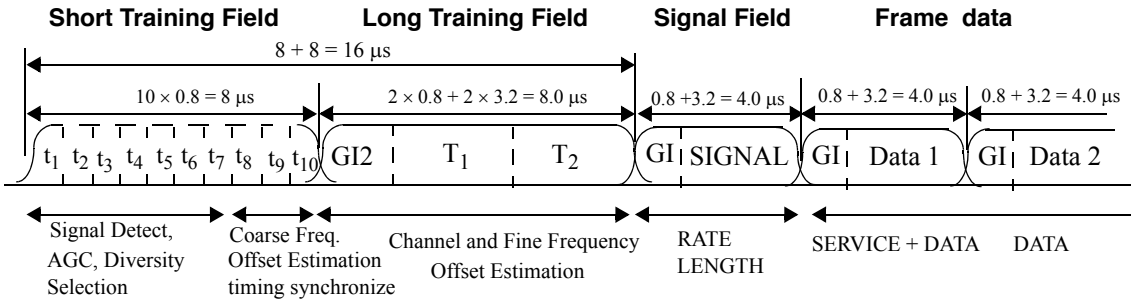

Figure 4.1: Wi-Fi OFDM PHY preamble format. Each Wi-Fi frame begins with a PHY layer preamble consisting of a 160-sample Short Training Field (denoted $t_1 \ldots t_{10}$), a Long Training Field comprised of a 32-sample guard interval (denoted $GI2$) and two identical 64-sample symbols (denoted T_1 and T_2), and a Signal Field comprised of a 16-sample guard interval (denoted GI), and 64 samples containing data about the Wi-Fi frame. Data symbols in the Wi-Fi frame follow the preamble. Reproduced from IEEE standard [57].

automatic gain control, coarse frequency offset estimation, and rough symbol timing synchronization. We discuss the STF in more detail in Section 4.4 below.

A Long Training Field (discussed next) follows the Short Training Field. After the preamble, a Signal field encoded with Binary Phase Shift Keying (BPSK) provides information about the rest of the Wi-Fi frame including the number of bytes and the encoding scheme used on the data portion of the frame. Data carried by the Wi-Fi frame follows the Signal field and each OFDM data symbol consists of a 16-sample guard interval (denoted GI in Figure 4.1) and 64 samples carrying the actual symbol data (see Section 3.5 for a more detailed discussion of OFDM).

4.2.2 Long Training Field

The LTF consists of a 32-sample guard interval (denoted GI2 because it is twice as long as other guard intervals in the frame) followed by two identical 64-sample OFDM symbols which are denoted T_1 and T_2 in Figure 4.1. The guard interval together with T_1 and T_2 make a total of 160 samples in the LTF. Because T_1 and T_2 are identical, the phase and amplitude of sample i in symbol T_1 matches the phase and amplitude of sample $i + 64$ in T_2, where $i = 0 \ldots 63$. This relationship

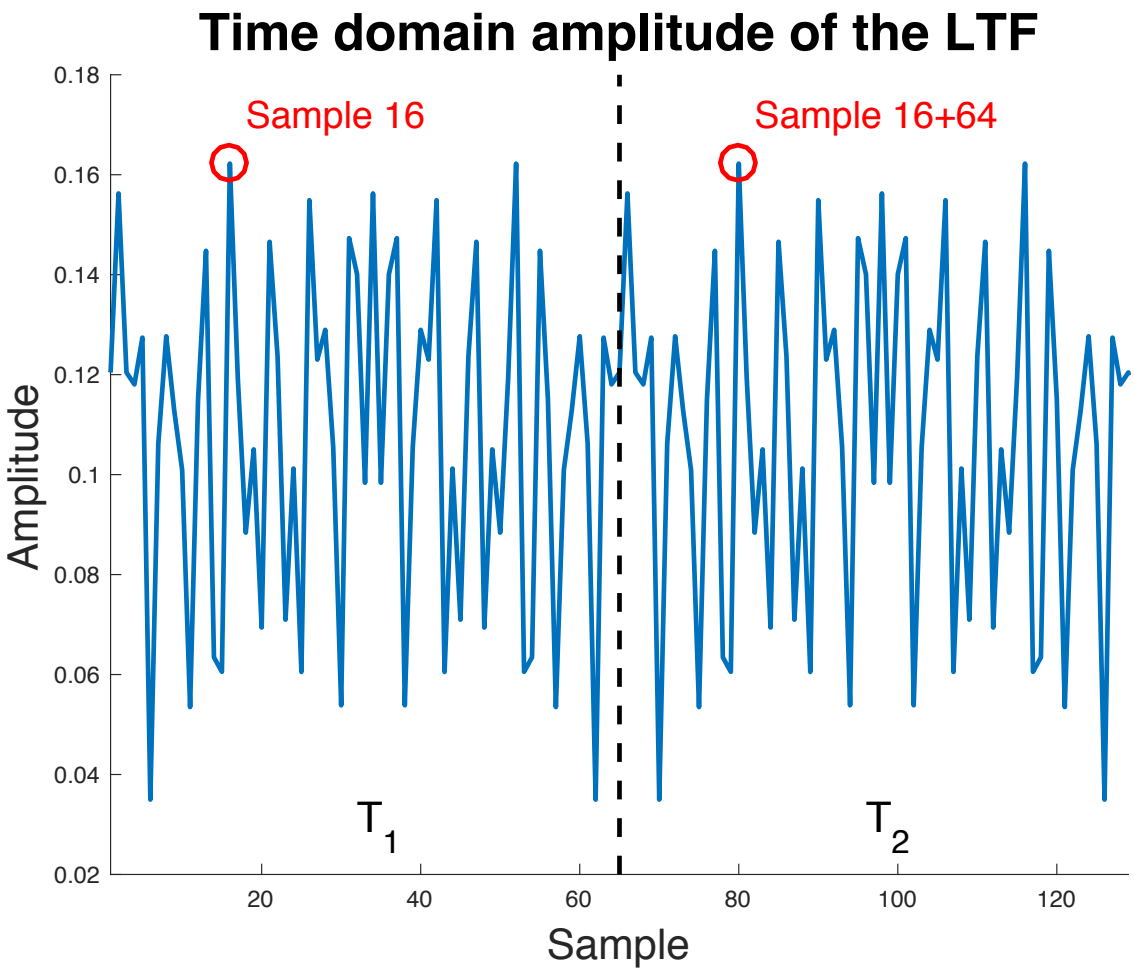

Figure 4.2: Time domain amplitude of the Long Training Field. In the time domain, samples i and $i + 64$ match in phase and amplitude. Here we highlight how sample 16 in T_1 matches sample 80 (64+16) in T_2.

between samples is shown in Figure 4.2. Each sample is represented by a complex number of the form $i + jq$, where i is the real component and q is the imaginary component of the complex number and $j = \sqrt{-1}$ (see Section 3.1). For clarity we plot the amplitude of the signal (i.e., $\sqrt{i^2 + q^2}$) in time.

We can convert the time-domain samples into an equivalent frequency-domain representing by taking a *Discrete Fourier Transform (DFT)*, which is nearly always implemented in real hardware with a *Fast Fourier Transform (FFT)*. As discussed in Section 3.5, Wi-Fi receivers perform a 64-point FFT over the received time-domain

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–32	0.000	0.000		–16	1.000	0.000		0	0.000	0.000		16	1.000	0.000
–31	0.000	0.000		–15	1.000	0.000		1	1.000	0.000		17	–1.000	0.000
–30	0.000	0.000		–14	1.000	0.000		2	–1.000	0.000		18	–1.000	0.000
–29	0.000	0.000		–13	1.000	0.000		3	–1.000	0.000		19	1.000	0.000
–28	0.000	0.000		–12	1.000	0.000		4	1.000	0.000		20	–1.000	0.000
–27	0.000	0.000		–11	–1.000	0.000		5	1.000	0.000		21	1.000	0.000
–26	1.000	0.000		–10	–1.000	0.000		6	–1.000	0.000		22	–1.000	0.000
–25	1.000	0.000		–9	1.000	0.000		7	1.000	0.000		23	1.000	0.000
–24	–1.000	0.000		–8	1.000	0.000		8	–1.000	0.000		24	1.000	0.000
–23	–1.000	0.000		–7	–1.000	0.000		9	1.000	0.000		25	1.000	0.000
–22	1.000	0.000		–6	1.000	0.000		10	–1.000	0.000		26	1.000	0.000
–21	1.000	0.000		–5	–1.000	0.000		11	–1.000	0.000		27	0.000	0.000
–20	–1.000	0.000		–4	1.000	0.000		12	–1.000	0.000		28	0.000	0.000
–19	1.000	0.000		–3	1.000	0.000		13	–1.000	0.000		29	0.000	0.000
–18	–1.000	0.000		–2	1.000	0.000		14	–1.000	0.000		30	0.000	0.000
–17	1.000	0.000		–1	1.000	0.000		15	1.000	0.000		31	0.000	0.000

Figure 4.3: Frequency domain representation of T_1 and T_2 in the Long Training Field. Re is the real component and Im is the imaginary component of the complex number representing the phase and amplitude of each complex number. Reproduced from IEEE standard [57].

samples to transform the time-domain samples into the frequency domain. The FFT operation yields 64 complex numbers representing the phase and amplitude of 64 subcarriers, indexed from -32 to +31. Figure 4.3 shows T_1 and T_2 represented in the frequency domain. Provided samples in the time domain in T_1 match corresponding samples in T_2 at the receiver, the phases and amplitudes of each subcarrier after an FFT of the samples in T_1 will also match the phases and amplitudes of each subcarrier after an FFT of the samples in T_2. If the samples in the time domain do not match, however, the phases and amplitudes of the subcarriers will also not match.

We note in Section 2.2.2, however, that the channel between the transmitter and receiver will modify the transmitted signal because the signal takes multiple

paths while in flight, reflecting off or passing through objects in the environment. These multi-path signals add up constructively or destructively at the receiver and the result is that the samples will not be received with the same phase and amplitude with which they were transmitted. This signal change suggests the possibility that samples in T_1 may not have the same phase and amplitude as the corresponding sample in T_2 when the signal is received. We see next, however, that those samples will match (except for random noise) when the receiver is not in the transmitter's near-field region.

4.2.3 Channel State Information

The channel between the transmitter and receiver can be mathematically expressed as [127]

$$\mathbf{y}[i] = \mathbf{H}\mathbf{x}[i] + \mathbf{w}[i] \tag{4.1}$$

where $\mathbf{y}[i]$ is the i^{th} received sample, $\mathbf{H}$ is the channel matrix representing the changes to the signal caused by the channel, $\mathbf{x}[i]$ is i^{th} the transmitted sample and $\mathbf{w}[i]$ is noise received with sample i. In a static environment (e.g., no moving objects), $\mathbf{H}$ is time invariant and causes the same shift in phase and amplitude for all samples in $\mathbf{x}$ because all transmitted samples take the same multipaths from sender to receiver. Neglecting noise, the result is that sample $\mathbf{y}[i]$ still matches sample $\mathbf{y}[i+64]$ in phase and amplitude, even though they no longer match $\mathbf{x}[i]$ due to the effects of $\mathbf{H}$.

This phase and amplitude change in the received sample compared with the transmitted sample is normal for wireless communication and is one of the reasons why Wi-Fi uses a preamble. The phase and amplitude of the preamble samples are pre-defined by the Wi-Fi specification and are known to both the transmitter and

receiver. The receiver uses these known phase and amplitude values in the STF to detect the start of the frame and apply a coarse frequency correction. Next it uses the LTF to synchronize symbol timing and apply fine frequency correction. Finally, because each subcarrier may be impacted differently by the channel, the receiver performs an FFT of the received time-domain signal to independently measure the phase and amplitude of each frequency-domain subcarrier in the LTF. The receiver computes the difference from the known transmitted phases and amplitudes for each subcarrier (see Figure 4.3) and the received phases and amplitudes to estimate the channel's impact on each subcarrier. As discussed in Section 3.7, this estimate is called *Channel State Information* or CSI. The receiver then applies the opposite CSI phase and amplitude to the remaining portion of the Wi-Fi frame to correct for the channel's changes.

4.2.4 Coherence time

Above we consider an environment with no moving objects and we see in Equation (4.1) that $\mathbf{H}$ is time invariant so corresponding samples in T_1 and T_2 will be received with identical phase and amplitude (except for noise). In the real world, however, the transmitter, receiver, or other objects may be moving and that movement may impact the signal. A channel is said to be *coherent* if it is stable over a particular time interval. We can calculate the needed coherence time, T_c, for the corresponding portions of the preamble. If the channel is coherent over T_c then the corresponding samples will be received with the same phase and amplitude.

Wi-Fi samples at 20 MHz, meaning it takes 20 million samples per second. The time for one sample, T_s, is then 1/(20,000,000 samples/second) which equates to 50 ns. T_1 and T_2 are a total of 128 samples long, and because we are interested in how T_1 matches T_2, we require a coherence time of 6.4 μs (50 ns/sample $\times$ 128 samples = 6.4 μs). In this case, if the channel is stable over 6.4 μs, then T_1 will match

T_2 (aside from noise). Moving objects, however, may cause a mismatch.

4.2.5 Moving objects

Moving objects can potentially cause changes in the signal in two ways: (1) by changing the signal's path length or (2) by causing a frequency shift via the Doppler effect. Next we examine both of these possibilities with regard to the required coherence time $T_c = 6.4$ μs.

Changing path length

The length of the path between the transmitter and receiver affects the phase and amplitude of the signal according to the following formula [127]

$$\mathbf{H} = \sum_{p=1}^{P} a_p e^{-j2\pi d_p/\lambda} \tag{4.2}$$

where a_p is the attenuation of the signal along the path p, d_p is the length of path p, and λ is the wavelength. The path length p for a path may change as the transmitter, receiver, or multipath-inducing objects move. To cause a significant change in the signal between corresponding samples, however, the movement would need to cause a change in path length of more than one-quarter wavelength (and one-half wavelength to cause maximum change) [127]. In Wi-Fi's 2.4 GHz band, the wavelength λ is approximately 12 cm, so an object would need to move approximately $\lambda/4 = 3$ cm in 6.4 μs to significantly impact the phase and amplitude between corresponding LTF samples. This translates to a speed of over 17,000 km/hour (and roughly twice this speed for Wi-Fi's 5 GHz band). Given the extraordinary speed an object would need to be moving to cause a substantial change in path length in the short coherence time needed for the preamble, we eliminate changing path lengths as a possible explanation for corresponding LTF samples to have different

phases and amplitudes.

Doppler effect

A moving transmitter, receiver, or other object may cause a shift in the received signal's frequency. Called the Doppler effect, if the shift is large enough, it may cause a change in corresponding portions of the LFT. Channel coherence time can be related to the frequency shift caused by moving objects as [127]

$$T_c = \frac{\lambda}{4fv/c} \tag{4.3}$$

where f is the signal frequency, v is the maximum speed of any moving object, and c is the speed of light.

In our case, the $T_c = 6.4$ μs, so we can solve for the minimum velocity that would cause a significant phase shift as:

$$v \geq \frac{\lambda}{4fT_c/c}. \tag{4.4}$$

With a wavelength $\lambda \approx 12$ cm and a coherence time $T_c = 6.4$ μs, Equation (4.4) suggests moving objects need to be traveling even more than 17,000 km/hour to substantially change the phase and amplitude of the signal. Again we find that the environment is unlikely to cause a significant different between corresponding portions of the LTF.

4.2.6 Summary

The implication of this section is that the channel between the target device and a transmitter will not cause a significant difference in the phase and amplitude of repeating portions of the LTF. This section, however, implicitly assumes that the signal produced by the transmitter is planar in nature. We find in the next

section that near-field effects can cause differences in the phase and amplitude of corresponding LTF samples. We use those differences to detect when the target is near the transmitter. If those changes are not present, based on this section, we assume the target is far from the transmitter.

4.3 Near Field

We've seen that due to the short time duration between the corresponding portions of the LTF, the environment is not likely to cause a difference in phase and amplitude. In this section we see that the near-field region around the antenna *can* cause changes between corresponding samples. The regions around a transmitting antenna are generally classified into three different regions as shown in Figure 4.4. The region boundaries are not sharp, but rather transition gradually from one region to another.

Using the orientation depicted in Figure 4.5, where the antenna is aligned vertically with the z axis, the magnetic fields $\mathbf{H}$ relative to each axis[1] are determined by the following formulas [9]:

$$H_r = H_\theta = 0 \tag{4.5a}$$

$$H_\phi = j\frac{kI_0 l_t \sin\theta}{4\pi r}\left[1 + \frac{1}{jkr}\right]e^{-jkr} \tag{4.5b}$$

and the electric fields $\mathbf{E}$ are determined by

[1]Here $\mathbf{H}$ refers to the magnetic field, whereas previously the same symbol referred to CSI. Unfortunately, this overloading is common in the literature.

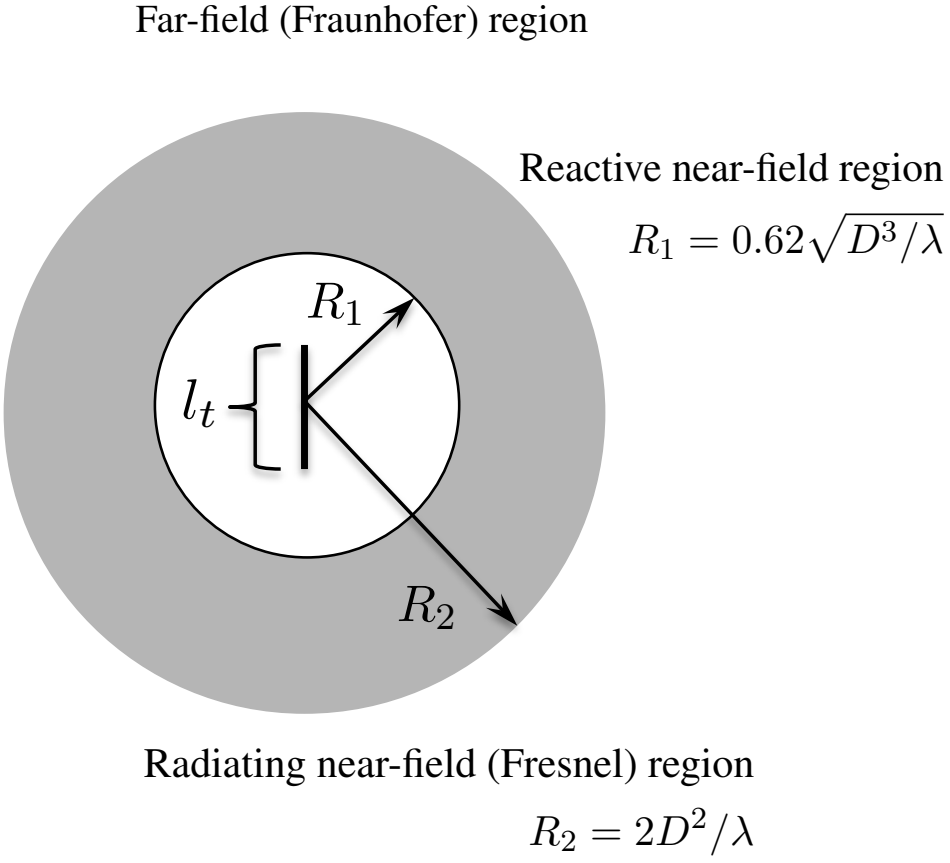

Figure 4.4: Regions surrounding a transmitting antenna. A transmitting antenna with length l_t has three surrounding regions: the reactive near-field is closest to the antenna and extends to a distance R_1, the radiating near-field begins where the reactive near-field ends and extends to R_2, and the far-field begins where the radiating near-field ends and extends to infinity. D is defined as the length of the transmitting antenna l_t plus the length of the receiving antenna, l_r.

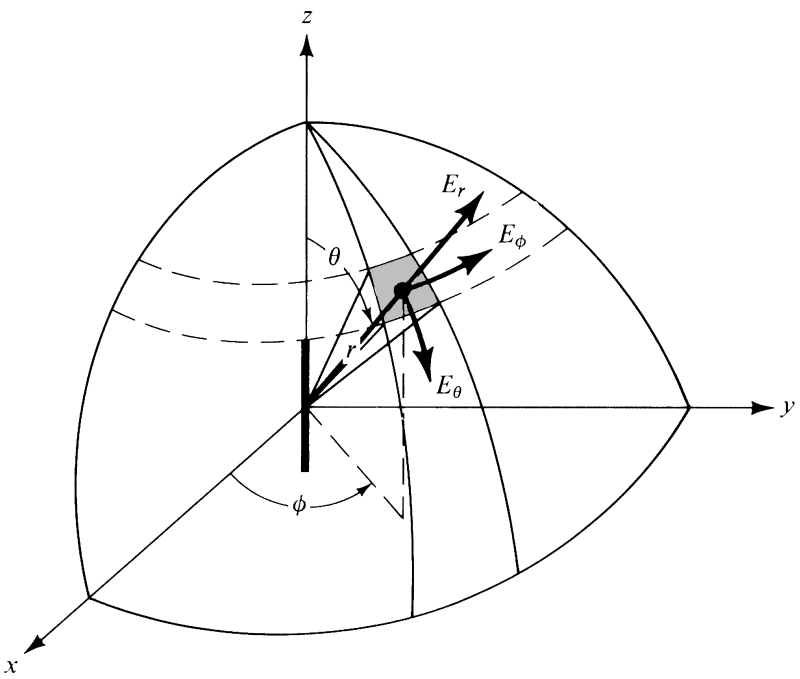

Figure 4.5: Antenna orientation. To provide a common reference, the transmitting antenna is typically assumed to be aligned with the vertical (z) axis as shown. Reproduced from Balanis [9].

$$E_r = \eta \frac{I_0 l_t \cos\theta}{2\pi r^2} \left[1 + \frac{1}{jkr}\right] e^{-jkr} \tag{4.6a}$$

$$E_\theta = j\eta \frac{k I_0 l_t \sin\theta}{4\pi r} \left[1 + \frac{1}{jkr} - \frac{1}{(kr)^2}\right] e^{-jkr} \tag{4.6b}$$

$$E_\phi = 0 \tag{4.6c}$$

where $j = \sqrt{-1}$, $k = 2\pi/\lambda$ is the wavenumber, I_0 is current applied to the transmitter, l_t is the length of the transmitting antenna, $\eta = 120\pi$ is the intrinsic impedance of free space, θ is the vertical angle between the transmitter and receiver, ϕ is the horizontal angle between the transmitter and receiver, and r is the distance extending radially from the transmitter.

4.3.1 Reactive near-field region

The reactive near-field region is the region closest to the transmitting antenna, where $kr < 1$ (or equivalently, where $r < \lambda/2\pi$). In this region the reactive (e.g, non-radiating) field dominates and there is a high content of non-propagating stored energy. We see this energy accounted for in the second term in the brackets in Equations (4.5b) and (4.6a). In Equation (4.6b), the third term in brackets ($\frac{1}{(kr)^2}$) dominates as $kr \ll 1$, and the second term ($\frac{1}{jkr}$) is greater than the first term (1). The result is that the wavefront is not spherical because the electric and magnetic fields are not yet aligned, and in addition to the radiated energy described by the first term in brackets in Equations (4.5) and (4.6), there is a great deal of stored, non-propagating energy due to the second term and third terms dominating at close range.

Because energy is stored close to the transmitting antenna, the presence of another resonant antenna in the reactive near-field region can cause mutual coupling and the transmitting antenna can detect an increased power draw due to the presence of the nearby antenna [19]. This is an area of future work, where the single antenna device may be able to detect the presence of a nearby antenna by monitoring for an increased power draw due to mutual coupling.

With real antennas, the reactive near-field region is commonly estimated to extend from the surface of the antenna to roughly R_1, defined as [9]

$$R_1 = 0.62\sqrt{D^3/\lambda} \tag{4.7}$$

where $D = l_t + l_r$ is the combined length of the transmitting antenna, l_t, and the length of the receiving antenna, l_r, and λ is the signal wavelength. With Wi-Fi 2.4 GHz band, and quarter-wavelength dipole antennas, this region extends to roughly 2.7 cm from the transmitter. In Wi-Fi's 5 GHz band this region extends to

roughly 1.1 cm.

4.3.2 Radiating near-field (Fresnel) region

Sometimes referred to as the *Fresnel* or *intermediate field*, the radiating near-field region is the area between the reactive near-field and far-field regions. In this region $kr > 1$ and the electric and magnetic fields are predominantly in phase, but the wavefront is still not yet spherical as it is in the far-field region. Examining Equations (4.5b) and (4.6a) we see that, unlike in the reactive near field, the first term in the brackets (1) begins to dominate the second term ($\frac{1}{jkr}$) because kr is greater than one. Likewise, in Equation (4.6b), the first term in the brackets (1) begins to dominate the second ($\frac{1}{jkr}$) and third terms ($\frac{1}{(kr)^2}$). Because of the increasing value of kr compared with the reactive near-field region, the energy in the radiating near field is largely real, that is, radiated energy.

We can estimate the average power of the signal, W, using the following equation [9]:

$$\mathbf{W} = \frac{1}{2}(\mathbf{E} \times \mathbf{H}^*) \tag{4.8}$$

where * denotes complex conjugate and **E** and **H** are determined using Equations (4.5) and (4.6). W can be decomposed into its radial, W_r, and vertical, W_θ components as follows [9]:

$$W_r = \frac{\eta}{8}\left|\frac{I_0 l_t}{\lambda}\right|^2 \frac{\sin^2\theta}{r^2}\left[1 - j\frac{1}{(kr)^3}\right] \tag{4.9a}$$

$$W_\theta = j\eta\frac{k|I_0 l_t|^2 \cos\theta \sin\theta}{16\pi^2 r^3}\left[1 + \frac{1}{(kr)^2}\right]. \tag{4.9b}$$

Mapping the power of each of these components, we see in Figure 4.6 for

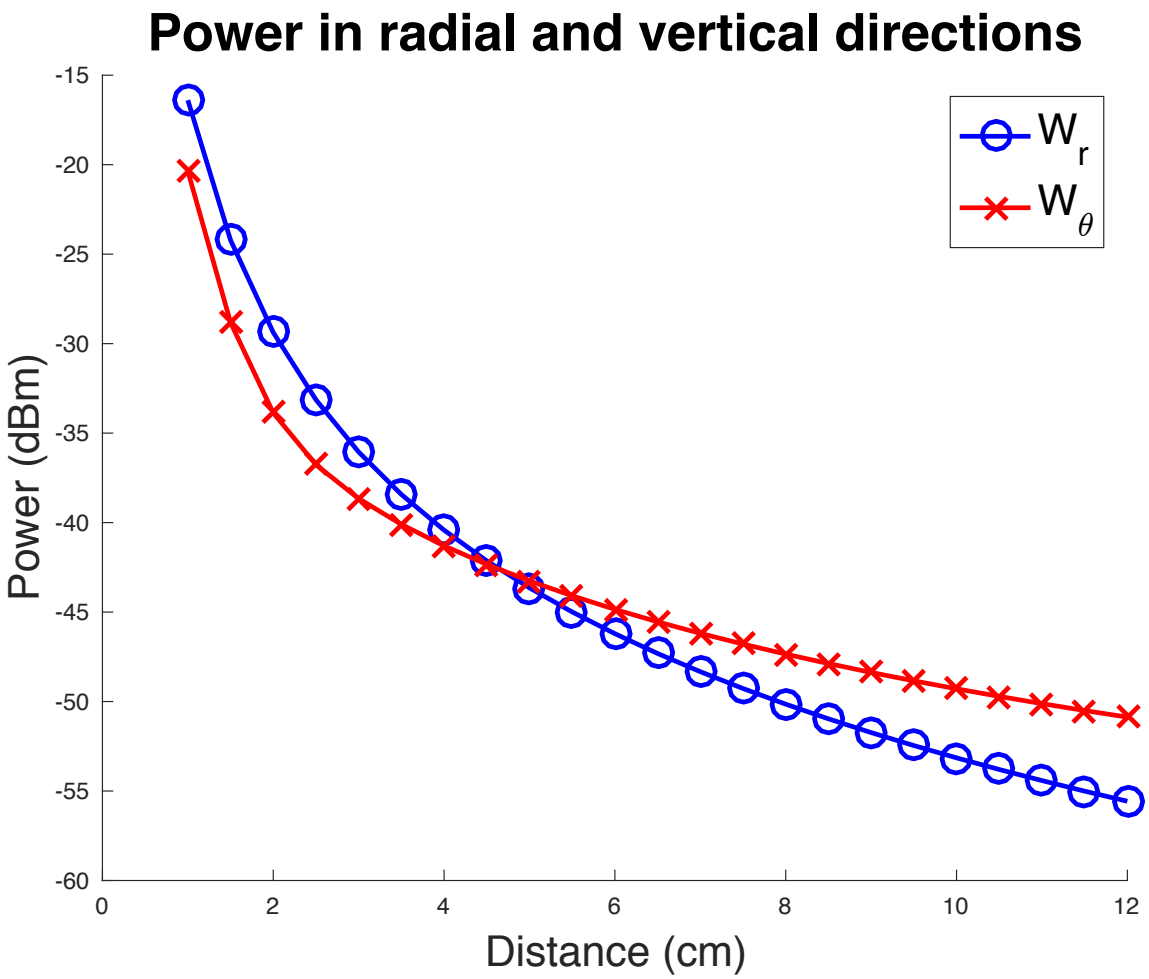

Figure 4.6: Power of the radial and vertical components of a signal. Using Wi-Fi's 2.4 GHz band and Equation (4.9), the vertical component W_θ begins to dominate the radial component W_r at about 5 cm.

Wi-Fi's 2.4 GHz band with quarter-wavelength antennas, at distances larger than roughly 5 cm the W_θ component begins to dominate the W_r component as it does in the far field. At distances closer than about 5 cm the radial component is stronger than the vertical component. This relative strength suggests the power pulses inward and outward near the transmitter, whereas at greater distances, the radial component dies out and vertical component takes over. This vertical component domination is indicative of signals in the far-field region, whereas radial component domination is indicative of signals in the radiating near-field region.

With real antennas, the radiating near-field region is commonly estimated to extend from R_1 to R_2, where R_2 is defined [9]

$$R_2 = 2D^2/\lambda \tag{4.10}$$

where D is the combined length of the transmitting antenna, l_t, and the length of the receiving antenna, l_r, and λ is the signal wavelength. With Wi-Fi 2.4 GHz band, and quarter-wavelength dipole antennas, Equation (4.10) suggests this region extends to roughly 6.2 cm from the transmitter. This roughly matches the results shown in Figure 4.6 using Equations (4.9) where the radial component of the energy begins to dominate as it does it the far field. We note, however, that this boundary is not a sharp distinction between the radiating near-field and the far-field. We see in Figure 4.6 that the radial and vertical components are nearly equivalent for some distance past this point, but that by 12 cm distance W_θ is roughly three times stronger than W_r.

4.3.3 Far-field (Fraunhofer) region

The far field, sometimes referred to as the *Fraunhofer* region, is the area far from the transmitting antenna where $kr \gg 1$. Because kr is large in the far field, several of the terms in Equations (4.5) and (4.6) become extremely small and the **E** and **H** fields can be approximated by the much simpler formulas [9]:

$$E_\theta \simeq j\eta \frac{kI_0 l_t e^{-jkr}}{4\pi r} \sin\theta \tag{4.11a}$$

$$E_r \simeq E_\phi = H_r = H_\theta = 0 \tag{4.11b}$$

$$H_\phi \simeq j\frac{kI_0 l_t e^{-jkr}}{4\pi r} \sin\theta. \tag{4.11c}$$

In Equation (4.11) we see that the electric and magnetic fields are aligned orthogonal to each other (e.g., θ is orthogonal to ϕ), transverse to the direction of propagation, and are in time synchronization. This arrangement creates a spherical wavefront with average power given by Equation (4.8).

4.3.4 Near-field impact on corresponding samples

At ranges closer than roughly R_2, because the overall **E** and **H** fields are not in phase with respect to time, and because those fields are do not have equal magnitude, they form a vector that rotates in time in a plane parallel to the direction of propagation, rather than the stable orthogonal relationship seen in the far-field region [9]. Wi-Fi samples taken as the **E** and **H** fields rotate can result in different phase and amplitude readings between corresponding samples in the LTF. We see in Section 4.5, as suggested by Figure 4.6, there is a difference in phase and amplitude in the corresponding LTF samples out to about 12 cm.

4.3.5 Pilot subcarriers

Because the **E** and **H** fields near the antenna are rotating in time and can cause differences in the corresponding portions of the preamble, the samples in the data portion of the frame will also be received with a rotation-induced phase and amplitude offset. One might wonder how Wi-Fi is able to receive frames at all at close range. The answer lies in Wi-Fi use of pilot subcarriers. In addition to estimating the channel from the LTF, also known as a *block-type* channel estimate where there is one block of data sent at known phases and amplitudes on all subcarriers, Wi-Fi also uses *comb-type* channel estimation where several *pilot* subcarriers, separated by S subcarriers, are included in each Wi-Fi symbol [111]. Figure 4.7 illustrates the difference between block and comb-type channel estimation. Wi-Fi uses subcarriers -21, -7, 7, and 21 as pilot subcarriers. The receiver then gets an initial channel

estimate from block-type LTF subcarriers, then updates its channel estimate for each Wi-Fi symbol by comparing the received phases and amplitudes on the comb-type pilot subcarriers with the known phases and amplitudes transmitted on those subcarriers. The receiver then corrects each Wi-Fi symbol using the refined channel estimate from the pilot subcarriers. In this way Wi-Fi is able to correct each symbol as the fields rotate in time.

Pilot subcarriers allow Wi-Fi to effectively deal with moving objects. We see in Section 4.2 that an object would need to be moving at an extremely high speed to affect the preamble. A Wi-Fi frame, however, may be composed on many OFDM symbols, each lasting 4 μ seconds. A moving object might be able to move $\lambda/4$ cm during the frame transmission time, and thus might affect the phase and amplitude at the receiver. For example, a Wi-Fi frame consisting of 1,500 OFDM symbols will take 6 milliseconds to transmit (plus the preamble). This transmit time suggests that an obstacle moving at 19 km/hr would move more than $\lambda/4$ cm and thus could make a significant change to the received signal over the transmission time. Additionally, Equation (4.3) suggests that an object moving roughly 230 km/hr would cause a significant frequency change. Wi-Fi's comb-type channel estimation allows it to dynamically correct each symbol during the relatively long frame transmission time and overcome these issues caused by moving objects. This same feature allows Wi-Fi to receive a frame in the presence of rotating **E** and **H** fields.

4.4 Implementation

We wanted to test the insight from Section 4.3 that the phase and amplitude of corresponding Long Training Field samples would change when the receiver is close to a transmitting antenna (e.g., when the receiver is in the transmitter's near-field region) and would not change when the receiver is far from the transmitter.

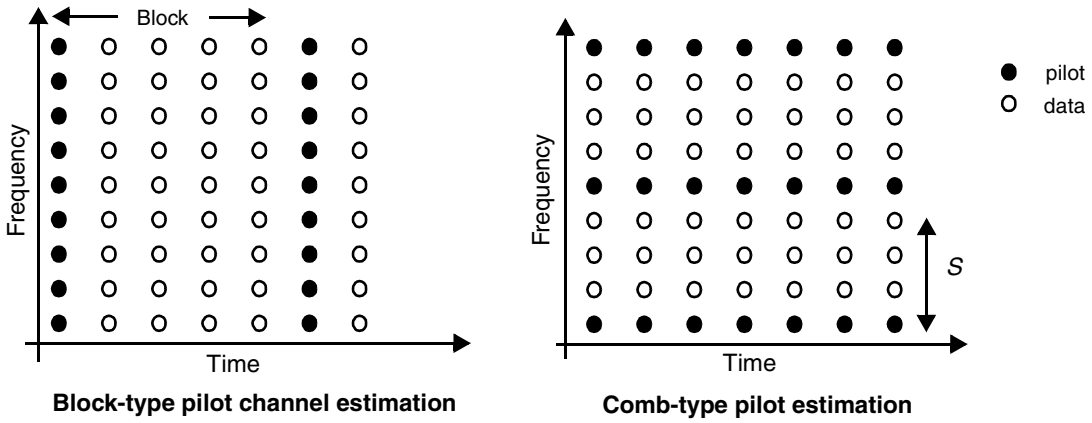

Figure 4.7: Block and comb-type channel estimation. Wi-Fi uses block-type channel estimation from the LTF and uses comb-type channel estimation from pilot subcarriers in each Wi-Fi symbol. Reproduced from Shen and Martinez [111].

Because we could not get granular phase and amplitude data for corresponding portions of the Wi-Fi LTF from Commercial-Off-The-Shelf (COTS) Wi-Fi adapters (even with the CSI tool frequently used by researchers [52]), we built our own Wi-Fi receiver using an Ettus Research USRP N210 Software Defined Radio (SDR) [30] with a UBX40 daughterboard and GNU Radio [125]. The USRP SDR hardware allowed us to receive signals from 10 MHz to 6 GHz, enough bandwidth to cover both the 2.4 and 5 GHz Wi-Fi bands, and GNU Radio is an open-source computer program that processes signals sent by the SDR hardware.

4.4.1 Custom Wi-Fi receiver steps

At a high level, *all* Wi-Fi receivers must accomplish two steps: (1) detect the presence of an incoming Wi-Fi frame and (2) decode the frame according to the Wi-Fi specification [57]. In the detection step, receivers look for the presence of the Short Training Field (STF) briefly discussed in Section 4.2. Since the Wi-Fi specification details both the frequency and time domain samples of the STF, both the transmitter and receiver know ahead of time the precise phases and amplitudes of that field. The receiver can then look for the STF pattern in the incoming signal. When it finds

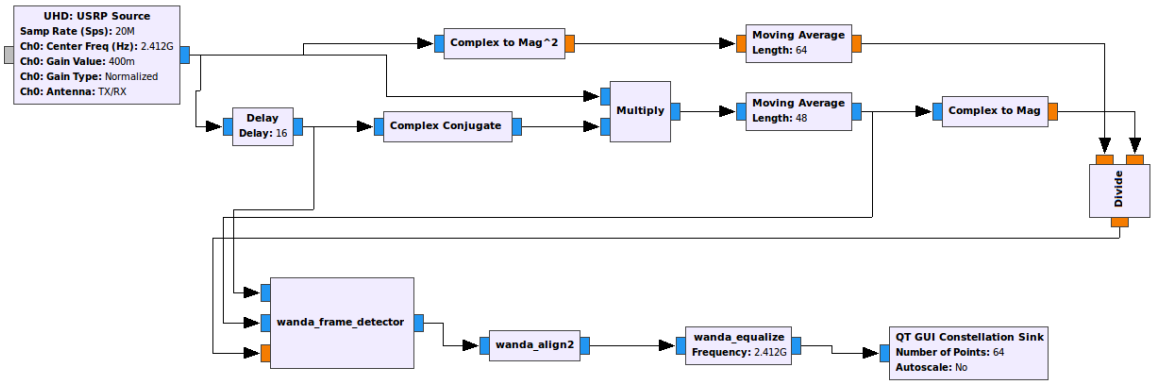

Figure 4.8: Custom Wi-Fi receiver software. Our receiver was implemented with GNU Radio and involved several custom signal-processing blocks: *frame_detector* looks for the presence of the Short Training Field that indicates the start of a Wi-Fi frame, *frame_align* synchronizes symbol timing and estimates CSI, and *frame_equalize* performs equalization and decodes the Signal field.

the pattern, it performs automatic gain control and coarse frequency correction, then passes the samples to the decoding phase. The decoding phase uses the LTF to determine the precise symbol alignment then estimates and corrects for the effects of the channel on the signal. After correcting for the channel effects, the receiver next reads the Wi-Fi Signal field to learn the number of bytes remaining in the frame and modulation coding scheme used on the remaining bytes. Our GNU Radio implementation is shown in Figure 4.8. We provide more details on the frame detection and decoding steps next.

4.4.2 Detecting incoming Wi-Fi frames

Because the receiver and transmitter both know the signal phase and amplitude of the STF, a natural approach would be to use a *matched filter* to look for the presence of the STF amid the other signals the receiver picks up. A matched filter is an optimal linear filter that maximizes the Signal-to-Noise Ratio (SNR) when looking for a known signal in a noisy signal [127]. To find the known signal (e.g., the STF), a matched filter convolves the noisy signal with a conjugated time-reversed version

of the known signal. We attempted to write matched filter to find the STF amid the incoming signals, but found that even with a gigabit connection between GNU Radio and the SDR, the matched filter was too slow to keep up in real time (later we learned others also tried and failed with this approach [14]). We also attempted to implement the well-known Schmidl & Cox method [106], but also found it to be too slow for SDR hardware and GNU Radio software. This method is better suited to implementation in a Field Programmable Gate Array (FPGA) than with an SDR.

Because the matched filter was too slow, we follow the approach outlined by Liu [72] and implemented by Bloessl [14]. The STF specification calls for 10 repeating 16-sample sequences, so we look for an autocorrelation between samples separated in time by 16 samples. If we see a large correlation over a window of samples, it is likely the samples are the Short Training Field of a Wi-Fi frame. Following Liu and Bloessl, we calculate an autocorrelation value as follows:

$$a[n] = \sum_{k=0}^{N_{win}-1} s[n+k]s[n+k+16]^* \tag{4.12}$$

where $s[n]$ is sample n produced by the SDR hardware, $s[n]^*$ denotes the complex conjugate of sample $s[n]$, and N_{win} is a window of 48 samples (set experimentally by Blossel).

To help reduce false positives, once the autocorrelation coefficients $a[n]$ have been calculated we then normalize the autocorrelation coefficients with the total power received in those samples as follows:

$$p[n] = \sum_{k=0}^{N_{win}-1} s[n+k]s[n+k]^* \tag{4.13a}$$

$$c[n] = \frac{|a[n]|}{p[n]} \tag{4.13b}$$

where $|a[n]|$ indicates the magnitude of $a[n]$. These steps are implemented using

standard GNU Radio blocks as shown in the first row of Figure 4.8. Samples are then passed to our *frame_detector* block to determine when a frame starts and to apply coarse frequency correction.

frame_detector

The *frame_detector* block attempts to find the start of a Wi-Fi frame by looking for two consecutive values of c from Equation (4.13b) that are above a pre-defined threshold τ (experimentally set to 0.56). Also, because we are interested in signals from nearby devices, we add a check to ensure the signal power (squared magnitude) of the most recent samples crosses a threshold m (set experimentally to 0.25). If both the normalized autocorrelation values and the magnitude cross their thresholds, we make a coarse frequency correction and for speed pass 400 samples on to following processing blocks. The 400 samples account for the STF (160 samples), the LTF (160 samples), and the Signal field (80 samples).

The reason we pass along enough samples to account for the entire STF instead of simply passing along samples for the LTF and Signal field is because the autocorrelation portion may not precisely align on the STF boundary and thus we cannot simply skip 160 samples ahead to where we expect to see the guard interval (GI2 in Figure 4.1) and T_1 and T_2 in the LTF to begin. To be safe, we pass along the samples of the STF in addition to the LTF and Signal field to the *frame_align* block that determines the exact starting sample for the LTF.

If the autocorrelation portion declares a frame start a few samples after the actual start, the next block will simply receive a few more samples than it needs. Due to the power normalization step in Equation (4.13), the autocorrelation portion should not indicate the start of a frame early, but even if does, or it causes a false positive, the error will be quickly detected in the equalizer because the Signal field will not decode properly.

frame_align

The *frame_align* block is a custom block that attempts to find the exact sample where a Wi-Fi LTF starts. Because Wi-Fi starting points are relatively infrequent, and this block will not start until it receives 400 samples from the *frame_detector* block, we find that we can implement a matched filter here and still keep up with incoming samples. To implement this match filter, we pre-calculate a time-reversed complex conjugate version of the LTF, then wait for a block of 400 samples to arrive. We then perform a convolution over the incoming samples with the conjugated, time-reversed LTF. The sample with highest value after convolution is the likely start of the LTF. This block then marks the sample with the maximal value as the start of the LTF, skips the guard interval, and passes 208 total samples to the *frame_equalize* block. The 208 samples are comprised of 128 samples for the two repeating portions of the LTF (T_1 and T_2 from Figure 4.1), plus 80 samples for the Signal field.

4.4.3 Decoding frames

Now that the signal is aligned to the starting sample, we decode the Signal field to ensure the samples comprise a valid Wi-Fi frame. Because we are only interested in the difference between matching samples in the Wi-Fi preamble, in our implementation we only decode the Signal field to ensure our algorithm has found a valid Wi-Fi frame and, for speed, we do not decode the remainder of the Wi-Fi frame. A real Wi-Fi receiver, however would need to decode the additional portions of the Wi-Fi frame, but the logic could be implemented in hardware and would likely outperform our SDR implementation by a large margin.

frame_equalize

This block receives 208 samples from the *frame_align* block where the first sample should be aligned with the first sample of T_1 in the LTF. This block then estimates the channel by performing an FFT over the first 64 samples, and a second FFT over the second 64 samples. This step converts the time-domain samples into the frequency domain. Because Wi-Fi uses an FFT of size 64, the FFT outputs 64 different frequencies, each representing a Wi-Fi subcarrier. This block compares the phase and amplitude of those subcarriers with the phase and amplitude mandated by the Wi-Fi specification to develop its estimate of the channel. We estimate Channel State Information for each of the 64 subcarriers using both portions of the LTF by following Liu [72] as:

$$H[k] = \frac{1}{2}\left(\frac{Y_1[k] + Y_2[k]}{X[k]}\right) \quad k = 0 \ldots 63 \tag{4.14}$$

where $H[k]$ is the CSI estimate for subcarrier k , $Y_1[k]$ is the received phase and amplitude of subcarrier k from an FFT of T_1, $Y_2[k]$ is the phase and amplitude of subcarrier k from an FFT of T_2, and $X[k]$ is the transmitted phase and amplitude prescribed by the Wi-Fi specification for subcarrier k.

This block then checks that the Signal field is valid. First it applies a CSI correction from Equation (4.14) to remove channel effects, then extracts the data bits from the 48 data-carrying subcarriers, performs a parity check on the Signal field, and computes the number of bytes and encoding used on the remainder of the frame. If the parity check passes and the encoding equates to a valid Wi-Fi MCS, we output the phase and amplitude of each of the k subcarriers from T_1 and T_2 as given by $Y_1[k]$ and $Y_2[k]$. We use that output to examine the difference in phase and amplitude between corresponding subcarriers in the LTF.

Based on the discussions in Sections 4.2 and 4.3, we expect the subcarriers to

match in phase and amplitude when the receiver is in the transmitter's far-field region, and to differ when the receiver is in the transmitter's near-field region. In the next section we test four different off-the-shelf Wi-Fi configurations using our SDR receiver and find we can determine reliably determine proximity with a single antenna receiver out to roughly 9 cm.

4.5 Evaluation

We used our custom Wi-Fi receiver software with the USRP SDR hardware discussed in Section 4.4 with a quarter-wavelength dipole antenna to receive Wi-Fi signals transmitted by several different Wi-Fi antennas. Because dipole and micropatch antennas are the most common antennas found in consumer electronics [9], we focused our testing on those types of transmitting antennas. All tests were conducted in a busy computer science lab where there were many moving obstacles.

4.5.1 Hardware setup

In Section 4.4 we discussed the custom Wi-Fi receiver software we created to capture the differences in corresponding portions of the LTF preamble. Here we briefly describe the hardware we used.

Receiver

We used a USRP N210 shown in Figure 4.9 with a quarter-wavelength dipole antenna as a Wi-Fi receiver. We connected the quarter-wavelength dipole to a circular base to hold the antenna upright, and we used a 3 m RF cable to connect the dipole antenna base to the USRP. We also connected the USRP to a laptop with a 2 m ethernet cable. The laptop ran GNU Radio and our custom Wi-Fi software described previously to process the samples coming from the hardware. We used

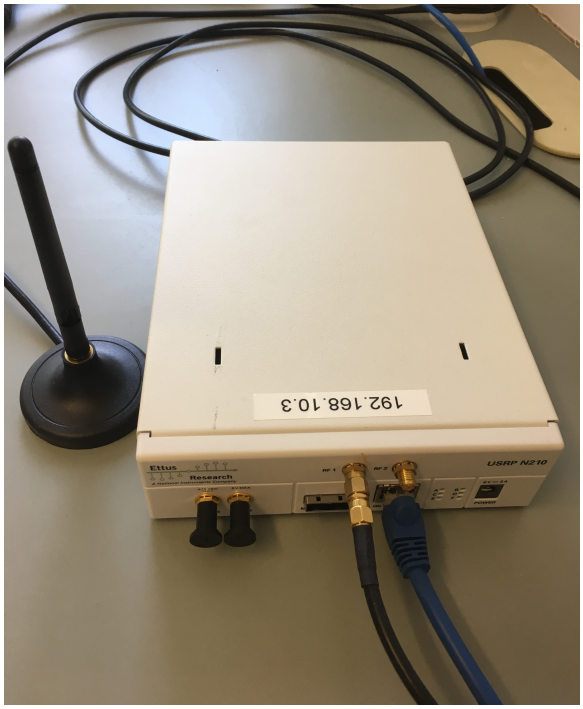

Figure 4.9: Wi-Fi receiver hardware. We used a USRP 210N Software Defined Radio with a quarter-wavelength dipole antenna as a Wi-Fi receiver. The USRP was connected to the antenna via a 3 m RF cable and to a laptop via a 2 m ethernet cable (laptop not shown). Photo by Timothy J. Pierson.

this configuration as a receiver for all tests.

Dipole transmit antennas

We used a half and quarter-wavelength dipole antennas connected to an internal Intel Ultimate N WiFi Link 5300 card [58] installed in a Linux laptop to test different transmitting dipole antennas. We attached each dipole antenna to a circular base to hold the antenna upright during testing and connected the base to the laptop with a 3 m long RF cable as shown in Figure 4.10. Each type of transmitting antenna was tested separately.

Figure 4.10: Transmitters. We used a quarter-wavelength and a half-wavelength dipole antenna as well as a micropatch antenna connected via a 3 m cable to an internal Intel Ultimate N WiFi Link 5300 adapter installed in a Linux laptop. We also used a Panda Ultra Wireless N USB adapter connected via a 1 m USB extender cable. Photo by Timothy J. Pierson.

Micropatch transmit antennas

We tested a Panda Ultra Wireless N USB Adapter [86] attached to a Linux laptop via a 1 m USB extender cable and micropatch antenna connected to the Intel 5300 card via a 3 m RF cable. Due to their small size, these antennas are popular in mobile consumer electronics such as cell phones.

4.5.2 Preamble error

We transmitted 1,000 Wi-Fi frames from each of the four different types of antennas using BPSK 1/2 encoding on Wi-Fi channel 1 at distances ranging from 2 cm to 3 m. The Wi-Fi frames were sent on Layer 2 using a Python program written with Scapy [13]. Frames were not acknowledged by the SDR.

We calculate the total Euclidean distance between the phase and amplitude of subcarriers in T_1 and T_2 as:

$$E_j = \sum_{k=-32}^{31} \sqrt{(\Re(Y_1[k]) - \Re(Y_2[k]))^2 + (\Im(Y_1[k]) - \Im(Y_2[k]))^2} \tag{4.15}$$

where E_j is the total Euclidean distance between the phase and amplitude of all subcarriers k for frame j, and where $\Re(Y_x[k])$ is the real component and $\Im(Y_x[k])$ is the imaginary components of the phase and amplitude of each subcarrier k in Y_x, for $x \in \{1, 2\}$. Recall from Section 4.4 that Y_1 is the result of an FFT over T_1 and Y_2 is the result of an FFT over T_2. We call this difference E_j the *preamble error* of a frame. If the subcarriers in the two corresponding portions of the LTF are substantially the same, then the preamble error will be small. If the subcarriers are different in the two corresponding portions of the LTF, then the preamble error will be large.

Figure 4.11 shows the difference between Y_1 and Y_2 for subcarrier 1 of one frame when the transmitter was located at 6 cm from the receiver and for subcarrier 1 of another frame sent from 30 cm. We see that at 30 cm the Y_1 matches Y_2, but at 6 cm Y_1 does not match Y_2. Figure 4.12 shows difference between Y_1 and Y_2 for all subcarriers of one frame. We see at 30 cm Y_1 and Y_2 match for all subcarriers, but at 6 cm many subcarriers do not match. The sum of the distances between Y_1 and Y_2 over all subcarriers makes up the preamble error and we see it is small at long range and large at close range.

We calculate the average preamble error over a number of frames for each antenna type as:

$$A_t = \frac{1}{n} \sum_{j=1}^{n} E_j \tag{4.16}$$

where $t \in \{\text{half-wavelength, quarter-wavelength, micropatch, Panda}\}$ is the type of antenna used to send Wi-Fi frames and $n = 1,000$ is the number of frames received. The average preamble error over all 1,000 frames sent at each distance for each antenna type is shown in Figure 4.13 for distances from 2 cm to 3 m. We show the distribution of preamble errors for these frames in Appendix B.1. As predicted in Section 4.3, at short range we see large preamble errors and at distances beyond roughly 12 cm, we see small preamble errors. This relationship holds across all

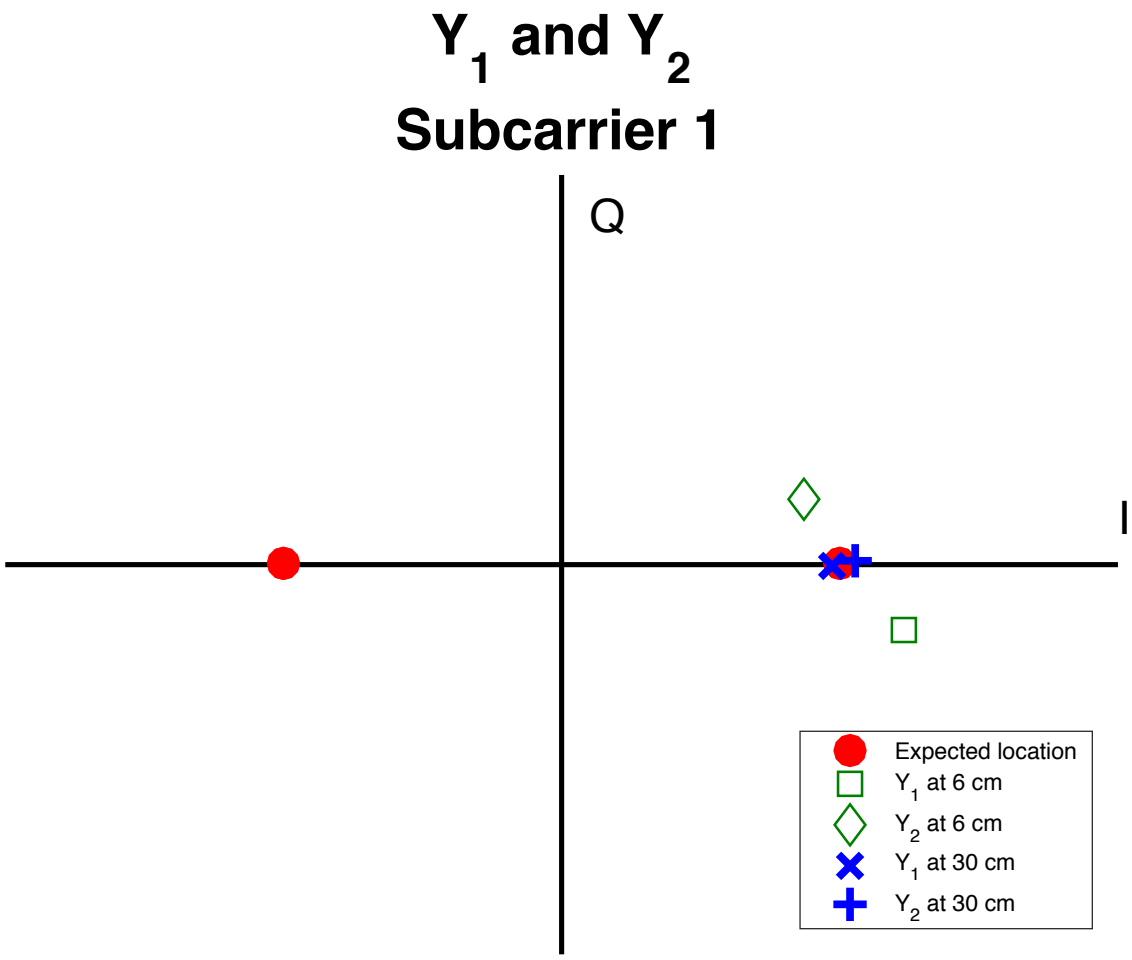

Figure 4.11: Difference between Y_1 and Y_2 for subcarrier 1. At long range (30 cm) the difference between Y_1 and Y_2 for subcarrier 1 is small. At close range (6 cm) the difference between Y_1 and Y_2 for subcarrier 1 is large.

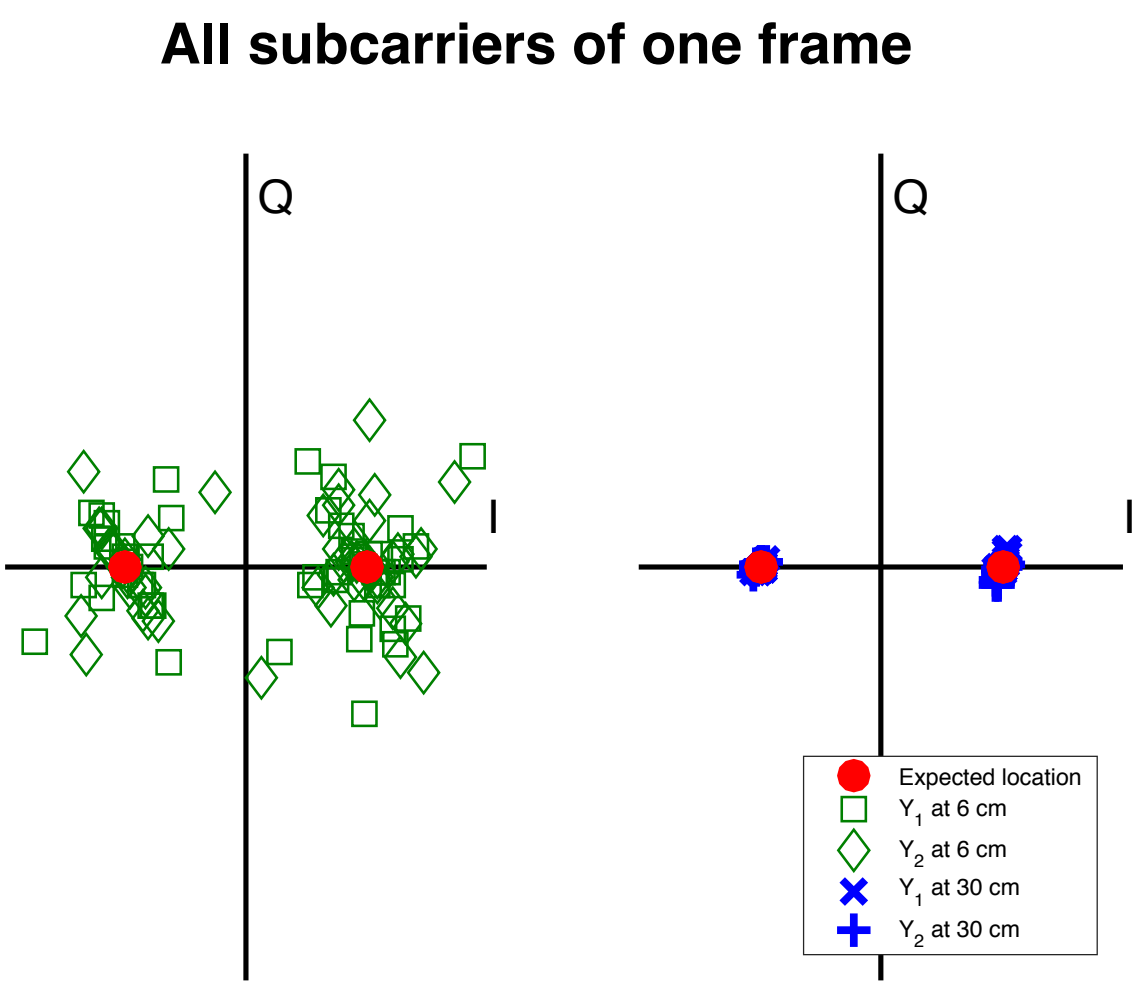

Figure 4.12: Preamble error for all subcarriers of one frame. Υ_1 and Υ_2 match at long (30 cm) range, but do not match at close (6 cm) range.

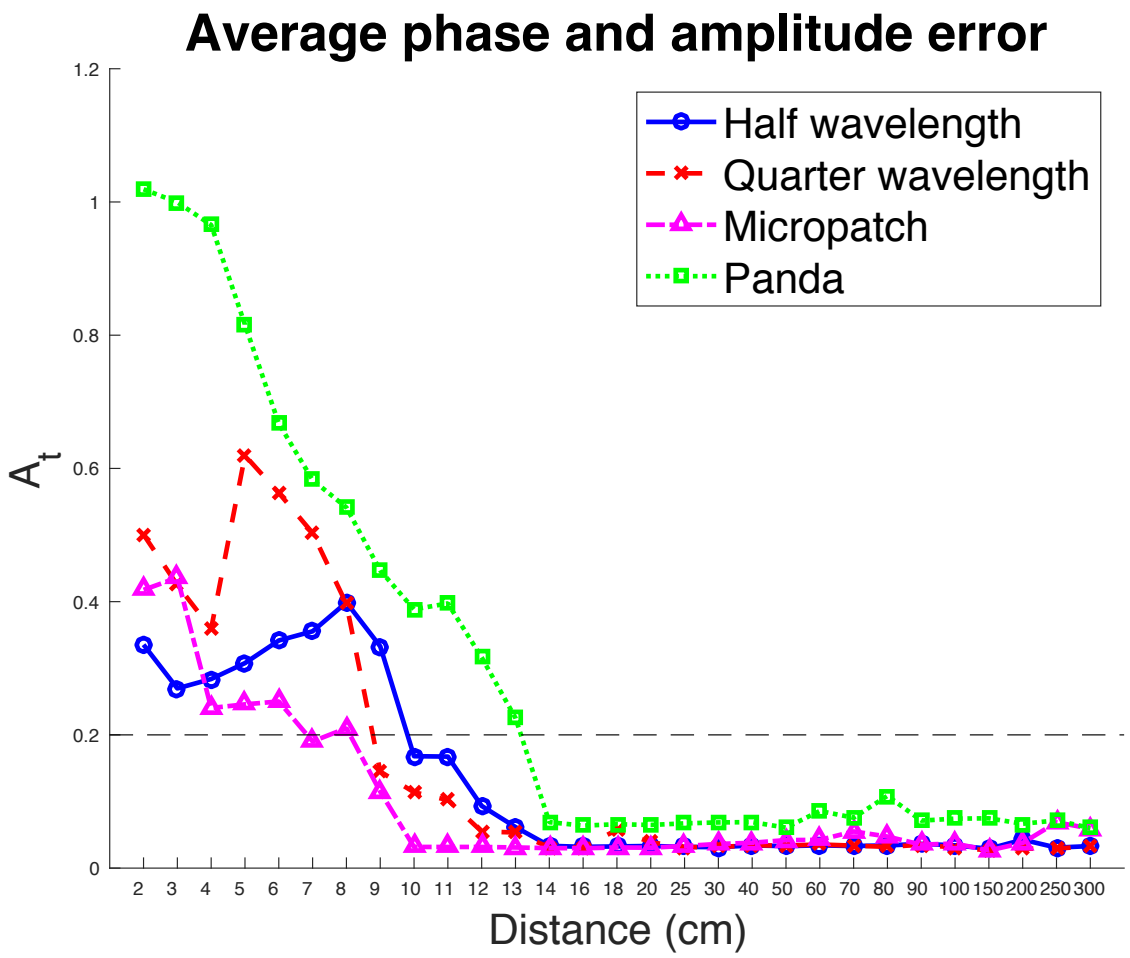

Figure 4.13: Average preamble errors by distance and antenna type using Equation (4.16). The average preamble error over 1,000 Wi-Fi frames is large at close range and small at long range for each antenna type.

all antenna types and suggests that a single-antenna target device can monitor the preamble error and declare proximity when the preamble error rises above a threshold.

4.5.3 Thresholds

We would like a single-antenna target device to be able to determine proximity with a transmitting device without help from another source. A simple way to make that proximity determination using the data in Figure 4.13 would be to set a threshold, τ,

where if the preamble error for a frame is greater than τ, the single antenna device declares proximity, otherwise it does not declare proximity.

If τ were set relatively high, say around 0.2 (indicated by the dashed line in Figure 4.13), then the single-antenna device would like not falsely declare proximity when the transmitter is far away because the preamble errors are never over the threshold for any transmitting antenna type at distances over 14 cm. If the target device uses only one frame to determine proximity, however, it could be the case that the particular frame happens to have have a low preamble error and the single-antenna device would not recognize proximity even though it should. Appendix B.1 shows that there is variability in the preamble errors, especially at close range.

This situation suggests that proximity detection with a single-antenna device may benefit from measuring the preamble error from multiple frames before declaring proximity. Instead of relying on the preamble error from a single frame, we can use Equation (4.16) to average the preamble error from multiple frames and then compare that average value with threshold τ.

To determine the likelihood of detecting proximity using the average preamble error from multiple frames, we created a Monte Carlo simulation where we randomly sampled n frames from the 1,000 Wi-Fi frames we captured at each distance between transmitter and receiver, and then calculated an average preamble error over those n frames for $n \in \{1, 2, 5, 10, 20\}$. We ran 1,000 trials at each distance and for each value of n. The idea is that the single-antenna target device declares proximity if the average of n frames is greater than τ.

In Figure 4.14 we show the likelihood of declaring proximity from 1,000 runs of the Monte Carlo simulation that randomly selected $n \in \{1, 2, 5, 10, 20\}$ Wi-Fi frames at each distance and then calculated the average preamble error over those frames. The simulation declared proximity if the average preamble error was greater than

$\tau = 0.2$. The results shown are the average over all antenna types. Appendix B.2 shows the results in detail for each antenna type and for several values of τ. We see there is a high likelihood of declaring proximity when the transmitter, regardless of antenna type, is within about 9 cm if the receiver uses more than one frame. In fact, we see that using only two frames performs much better than a single frame, consistently predicting proximity at short ranges, and never predicting proximity at ranges over 14 cm (although we see in Appendix B.2 that proximity is detected on one frame sent by the Panda adapter at 80 cm with $\tau = 0.15$ and $n = 2$, otherwise with τ or n at higher values, proximity is never detected at ranges greater than 14 cm). Because the single-antenna device is able to determine proximity with a small number of Wi-Fi frames, this proximity detection process is fast.

4.5.4 Future exploration

Our experiments suggest that the preamble errors caused by the near field of a transmitting antenna allow a single-antenna target device to reliably detect proximity at distances under 9 cm. We focused our testing on dipole and micropatch antennas because they are the most common antennas used in consumer devices. There are a myriad of other types of antennas including: horn, helical, parabolic dish, spiral, loop, spiral, Yagi, and bow tie to name a few. In future work we may examine how the near field affects these other types of antennas. We may also test with more off-the-shelf systems, other than the Panda adapter. We also used the default transmission power settings on the Wi-Fi devices and have not yet tested with non-standard power settings. We expect little difference in results. We believe, however, that our work here is a starting point and opens an important new area of research that warrants further investigation.

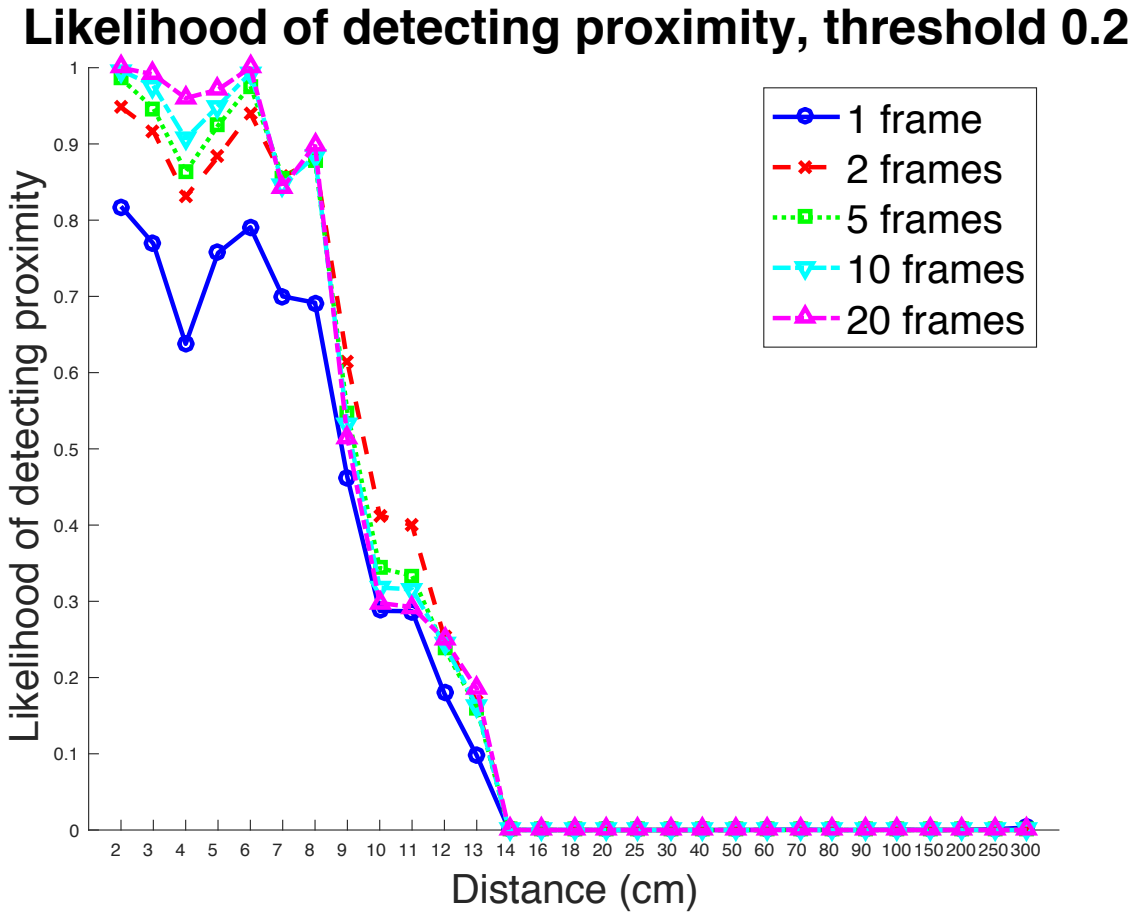

Figure 4.14: Likelihood of declaring proximity. This figure shows the likelihood of a single-antenna device declaring proximity in 1,000 Monte Carlo simulation runs at each distance where proximity was declared if the average preamble error for $n \in \{1, 2, 5, 10, 20\}$ randomly selected Wi-Fi frames was greater than $\tau = 0.2$. We see proximity declared with high probability at close range and never detected at ranges over 14 cm.

4.6 Security

Above we see that a single-antenna device is able to reliably detect proximity with a transmitting device if the single-antenna device is in the near field of the transmitter. Our experiments suggest that the single-antenna device can reliably detect proximity out to roughly 9 cm for dipole and micropatch antennas. These results assume, however, that the transmitter is sending properly formed Wi-Fi frames where T_1 in the Long Training Field matches T_2. It could be the case that a sophisticated adversary transmits a malformed Wi-Fi preamble where T_1 does not match T_2 in an attempt to trick the single-antenna device into falsely declaring proximity. That adversary might pre-compute a mismatched LTF and send those samples with a Software Defined Radio. In this case, a single-antenna device could potentially be tricked into declaring proximity when in fact the transmitter is far away. We propose two ways a single-antenna device might overcome such a sophisticated adversary.

4.6.1 Help from a trusted device

A single-antenna device might be able to overcome an adversary transmitting malformed preambles if the single-antenna device has a pre-existing trusted relationship with another device known to be far away (perhaps by measuring the preamble of signals from the trusted device) such as a Wi-Fi router. If the single-antenna device detects a preamble error greater than τ, it could ask the trusted device to confirm the trusted device sees a matching preamble. Provided the trusted device is located more than about 18 cm from the single-antenna device (e.g., two times the effective range of the preamble detection technique to rule out a legitimate transmitter positioned in between the single-antenna device and trusted device), the trusted device will see a matching preamble if the preamble is properly formed

and can inform the single-antenna device. If the preamble is malformed, both devices will see the high preamble error and with the trusted device's input, the single-antenna device can conclude an adversary sent the frames with malformed preambles.

4.6.2 Signal strength

In many use cases there will not be a trusted device with which the single-antenna device can confer. In those instances, the single-antenna device can examine the strength of the signal when it detects a high preamble error. While signal strength is a notoriously bad indicator of distance, because signal strength drops with the square of distance, a distant adversary will need to transmit a high power signal for the single-antenna device to receive it with the same strength as a signal from a legitimate device located a few centimeters away. To prevent the distant adversary from tricking the single-antenna device into believing malformed preambles are legitimate signals from a nearby device, the single-antenna device can measure the signal strength of frames with high preamble errors and reject frames with a signal strength below a threshold. We see next that even with a high-gain antenna, and an unlawfully high transmit power, an adversary's signal strength will be well below the signal strength of a legitimate device located nearby, making it possible for the single-antenna device to reject the weaker signals.

Assuming the adversary is in the far-field region, then signal strength will drop off as predicted by the well-known Friis equation [127]

$$P_r = P_t + G_t + G_r + 20\log\left(\frac{\lambda}{4\pi d}\right)^2 \tag{4.17}$$

where P_r is the power received in dBm, P_t is the power at the surface of the transmitting antenna in dBm, G_t and G_r are the gains of the transmitting and receiving

antennas, λ is the frequency of the signal, and d is the distance between the transmitting and receiving antennas.

Using Equation (4.17), we see that a signal from a legitimate device with $P_t = 27$ dBm, $G_t = 3$ dBi, and located 5 cm away would be received on a single-antenna device with a signal strength of slightly over 19 dBm, assuming $G_r = 3$ dBi on the single-antenna device. To estimate the amount of power a distant adversary could need to transmit to achieve the same signal strength at the single-antenna device, we can rewrite Equation (4.17) as

$$P_t = P_r - G_t - G_r - 20\log\left(\frac{\lambda}{4\pi d}\right)^2. \tag{4.18}$$

To match the signal strength of a nearby device, Equation (4.18) suggests that an adversary with a standard 3 dBi antenna located 1 m away would need to transmit a signal at a whopping 53 dBm – well above the capabilities of SDRs. The Ettus Research UBX-40 daughterboard for the N210 SDR, for example, transmits at roughly 20 dBm in the Wi-Fi frequency range. The popular HackRF One [47] tops out at 15 dBm in the Wi-Fi frequency bands. In fact, in the United States the Federal Communications Commission sets the legal limit for transmitters in the 2.4 GHz band at 30 dBm [31]. Nonetheless, we assume that an adversary does not stay within the FCC's limits and may have a more powerful transmitter. We acknowledge that there is theoretically no limit to how much power the adversary can transmit, but we assume a realistic adversary has some practical bounds on its transmitting capability. For discussion purposes, we consider an adversary capable of transmitting at 36 dBm – four times the limit set by the FCC and more than 32 times the power of the HackRF One. Furthermore, we assume the adversary may use a high-gain antenna to increase the strength of the signal received by the single-antenna device.

Commercially available high-gain Wi-Fi antennas are normally relatively large

Figure 4.15: Altelix high-gain antenna. This antenna provides 15 dBm gain, but is nearly one-half meter wide. Photo by Altelix LLC.

panel or parabolic dish antennas that provide up to approximately 20 dBi gain. For example, the Altelix 2.4 GHz parabolic dish antenna provides 15 dBi gain and is nearly one-half meter long [5]. While it is possible this antenna, or one like it, could be concealed inside of furniture or possibly behind a low signal-attenuating wall, in many cases a user would notice the presence of the one-half meter long antenna if it is within two meters from the single-antenna device. Additionally, the single-antenna device may be mobile, making it difficult to preposition a high-gain antenna such that it is focused on the mobile single-antenna device while data is transferred.

Using Equation (4.18), and assuming $G_r = 3$ dBi on the single-antenna device, in Table 4.1 we show the amount of power in dBm that an adversary would need to transmit to match the signal strength of a legitimate device transmitting from a few centimeters away. We assume the legitimate device transmits at 27 dBm (e.g., half

	Adversary distance (cm)				
G_t	*100*	*150*	*200*	*250*	*300*
0	56	60	62	64	66
3	53	57	59	61	63
6	50	54	56	58	60
9	47	51	53	55	57
12	44	48	50	52	54
15	41	45	47	49	51
18	38	42	44	46	48
21	35	39	41	43	45
24	32	36	38	40	42

Table 4.1: Required adversary transmit power. This table shows the amount of power in dBm that an adversary located a given distance from a single-antenna device using an antenna with gain G_t would need to transmit to have a signal arrive at the single-antenna device with the same signal strength as a legitimate device transmitting a 27 dBm signal from 5 cm away. Highlighted cells indicate configurations where the adversary would be able to match or exceed the legitimate signal if transmitting at four times the legal limit set by the FCC or below. We see that in most cases, even with the high transmit power and high-gain antenna, the adversary would need to achieve a higher signal strength than a legitimate nearby device transmitting at one-half the maximum set by the FCC.

the FCC limit) from 5 cm away. We assume the adversary is located one to three meters away and has a high gain antenna with G_t ranging up to 24 dBi. Highlighted cells indicate configurations where the adversary would be able to match or exceed the legitimate signal if transmitting at four times the legal limit set by the FCC (36 dBm) or below. We see that in most cases, even with the high transmit power and high-gain antenna, the adversary cannot achieve a higher signal strength than a legitimate nearby device transmitting at one-half the maximum level set by the FCC.

4.6.3 Raising the bar for an adversary

In most practical cases the single-antenna device can use signal strength to determine if a frame with a high preamble error was sent by a distant adversary, even if that adversary transmits at unlawfully high power and uses a high-gain antenna.

We consider our methods as a way of "raising the bar" that an adversary must overcome. It could be the case that the adversary has an extraordinarily high transmit power and extremely high-gain antenna. In those cases our method could fail, but our method raises the bar well above the capabilities of off-the-shelf hardware.

4.7 Related work

Exploiting the repeating nature of Wi-Fi OFDM preambles has not been widely explored in the literature. Work thus far has primarily focused on covertly embedding a small amount of data into the Wi-Fi frame and on device fingerprinting based on PHY layer attributes. Both of these approaches are substantially different from our methods and we examine each of them in this section. Additionally, because Near Field Communications (NFC) also uses the near field of a transmitting antenna, we briefly review that approach as well.

4.7.1 Embedding covert information in Wi-Fi frames

Classen et al. examined covert channels possible within standard Wi-Fi frames [21] to covertly embed information. In particular they analyzed methods to utilize the Short Training Field with Phase Shift Keying (STF PSK), Carrier Frequency Offset with Frequency Shift Keying (CFO FSK), using additional subcarriers while still conforming to Wi-Fi standards, and replacing portions of the Cyclic Prefix with covert data. In each of these cases the idea is to embed information in the Wi-Fi frame in a way that a standard receiver would not notice. While these techniques involve the PHY layer and often use the preamble, they do not exploit the repeating nature of the preamble for proximity detection.

In related work Rahbari and Krunz proposed a technique they call *P-modulation* to modulate the STF in a standards-compliant manner to include up to eight user-

chosen bits in a 20 MHz wide Wi-Fi frame [95]. These bits can be used to inform other devices of the transmitter's status, possibly eliminating the need for additional control frames. This technique, however, is different from our technique in that we use the repeating nature of the Long Training Field to establish proximity, not to use it to include a small number of indicator bits.

4.7.2 Device fingerprinting

There have been numerous studies on fingerprinting wireless devices. Many of these approaches focus on PHY layer imperfections resulting from manufacturing such as clock skews [7] and can determine subtle radiometric differences between devices. These radiometric differences can often lead to identification of device type, and can sometimes also lead to specific device identification [85, 135].

While identifying specific devices based on their radio characteristics can be useful in some use cases, these techniques do not provide an indication of proximity. It could be the case that a single-antenna device is able to distinguish between transmitters based on the (presumably non-spoofed) radiometric signatures of wireless signals it receives, but the single-antenna device does not know if the transmitter is a legitimate nearby device or a distant adversary – it simply knows transmissions came from the same device. Our techniques focus on exchanging data when devices are in close proximity.

Fingerprinting and our single-antenna proximity detection technique could, however, work together. One solution would be to use our proximity detection technique to know when a user brings a new device near a single-antenna device. The idea is that proximity determines the trustworthiness of the new device (whether the new device is compromised in some manner is out of scope for this work). In addition to detecting proximity, the single-antenna device could record radiometric details about the signal it receives from the new device. If the user later separates

the devices, the single-antenna device could examine the radiometric details of the signal to identify data sent by the new device.

The goal of re-identifying a known device could also be accomplished, however, if the single-antenna device shares a key with the new device when they are in close proximity, perhaps using the techniques described in Chapter 2. In future communications the new device could use the key to encrypt or sign messages intended for the single-antenna device, even if the two devices are now far apart. The single-antenna device could do the same for messages intended for the new device. We explore this concept further in Chapter 5.

4.7.3 Near Field Communications

Near Field Communications (NFC) is a short-range communication technique that uses two loop antennas to transfer data between devices located roughly 10 cm apart. These systems transmit in an ISM (Industrial, Scientific, Medical) frequency band at 13.56 MHz [23] and rely on magnetic induction to transfer data [100]. Induction occurs when a receiver is in the transmitter's near-field region as described in Section 4.3 and does not occur when the receiver is in the transmitter's far-field region. This near-field requirement implies that data can be securely transfer at short ranges (e.g., within R_1 from Equation (4.7)), without fear of interception by more distant eavesdroppers. Recent work, however, has shown that NFC communications can be read up to 2.4 m from the transmitter [139], making NFC a questionable choice for secure communications.

Despite the questionable security aspects of NFC, it is becoming increasingly popular on cell phones. This increased popularity, however, has not translated into wide-spread usage. Android Beam, for example, is a feature of the Android operation system initially deployed with version 4.0, Ice Cream Sandwich, that can bootstrap a Bluetooth connection by exchanging keys over NFC [44]. Android Beam

initially generated a great deal of excitement but has seen limited adaption. In fact ComputerWorld noted that "... despite the admirable marketing effort, Android Beam never quite worked particularly well ..." [63].

Additionally, because NFC requires specialized radios and antennas, it is not commonly found on IoT-type devices. Our approach, however, accomplishes some of the same tasks NFC was designed to accomplish, but does not require specialized radios or antennas. Instead, our approaches uses the in-band Wi-Fi radio commonly found on such devices.

4.8 Conclusion

In this chapter we have shown that a single-antenna device can reliably determine when it is in close proximity to a transmitting device by leveraging the repeating nature of Wi-Fi's preamble and the physical characteristics of signals in the transmitter's near-field region. Our experiments suggest that mismatches in the preamble caused by the near field of a transmitting antenna can allow a single-antenna target device to reliably detect proximity at distances under 9 cm. We focused our testing on dipole and micropatch antennas because they are the most common antennas used in consumer devices, but there are a myriad of other types of antennas we have not yet tested. In future work we may examine the near-field effects on these other types of antennas. We believe, however, that our work here is a starting point and opens an important new area of research that warrants further investigation.

5

JamFi: secure information transfer between nearby wireless devices

5.1 Introduction

As noted in Chapter 1, analysts predict *billions* of everyday devices will soon become "smart" with the addition of wireless communication capabilities. To help alleviate some of the difficulties in bringing new devices into an environment, in Chapter 2 we introduced Wanda. Wanda works well for imparting information onto the target devices, but requires the target to run a small amount of Wanda code to decode

the frames sent by the Wand. In this chapter we discuss another approach called *JamFi*, which transfers data to a nearby target device, but does not require the target to run additional software. In fact, the target device need not even be aware that the sender is using JamFi. The target simply receives Wi-Fi frames as it normally does with any other sender. Additionally, JamFi runs up to 18,000 times faster than Wanda.

JamFi uses jamming to thwart adversaries, while still allowing ad hoc communication between devices in close physical proximity. In this chapter we present a theoretical and practical evaluation of JamFi that exploits MIMO antennas and the Inverse Square Law to accomplish those goals. JamFi does not require any specialized hardware or sensors in the new devices, and does not require complex algorithms or complicated cryptography libraries. We find JamFi is able to facilitate secure in-band communication between two devices in close physical proximity (about 5 cm), even though they have never met nor shared a key.

As with Wanda, with JamFi we assume devices share data and control information among themselves, with new devices entering and exiting a particular environment frequently. People and the devices they wear or carry may encounter dozens, possibly hundreds, of other devices each day. Many of these devices encountered will be seen for the first time. Furthermore, some of the information the devices share may be privacy sensitive or have security implications. On top of that condition, many of these new devices will have limited or non-existent user interfaces. Manually configuring a large number of these interface-limited devices each day for ad hoc data exchange will not scale well. This situation implies that devices that have never met, nor shared a secret, but that are in physical proximity, must somehow have a way to securely communicate that requires minimal manual intervention and yet captures user intent.

As an illustration of this problem in the healthcare domain, imagine that a

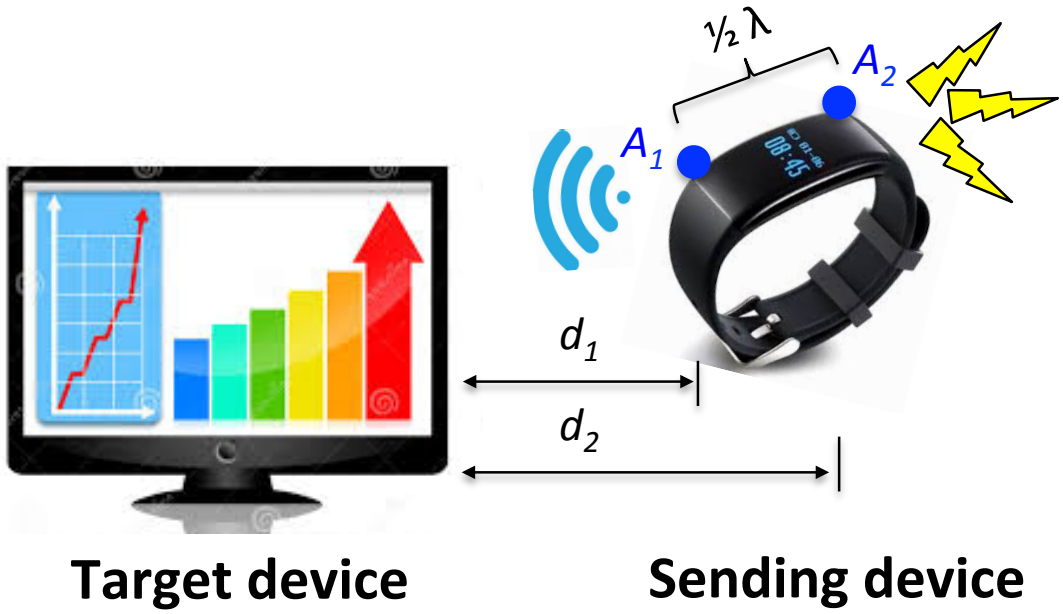

Figure 5.1: Sender and target. A multiple-antenna "sending" device uses antenna A_1 to send a data signal to a "target" device located at distance d_1, while antenna A_2 located d_2 from the target transmits barrage jamming.

patient has collected health-related data on a medical device and wants to show that data to a physician. The medical device has a limited user interface, or may have no display whatsoever, making it difficult or impossible for the patient and physician to view the information together. There might, however, be a display in the physician's exam room. The goal would be to get data securely from the patient's device to the physician's display so that the patient and physician can review it together as shown in Figure 5.1.

In this scenario the two devices have never met nor shared a key, but need to share sensitive medical information. If that data were revealed, others may learn something about the patient that the patient would prefer remain confidential. In addition to possible data leaks while setting up and using the link to the intended display, if the health device connects to the wrong display, the confidential data may be exposed.

There is a broad range of situations like this – where data must be securely

transferred from one device to another device that is in close physical proximity, or where it is important to ensure data is not accidentally exposed to more distant devices. In many such applications, physical proximity is a foundation for trust and user intent.

5.1.1 JamFi

JamFi uses jamming to cover information exchanged between a multiple-antenna "sending" device and another nearby "target" device in close physical proximity as shown in Figure 5.1. We use Wi-Fi to demonstrate JamFi, but the technique could be adapted for other protocols such as Bluetooth or Zigbee.

One antenna transmits data, another antenna jams

The sending device uses antenna A_1 to transmit a data signal to the target device located at distance d_1, while using a second antenna A_2 located d_2 from the target to broadcast barrage jamming (random noise).

Multiple antennas are becoming common in mobile devices, and in fact multiple antennas are required to take advantage of advanced features such as beam forming in Multiple-Input, Multiple-Output (MIMO) configurations of 802.11n and 802.11ac [56, 57]. Some devices, however, may simply be too small to support multiple antennas. In these cases, the small device can act as a JamFi target, but not a sender. We discuss this bi-directional communication scenario in Section 5.7.

Inverse-Square Law ensures and protects data transfer

Below we show that when a target device is in close physical proximity (about 5 cm) to a sending device with antennas separated by one-half wavelength, due to the Inverse-Square Law governing radio signal propagation, the data signal can be received with up to 50 times more strength than the jamming. This arrangement

ensures data is correctly received by nearby devices. We also show that more distant devices receive roughly equal data and jamming strength, making data recovery difficult.

Meant for ad hoc encounters (send keys otherwise)

JamFi is intended to transfer data under jamming cover when devices move into close proximity. Some situations such as sending medical data to the physician's display described above can be completed while the devices are close and JamFi's jamming can cover the one-time data transfer. If devices need to communicate many times or at long distances, JamFi can transfer a secret key which the two devices can then use to bootstrap a secure data-transfer session or long-term / long-distance secure relationship using traditional methods like TLS, with or without an AP, with or without the Internet.

No need for additional hardware, pre-shared secrets, or complex algorithms

Unlike many other approaches, JamFi does not require any specialized hardware in the target devices, any pre-shared secrets, any complex algorithms or complicated cryptography libraries, or any infrastructure such as access points or PKI authorities.

Jamming causes no additional network interference

The Wi-Fi specification requires devices to perform Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to avoid interfering with other devices operating in an environment [57]. JamFi follows this approach, transmitting both the data and jamming signals simultaneously during the time a normal Wi-Fi device would transmit only data. In this way JamFi does not create additional interference for other devices operating in the local area.

5.1.2 Assumptions

Throughout this chapter we make the following assumptions about the target device: 1) it has at least one radio antenna to receive wireless data, 2) it cannot be relied upon to have additional sensors such as cameras, microphones or accelerometers, and 3) it cannot be altered to add new hardware.

We assume the transmitting device has: 1) a radio compatible with that of the target device, 2) at least two antennas located approximately one-half wavelength apart, and 3) one antenna can send data while a second antenna transmits barrage jamming.

Finally, we assume either the target or the sender (or both) can be moved so that the devices can be placed in close physical proximity and that adversaries are located more than about 7 cm away.

5.1.3 Contributions

JamFi is a novel approach for securely transferring data between adjacent devices, even though the devices have never met, nor have any secrets been pre-shared. We make the following contributions in this thesis:

1. a consistent, fast, easy, and secure method to transfer any kind of information between commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the devices;

2. a theoretical analysis of jamming at close range to facilitate data transfer; and

3. an experimental evaluation using several different commercial off-the-shelf Wi-Fi receivers.

In the next section we review some background information useful to understanding JamFi's approach.

5.2 Radio signal propagation

JamFi's approach to overcoming jamming for devices in close physical proximity relies on the fact that radio waves attenuate proportionally with distance the signal travels. The nature of the signal, however, depends on the distance between transmitter and receiver. When a receiver is extremely close to a transmitter, the receiver is said to be in the *near field* of the transmitter. At longer range, the receiver is said to be in the *far field* (also called the *Fraunhofer* region).

As discussed in Chapter 4, the boundary between the radiating near-field region and far-field region for a finite-length antennas is estimated at distance R_2 from the antenna as follows [9]:

$$R_2 = \frac{2D^2}{\lambda} \tag{5.1}$$

where D is the length of the transmitting antenna plus the receiving antenna and λ is the signal wavelength. Equation (5.1) projects that the far field for a quarter-wavelength transmitting antenna at Wi-Fi's 2.4 GHz band begins at roughly 6.2 cm and is as short as 2.5 cm for Wi-Fi's 5 GHz band.[1]

5.2.1 Estimating signal power density at close range

Equation (5.1) gives an estimate for the boundary between the near and far field, but in reality the boundary is not sharply defined. Instead, the electric **E** and magnetic **H** fields generated by a transmitting antenna begin to align more fully so that they are orthogonal (perpendicular) to each other, transverse to the radial direction of propagation, as the signal moves substantially into the far field. Because the boundary is not sharp and JamFi is designed for communications between devices

[1] Some sources suggest the far field for short antennas (where $l \ll \lambda$) are best approximated by $R_2 = \frac{\lambda}{2\pi}$ which yields distances of 1.9 cm and 0.8 cm for the 2.4 and 5 GHz bands respectively.

separated by approximately the estimated distance from Equation (5.1), we cannot simply use the well-known Friis equation given in Chapter 2 to estimate signal strength at the receiver because the Friis equation is only valid in the far field.

Balanis, however, gives an approximation of the average power for a thin-wire (radius $r \ll l$) finite-length dipole that is valid everywhere except on the surface of the antenna: [9]

$$\mathbf{W}_{av} = \eta \frac{|I_0|^2}{8\pi^2 d^2} \left[\frac{\cos\left(\frac{kl}{2}\cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta} \right]^2 \tag{5.2}$$

where $\eta = 120\pi$ is the intrinsic impedance of free space, I_0 is the current applied to the transmitter, $k = 2\pi/\lambda$ is the wavenumber, l is the length of the transmitting antenna, d is the distance from the transmitting antenna, and θ is the vertical angle between the transmitter and receiver (below we assume $\theta = \pi/4$).

The d^2 term in the denominator of Equation (5.2) suggests that power density drops with the square of distance. If the distance d between transmitter and receiver is reduced by one-half, then the received power is increased by a factor of four. This relationship between distance and power is often referred to as the *Inverse Square Law*.

The relationship is particularly stark when a receiver is in close proximity to a transmitter. Figure 5.2 shows the expected average power density according to Equation (5.2), where transmitting antennas A_1 and A_2 are separated by a fixed distance of one-half wavelength, and a receiver is located d_1 cm away from A_1, such that $d_2 = d_1 + \lambda/2$. Antenna A_1 transmits a data signal while antenna A_2 transmits barrage jamming. Each antenna transmits at equal amplitude. In this figure we model a 24 dBm Wi-Fi signal transmitted on channel 1's center frequency of 2.412 GHz, which has wavelength $\lambda \approx 12.4$ cm.

We see in Figure 5.2 that when a receiver is very close to a transmitter, it

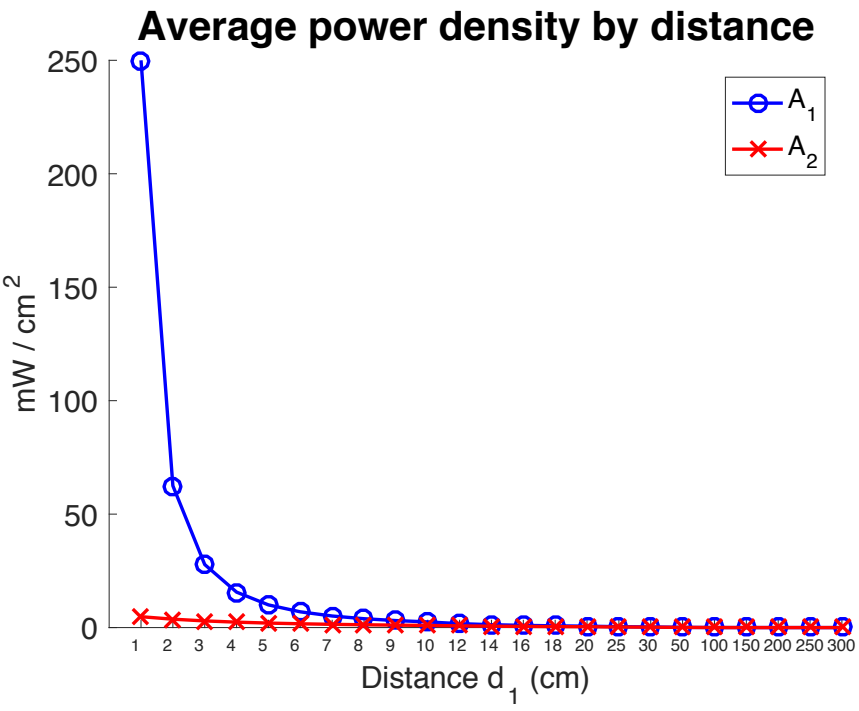

Figure 5.2: Expected power received from two transmitting antennas, each sending a 24 dBm signal, with antenna A_1 located at distance d_1 cm from the receiver and antenna A_2 located $d_2 = d_1 + \lambda/2$ from the receiver.

receives a significantly stronger signal than a signal from a transmitter located only one-half wavelength farther away. In this case, when antenna A_1 is located at $d_1 = 1$ cm, then $d_2 \approx 7.25$ cm, that is, 7.25 times farther than d_1. Because the power received is relative to the square of distance, even though both transmitting antennas are physically close to the receiver, the signal from A_1 is roughly 50 times stronger than the signal from antenna A_2. The difference in power between a signal sent from antenna A_1 and A_2 drops quickly as distance from the transmitter increases. When A_1 is more than about 7 cm away from the target, the received signal strength from each transmitter is virtually identical. A distant device therefore receives roughly equal-strength signals from each antenna.

When devices are in close proximity they enjoy a unique channel advantage over devices located farther away. That channel superiority vanishes quickly as devices move apart.

5.3 Signal errors

The performance of wireless digital communication systems carrying data in the presence of noise (both natural and intentional) has been well studied and has produced analytical models that predict the number of communication errors expected to occur given three factors: 1) data signal strength, 2) noise intensity, and 3) modulation scheme. We use those models to calculate the theoretical error rates given the physical arrangement of transmitter and receiver described in Section 5.2 where a target device is located near data antenna A_1 and one-half wavelength farther from jamming antenna A_2. In Section 5.5 we present the results from experiments using real, commercial-off-the-shelf (COTS) Wi-Fi devices as receivers.

5.3.1 Data signal strength and noise intensity

The relationship between a signal and noise is captured by the Signal-to-Noise Ratio (SNR) [38]:

$$\text{SNR} = \frac{P_r}{N_0 B} = \frac{E_s}{N_0 B T_s} = \frac{E_b}{N_0 B T_b} \tag{5.3}$$

where P_r is the received power of the data signal, N_0 is the power spectral density of the noise, B is the bandwidth, E_s is the energy per symbol, E_b is the energy per bit, T_s is the symbol time, and T_b is the bit time. For pulse-shaping systems such as Wi-Fi where $T_s = N/B$, Equation (5.3) simplifies to $\text{SNR} = E_s/(N_0 N)$ where N is the number of samples per symbol [38].

In the presence of barrage noise jamming, where the jammer interferes across the entire signal bandwidth (as opposed to tone jamming where noise is only transmitted on specific frequencies), the total power spectral density of the noise

becomes [64]:

$$N_t = N_0 + N_j \tag{5.4}$$

where N_t is the total noise power spectral density, N_0 is the power spectral density of any background noise, and N_j is the power spectral density of the barrage jamming. Accounting for noise provides the Signal-to-Interference-plus-Noise Ratio (SINR) where:

$$\text{SINR} = \frac{P_r}{(N_0 + N_j)B} = \frac{P_r}{N_t B}. \tag{5.5}$$

Equation (5.5) can be used to provide the SINR per symbol, γ_s [38]:

$$\gamma_s = \frac{P_r T_s}{N_t B T_s} = \frac{E_s}{N_t B T_s} = \frac{E_s}{N_t N}. \tag{5.6}$$

5.3.2 Modulation schemes

As discussed in Chapter 3, 802.11a/g/n/ac uses Orthogonal Frequency Division Multiplexing (OFDM) to send data symbols over several different subcarriers simultaneously, resulting in higher data rates than serial single-channel communications. Speed can be further enhanced with the type of modulation used on each subcarrier. In Wi-Fi the simplest modulation type is Binary Phase Shift Keying (BPSK), where each symbol represents one bit. More complex than BPSK, Quadrature Phase Shift Keying (QPSK) symbols represent two bits of information. Finally, Quadrature Amplitude Modulation (MQAM) is the most complex Wi-Fi modulation type where each symbol represents $\log_2(M)$ bits and M is 16, 64, or 256. More complex modulation schemes increase the data rate because each symbol represents more bits. Figure 5.3 shows these modulation types in a constellation diagram where a symbol,

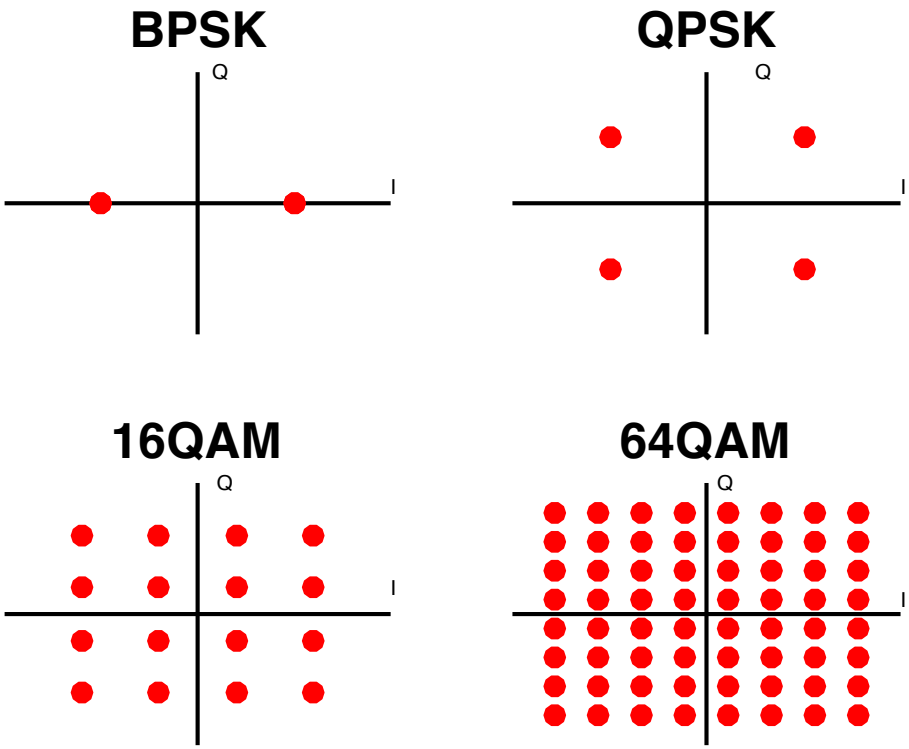

Figure 5.3: Wi-Fi constellation diagrams. Dots represent symbols in the complex plane. Except for BPSK, each symbol represents multiple bits. With QPSK each symbol represents two bits. With 16QAM each symbol represents four bits and with 64QAM each symbol represents six bits.

representing one or more bits, is shown as a dot in the complex plane.

To send a symbol, a transmitter selects the complex number on the constellation diagram representing the desired bit pattern, then modulates a cosine wave on a carrier frequency with the real component of the complex number, and also modulates a sine wave on the same carrier frequency with the imaginary component of the complex number. This modulation determines the phase and amplitude of the signal as discussed in Chapter 3. Assuming an Additive White Gaussian Noise (AWGN) channel, the receiver receives the signal as [38]:

$$\mathbf{y}[t] = \mathbf{x}[t] + \mathbf{n}[t] \tag{5.7}$$

where $\mathbf{y}[t]$ is the received signal, $\mathbf{x}[t]$ is the transmitted signal, and $\mathbf{n}[t]$ is the noise on the channel at time t.

The receiver then determines the nearest symbol to $\mathbf{y}[t]$ on the complex plane. Because $\mathbf{y}[t]$ includes noise, it may not fall exactly on a symbol, so the receiver chooses the closest symbol and infers that symbol is what the transmitter sent. Using a more complex modulation increases the susceptibility to noise because there are more possible symbols and smaller amounts of noise can cause the receiver to misinterpret a symbol corrupted by noise.

To compensate for noise, Wi-Fi uses convolutional coding to create redundancy by adding duplicate bits to each transmission. For example, 1/2 coding means that each bit is duplicated, resulting in 2 bits for every input bit. Coding redundancy reduces the overall data rate (e.g., 1/2 coding reduces the data rate by half), but can improve throughput by increasing reliability, especially in noisy environments.

A modulation type combined with a coding scheme is known as a Modulation Coding Scheme (MCS). 802.11g can use one of eight different schemes: BPSK 1/2, BPSK 3/4, QPSK 1/2, QPSK 3/4, 16QAM 1/2, 16QAM 3/4, 64QAM 2/3,

and 64QAM 3/4. More recent Wi-Fi versions specified in 802.11n and 802.11ac can use these modulation schemes as well as more complex modulation schemes. In Section 5.5, however, we see that more complex schemes cannot survive the jamming from antenna A_2, so we focus on these eight modulation coding schemes.

5.3.3 Energy per bit

The chosen MCS influences the energy per bit because a symbol may represent many bits, and each bit may be duplicated. Taking the energy per symbol from Equation (5.6) as a constant, the bit redundancy yields the SINR per bit, γ_b [38]:

$$\gamma_b \approx \frac{\gamma_s}{R_c \log_2 M} \tag{5.8}$$

where $\log_2 M$ is the number of bits per symbol and R_c is the coding rate (e.g., 1/2). There is a trade off in Equation (5.8): as the number of bits per symbol increases, the energy per data bit deceases, but as the coding scheme produces more redundant bits, the energy per data bit increases.

5.3.4 Estimating errors

Assuming an AWGN channel between sender and receiver, that all symbols in a modulation scheme are equally likely to be transmitted, and that Gray coding is used, so that one symbol error corresponds to one bit error (a conservative estimate, especially for complex modulation schemes), we can calculate the probability of a symbol error, P_s. Goldsmith [38] gives an excellent derivation of the error estimate equations shown in Table 5.1 where the Q function is

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx. \tag{5.9}$$

Modulation	M	P_s
BPSK	2	$Q\left(\sqrt{2\gamma_b}\right)$
QPSK	4	$2Q\left(\sqrt{\gamma_b}\right) - Q^2\left(\sqrt{\gamma_b}\right)$
16QAM	16	$4Q\left(\sqrt{\frac{4\gamma_b}{5}}\right)$
64QAM	64	$4Q\left(\sqrt{\frac{3\gamma_b}{7}}\right)$

Table 5.1: Probability of symbol error P_s by modulation type [38].

Table 5.1 indicates the probability of a symbol error depends on the signal's power relative to noise and the modulation type chosen. Assuming Gray coding, we can also estimate the probability of a bit error, P_b, as

$$P_b \approx \frac{P_s}{\log_2 M}. \tag{5.10}$$

Next we use these estimates to predict the ability of JamFi to successfully transmit data to nearby devices while denying more distant devices.

5.4 JamFi theoretical performance

Section 5.3 provided the mathematical underpinning to estimate JamFi's theoretical performance. In this section we use those equations to model JamFi's expected performance and in Section 5.5 we provide the results of experiments using COTS Wi-Fi devices. For all experiments and theoretical estimates we separate antennas A_1 and A_2 by one-half wavelength, with $d_2 = d_1 + \lambda/2$, and arbitrarily choose Wi-Fi channel 1. We model jamming phase and amplitude using a normal Gaussian distribution with zero mean and unit standard deviation, $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$.

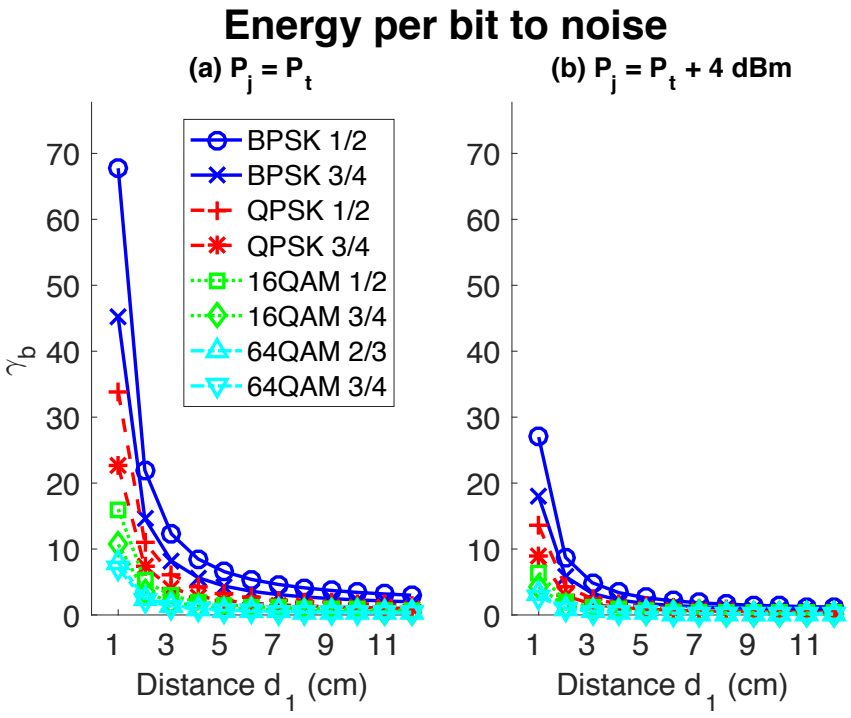

Figure 5.4: Energy per bit vs. noise at close range. At close range the energy per bit to noise is high, but drops off quickly with increasing distance.

Table 5.1 shows that the key to estimating errors, regardless of modulation scheme, is the ratio between the energy per bit and the energy in the noise. For JamFi, that ratio is primarily driven by two factors: 1) the geometry between the target device and the sending device's antennas, and 2) the ratio of transmit power of the two antennas (there is of course other noise in the environment; we model it at -92 dBm [51] but it typically has little impact on the error estimates).

5.4.1 Geometry

Geometry drives the ratio between signal and noise as shown in Figure 5.4a. We estimate the received power of the data signal, P_r, using Equation (5.2) at distance d_1. We estimate the noise power similarly, but using the transmit power P_j from jamming antenna A_2 at distance d_2. Assuming that antenna A_1 transmits data at the same strength that A_2 transmits jamming, then when the target device is located

near antenna A_1, the energy per bit will be up to 70 times stronger than the jamming signal. That ratio is maximized when the target device is located where d_1 is small and the antennas are aligned so that $d_2 = d_1 + \lambda/2$ as shown in Figure 5.1.

We assume JamFi devices can be adorned with an indicator such as an arrow to reveal how to align the devices. If the target is not well aligned relative to the transmit antennas, the ratio of signal strength to jamming will be reduced, resulting in increased noise relative to the signal. This works to JamFi's advantage because legitimate receivers can be placed near the transmit antennas and easily aligned to maximize d_2, leveraging the Inverse-Square Law, whereas more distant or less geometrically aligned devices will see a lower γ_b as shown in Figure 5.4a.

5.4.2 Jamming transmit power

Another factor that can affect the ratio of energy per symbol to noise is the transmit power of the data and jamming signals. We model the jamming transmit power as $P_j = P_t + \delta$ dBm, where $\delta \in \{0, 4\}$. In the first case the data and noise signal transmit power are equal; in the second case the jamming power is 4 dBm (2.5 times) higher than the data signal. In this latter case, shown in Figure 5.4b, JamFi relies even more heavily on the geometry and Inverse-Square Law to ensure the receiver is able to recover the data signal in the presence of more noise. If the legitimate target device is placed near the data antenna, the received data signal can still be almost 30 times stronger than the jamming signal.

Figure 5.5 plots the theoretical probability of a symbol error, P_s, using the equations in Table 5.1 and the energy per bit to noise, γ_b, when JamFi uses each of the eight modulation schemes and the target is aligned with the transmit antennas. Symbols transmitted with the simpler modulation types of BPSK and QPSK are more likely to be received without error than the more complex MQAM modulation

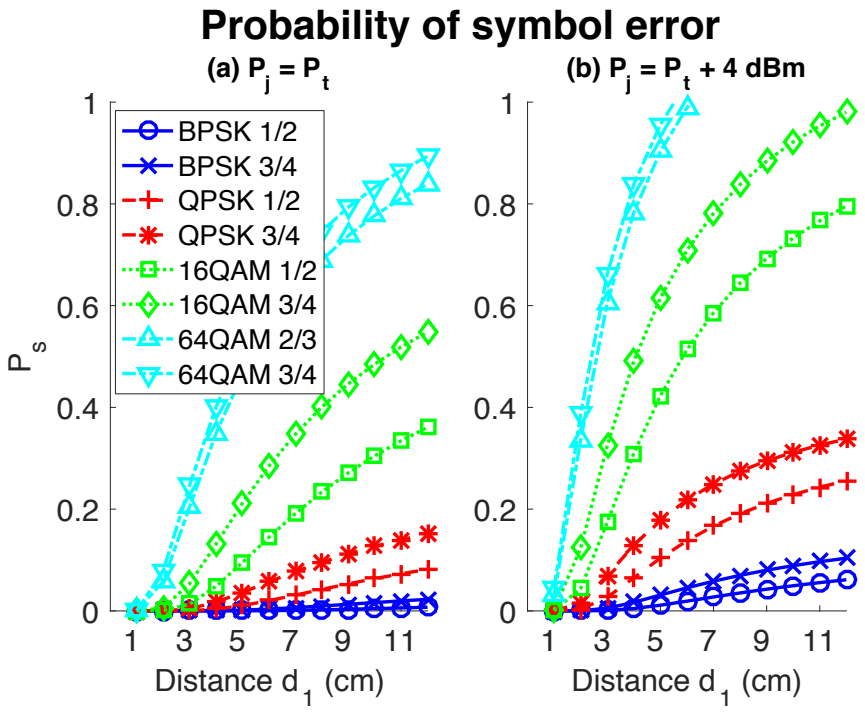

Figure 5.5: Probability of symbol error by MCS for devices in close proximity. Frames sent with simpler modulations (e.g., BPSK and QPSK) are more likely to be received without symbol errors than more complex modulations (e.g., 16QAM and 64QAM).

schemes.

Wi-Fi groups bits into frames for transmission. If a frame contains b bits, then the probability a frame is received without error, P_f, is:

$$P_f = (1 - P_s)^{b/\log_2 M}. \tag{5.11}$$

Figure 5.6 shows P_f, assuming the frame contains a modest-sized payload of $b = 1,024$ bits. We see that frames are likely to be received without error for BPSK and QPSK when the target is close (less than about 5 cm), and the probability of receiving a frame without error becomes extremely low at greater distances.

These estimates suggest that BPSK will likely be a good candidate to securely and reliably transfer data to a device in close physical proximity in the presence of jamming, while denying a more distant eavesdropper. This distance limitation may also help mitigate innocent errors where data would be unintentionally transferred

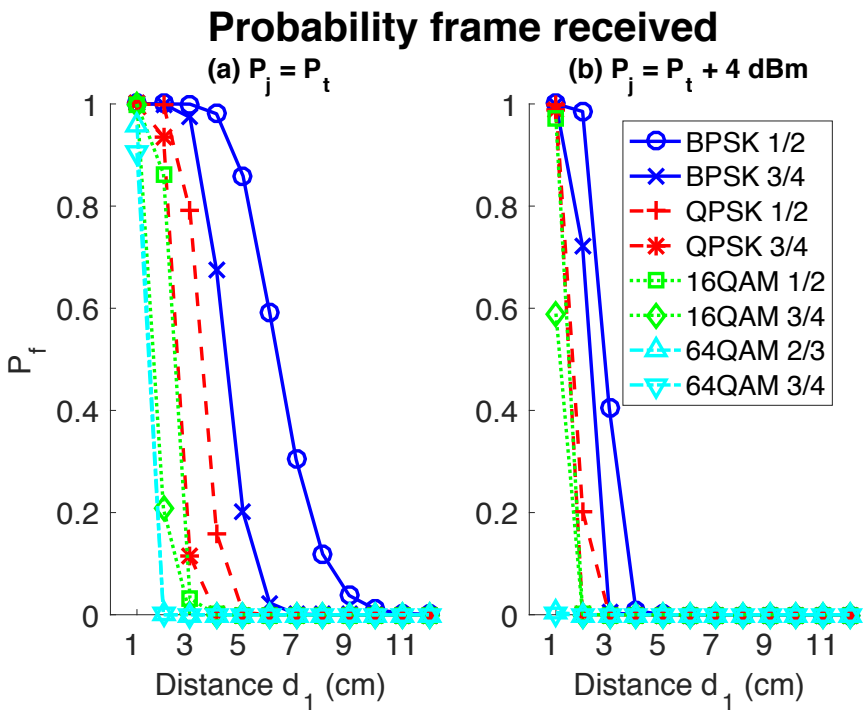

Figure 5.6: Probability a frame is received without error, given a 1,024 bit frame. BPSK 1/2 frames are likely to be received at close range without bit errors.

to a device located farther away from the multiple-antenna device.

5.4.3 Data transmit power

Another possible approach to securely transferring data between two nearby devices would be to lower the data transmit power and naively hope that a more distant eavesdropper would not able to receive the weak signal. Reducing the typical Wi-Fi transmit power of approximately 24 dBm to 4 dBm would reduce the transmit power by a factor of 100. Intuitively, that approach appears to be an easy way to reduce an adversary's range by a factor of 100. Because the signal attenuates with the square of distance, however, that is not the case. If we know the minimum signal strength at which a device can receive a signal, P_r, and assuming the device is in the far field, we can derive the maximum distance where a transmitted signal

is recoverable by re-writing the Equation (2.1) from Chapter 2 as:

$$d = \frac{\lambda}{4\pi\sqrt{\frac{P_r}{P_t G_t G_r}}} \tag{5.12}$$

where P_t is the transmit power, G_t and G_r are the gain of the transmitter and receiver respectively.

For example, if a system is able to recover a signal at $P_r = -73$ dBm [51], and no obstacles attenuate a 24 dBm signal, then by Equation (5.12), the received power will reach the device's minimum after the signal travels approximately 700 m. Dropping the transmit power to 4 dBm, however, yields a distance of roughly 70 m, only 10 times less than when transmitted at high power, not the 100 times reduction in range that one might have expected.

These calculations suggest that to avoid detection by an adversary located less than 1 m away, the transmit power will need to be reduced to an extremely low level. In theory, reducing the transmit power to -50 dBm would result in a -73 dBm received signal at 20 cm. While these calculations suggest the possibility that extremely low power could be helpful, there are two important considerations. First, an adversary can use a high-gain directional antenna to boost his receive range. A 9 dBi antenna would increase G_r, making the adversary's effective range roughly one-half meter. Second, environmental noise is likely to create significant issues for the legitimate target device at these levels.

Even though reducing transmit power alone does not provide assurance that a signal will not be recovered by a distant device, lowering transmit power still makes an eavesdropping adversary's task more difficult. In the next section we experiment with commercial-off-the-shelf Wi-Fi devices and 4 dBm data transmit power.

5.5 Evaluation

To test the effectiveness of COTS Wi-Fi devices to receive a signal in the presence of jamming, we tested four devices with electronics similar to those found in embedded devices: a Panda Ultra Wireless N USB Adapter [86], an Edimax EW-7811Un [28], an external Alfa Networks AWUS036H [4], and an internal Intel Ultimate N WiFi Link 5300 [58] connected to a Planar Inverted-F antenna.

On the transmit side, we used two calibrated Ettus Research N210 Universal Software Radio Peripheral (USRP) radios [30], each connected to a quarter-wavelength dipole antenna to simulate a multiple-antenna device. One USRP transmitted data using the GNU Radio 802.11a/g/p transceiver code developed by Bloessl [14], while the second USRP transmitted barrage jamming across the Wi-Fi 20 MHz channel during frame transmission. This arrangement allowed us to precisely control the signal strength and coordinate the timing of both the data and the jamming signals. The antennas were separated by one-half wavelength in keeping with Figure 5.1. We conducted all experiments on Wi-Fi channel 1, used 4 dBm to transmit power for data, and either 4 or 8 dBm to transmit jamming.

We first tested the ability of the four COTS devices to receive frames containing a 1,024-bit payload sent from the USRP without the presence of jamming. In this test we transmitted 1,000 Wi-Fi frames for each of the eight modulation schemes, with an interval of 100 ms between frames. To minimize outside interference, we tested these receivers in a remote indoor facility where there were no other Wi-Fi transmitters within at least 100 meters. We found each commercial device performed similarly; in this chapter we report the average results across all four devices. Appendix C provides details on each receiver's performance.

Figure 5.7 shows the average Frame Reception Ratio (FRR) – the number of frames received by the Wi-Fi device, divided by the number of frames transmitted,

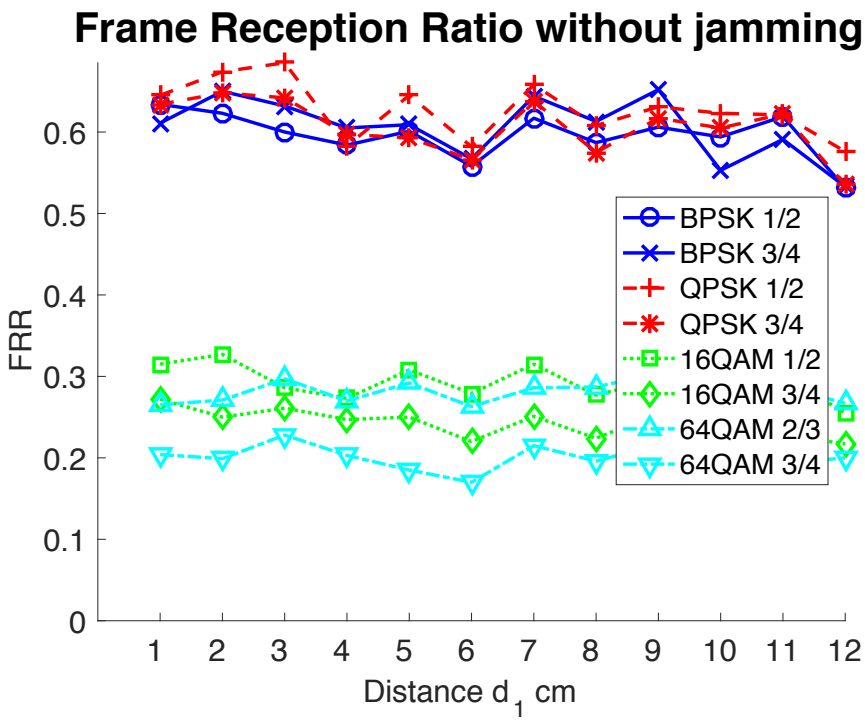

Figure 5.7: Frame Reception Ratio for 1,000 packets sent on each MCS. BPSK and QPSK frames were received with fewer errors than 16QAM and 64QAM frames.

for all four receivers where d_1 ranged from 1 to 12 cm. We see that the simpler modulation schemes were received with significantly higher probability than the more complex modulation schemes. Unsurprisingly, while theory suggests frame reception should have been near 100 percent when the devices are in such close proximity, real-life performance was well below predicted.

Next we tested the ability of the Wi-Fi devices to receive frames in the presence of jamming. Figure 5.8 shows the average FRR across all four devices when jamming signal strength was equal to the data signal strength (i.e., $P_j = P_t$), normalized to the FRR when no jamming was present (we refer to this ratio as NFRR). We see that BPSK 1/2, BPSK 3/4, and QPSK 1/2 performed relatively well when d_1 was less than 5 cm. More complex modulation schemes were received with low probability at close range, and all modulation schemes performed poorly at longer ranges. This is by design, as JamFi's purpose is to transfer data to nearby devices,

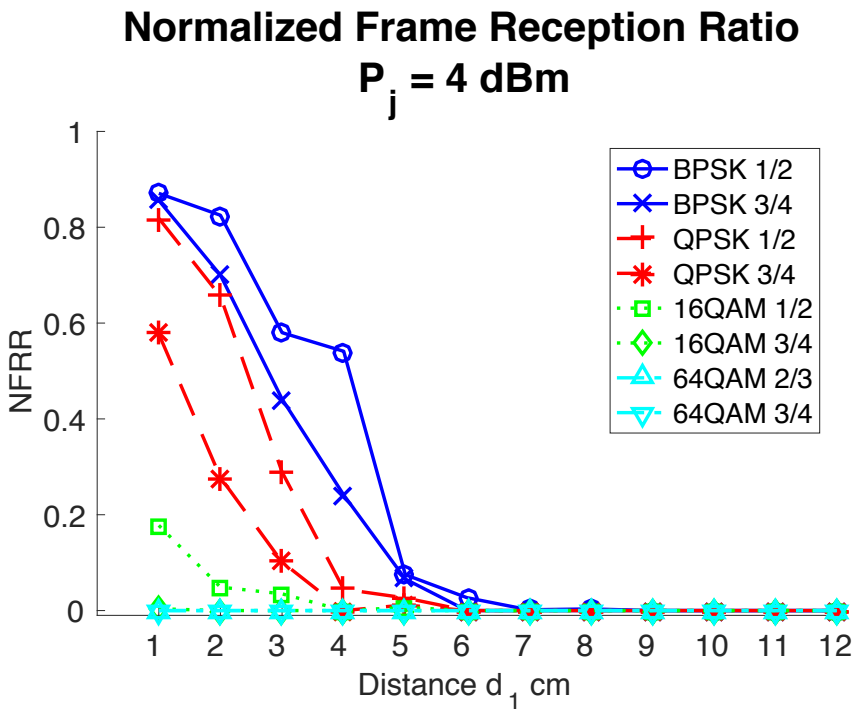

Figure 5.8: NFRR for 1,000 packets sent on each MCS when $P_j = 4$ dBm. BPSK 1/2 performs better than other modulation schemes when the jamming signal is the same strength as the data signal.

but not allow reception by more distant devices.

We then tested JamFi's ability to transfer data to nearby devices when the jamming signal was 2.5 times stronger than the data signal (i.e., $P_j = P_t + 4$ dBm). Figure 5.9 shows the results when d_1 ranged from 1 to 12 cm. We see that BPSK has some ability to transfer data in this environment up to 3 cm, but all other schemes and distances had virtually no reception. In all cases after 6 cm, no frames were received using any modulation scheme.

Unsurprisingly, we see that BPSK 1/2 performs much better in the presence of noise than other modulation schemes. In Figure 5.10 we compare BPSK 1/2 performance with the theoretical performance discussed in Section 5.4 and shown in Figure 5.6. We see that actual performance follows the theoretical performance, but lags somewhat. In the real world, data recovery on radio interfaces are never as

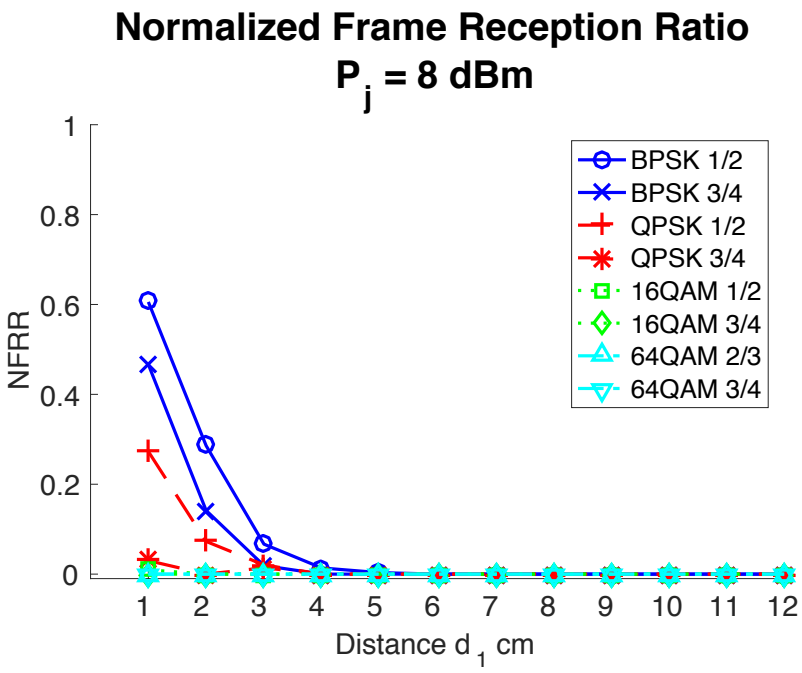

Figure 5.9: NFRR for 1,000 packets sent on each MCS when $P_j = 8$ dBm. BPSK 1/2 performs better than other modulations schemes when the jamming signal is 4 dBm higher than the data signal, but does not perform as well as when the jamming and data signals are of equal strength.

perfect as theory assumes. For example, if an interface misses the frame preamble, it will not attempt to decode the rest of the frame, whereas theory does not account for these types of issues. Despite these issues, theory elucidates the real world.

Finally, we examine the performance of BPSK 1/2 when the jamming signal was 2.5 times stronger than the data signal. Figure 5.11 also shows that the real world lags theory. We see that no data was successfully exchanged at ranges longer than 3 cm.

In summary, we see that JamFi was able to use BPSK 1/2 to provide communication in the presence of jamming when the data and jamming signals are of equal strength and the devices were closer than about 5 cm. Data could not be recovered by devices at longer ranges.

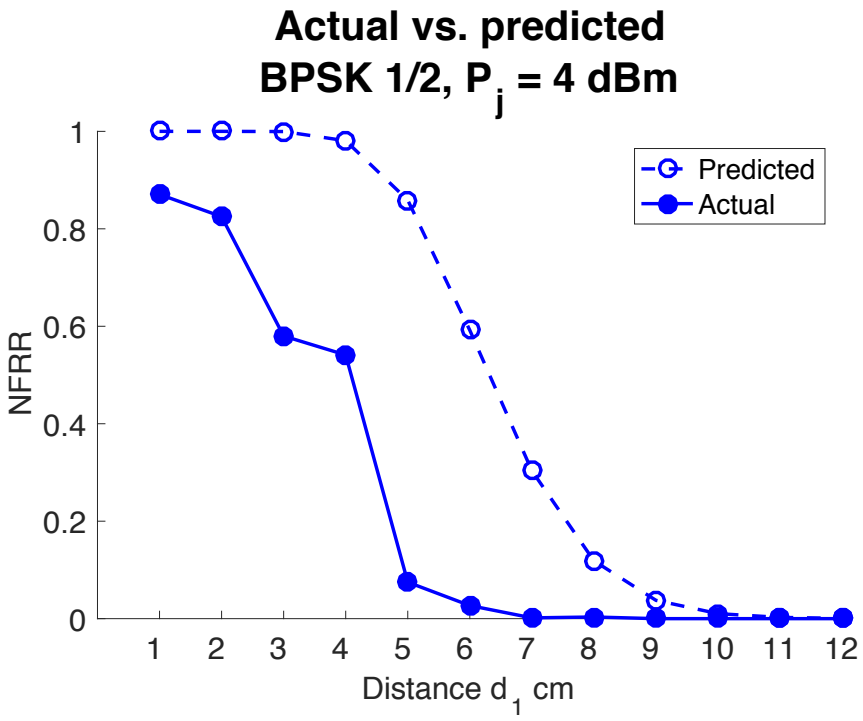

Figure 5.10: NFRR for 1,000 packets sent with BPSK 1/2 vs predicted when $P_j = 4$ dBm. Experimental results lagged somewhat behind the expected theoretical results, but were generally in line with predictions.

5.6 Security

In Section 5.5 we see that JamFi is able to provide communication while devices are in close physical proximity; in this section we discuss an adversary attempting to eavesdrop the data transfer or inject frames. We assume the adversary has full knowledge of how JamFi works and is able to employ more sophisticated equipment than COTS devices.

5.6.1 Eavesdropping

An adversary might attempt to eavesdrop on the data transfer between JamFi devices. Assuming the adversary is located more than about 7 cm away, the data and jamming signal strength the adversary receives will be roughly equal. An adversary might then attempt to separate the data and jamming signals with a

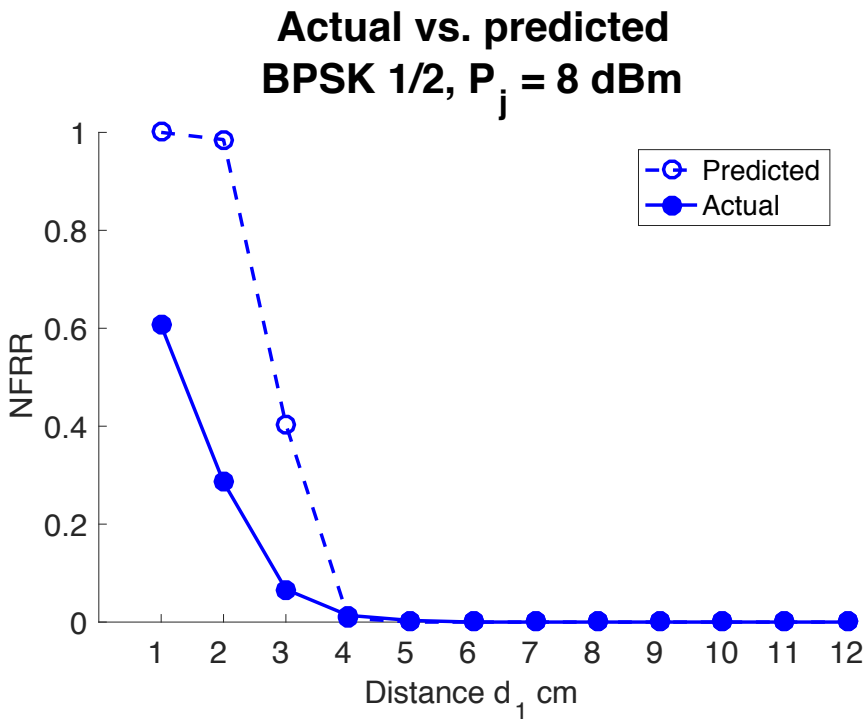

Figure 5.11: NFRR for 1,000 packets sent with BPSK 1/2 vs predicted when $P_j = 8$ dBm. Experimental results lagged somewhat behind the expected theoretical results, but were generally in line with predictions.

directional antenna or with signal processing and MIMO antennas.

Directional antennas

A directional antenna with a narrow main lobe pointed precisely at the data antenna, but excluding the jamming antenna, would allow the adversary to receive the data signal only. JamFi's antennas, however, are only one-half wavelength apart and because the main lobe expands with distance, the lobe will encompass both antennas if the adversary is located a reasonable distance away or is inline with JamFi's antennas. We evaluate this scenario in Section 2.7 and find that an adversary with a high-gain antenna located only 1 m away and bore-sighted on one of JamFi's antennas would need a one-half beam width of roughly four degrees – well below most real antennas or antenna arrays. Furthermore, because at least one of the JamFi's devices is mobile, the exact orientation and location of devices is difficult to

predict a priori.

Signal processing and MIMO antennas

Alternatively, an adversary might try sophisticated signal-processing techniques to separate the data from the jamming signal. Researchers have shown that, provided the adversary is located at a distance much greater than the separation between transmit antennas (in this case antenna A_1 and A_2), and the two transmit antennas are within one-half wavelength of each other (as they are with JamFi), the channel matrix has Rank 1 and the signals cannot be separated [127].

As an illustration, in Figure 5.12 we see a transmitter with antennas Tx_1 and Tx_2 separated by Δ_t and oriented with angle ϕ_t relative to a receiver with antennas Rx_1 and Rx_2 separated by Δ_r and oriented with angle ϕ_r relative to the transmitter. Tx_1 and Rx_1 are separated by distance d. If the transmitter has n_t antennas and the receiver has n_r antennas, then the channel between receive antenna r and transmit antenna t can be represented by following when the receiver is located a distance significantly greater than the spread between transmit antennas [127]:

$$h_{rt} = a\sqrt{n_r n_t}(e^{-j2\pi d/\lambda})(e^{j2\pi(t-1)\Delta_t cos\phi_t})(e^{-j2\pi(r-1)\Delta_r cos\phi_r}) \tag{5.13}$$

where a is an attenuation factor, Δ_r and Δ_t are the spreads between the receive and transmit antennas respectively, $r = 1 \ldots n_r$, and $t = 1 \ldots n_t$.

For the 2 x 2 MIMO arrangement depicted in Figure 5.12, the resulting channel using Equation (5.13) is

$$\mathbf{H} = a2e^{-\frac{j2\pi d}{\lambda}} \begin{bmatrix} 1 & e^{-j2\pi\Delta_r cos\phi_r} \\ e^{-j2\pi\Delta_t cos\phi_t} & e^{-j2\pi\Delta_r cos\phi_r}\ e^{-j2\pi\Delta_t cos\phi_t} \end{bmatrix}. \tag{5.14}$$

We see the second column is the same as the first column, except that the second

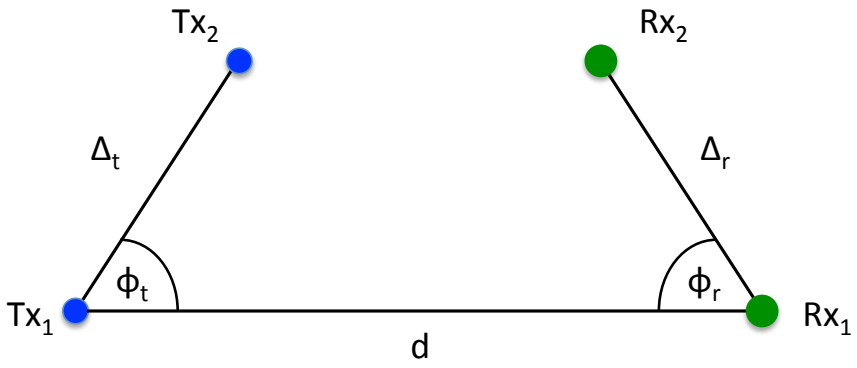

Figure 5.12: Channel Rank. A transmitter with antennas Tx_1 and Tx_2 separated by Δ_t and oriented with angle ϕ_t relative to a receiver with antennas Rx_1 and Rx_2 separated by Δ_r and oriented with angle ϕ_r relative to the transmitter. The channel between these two devices separated by distance d and with $\Delta_t < \lambda/2$ can be shown to be Rank 1 when $d \gg \Delta_t$. Adapted from Tse and Viswanath [127].

column is multiplied by a factor of $e^{-j2\pi\Delta_r cos\phi_r}$. This demonstrates the channel matrix **H** has Rank 1, holds even if the receiver has more than two antennas, and indicates signals cannot be separated by the receiver [127].

Given matrix **H** has Rank 1 when devices are far apart, one might wonder how spatial multiplexing (e.g., multiple simultaneous data streams) is possible with MIMO configurations if the transmit antennas are located near each other. Tse and Viswanath show multiple streams are not possible if the receiver is located at a distance that is significantly greater than the transmit antenna spread, the transmit antennas are separated by $\lambda/2$ or less, and there are no reflectors in the environment [127]. Those authors go on to show that multiple streams are possible in some cases if there is at least one reflector in the environment. As shown in Figure 5.13, a reflector in the environment can create virtual "relays" (denoted as points A and B) in the line of sight and reflected path that act as if the signal were transmitted from those locations [127]. These geographically separate relays may allow for spatial multiplexing if the transmitter is aware of the channel state and carefully beam forms its transmissions to bounce a stream off from the reflector. The result mimics a transmitter with antennas located at A and B. A receiver that

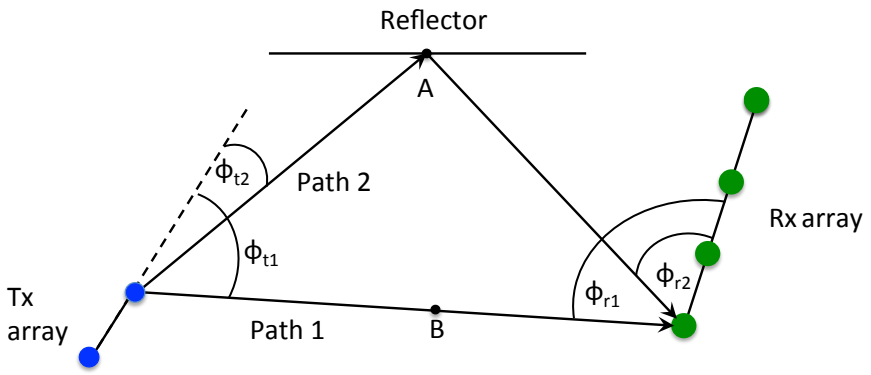

Figure 5.13: MIMO with reflected path. If there is a reflector in the environment, multiple spatial streams are possible between MIMO antennas, even if the transmit antennas are located near each other. Adapted from Tse and Viswanath [127].

has a good estimate of the channel can beam form similarly.

Even with a reflector in the environment, for the channel to have a Rank greater than one, both of the following conditions must be met [127]:

$$\begin{aligned} |\cos \phi_{t_2} - \cos \phi_{t_1}| &\geq 1/L_t \\ |\cos \phi_{r_2} - \cos \phi_{r_1}| &\geq 1/L_r \end{aligned} \tag{5.15}$$

where L_t and L_r are the lengths of the transmit and receive arrays normalized to the wavelength, and ϕ_{ti} and ϕ_{r_i} are the departure and arrival angles for path i as shown in Figure 5.13. With JamFi, the transmit array length is one-half wavelength, yielding $L_t = 1/2$. This length requires $|\cos \phi_{t_2} - \cos \phi_{t_1}| \geq 2$ for the channel matrix to have Rank greater than one. Because a cosine ranges from -1 to +1, the difference between two cosines is at most two. This fact, combined with the requirements of Equation (5.15) prompt many papers and books to declare signals from two antennas separated by $1/2$ wavelength or less to be inseparable [40, 126, 127].

Tippenhauer et al., however, exploited the fact that in the equations above a receiver must be located at a significantly greater distance than the transmit antenna spread to ensure the channel matrix has Rank 1. They showed that by using MIMO receive antennas at relatively close range, signals *can* be separated

in some cases [126]. Their analysis evaluated an adversary attempting to separate a 400 MHz data signal sent by one antenna using simple Frequency Shift Keying (FSK) from a jamming signal sent by a second antenna separated by 15 cm or more. They showed that it is theoretically possible to extract a signal with less than a 20% bit error rate at ranges around two meters. In practice, however, they found that (even with precise alignment of the antennas) multipath signals often defeated separation attempts. Furthermore, separating Wi-Fi's more complex modulation schemes at higher frequencies and smaller antenna spreads is more difficult than separating simple low-frequency FSK signals with large antenna separation. Those researchers did not demonstrate any capability to separate more complex Wi-Fi signals from jamming.

5.6.2 Frame injection

An adversary may attempt to inject his own frames while data is transferred between JamFi devices. In that case the adversary's signal would have to exceed the jamming signal strength. Because the jamming signal is located in close proximity to the receiving device, the Inverse-Square Law helps JamFi defend against such an attack. Even though JamFi transmits at 4 dBm, an adversary located only 2 m away using a 9 dBi omni-directional antenna would need to roughly double the maximum transmit power limits set by the U.S. Federal Communications Commission to exceed JamFi's signal strength.

5.7 Bi-directional communications

Above we discuss uni-directional communication – data moves from a multiple-antenna device to a target device that has one antenna. Here we discuss bi-directional communication.

If the target device also has two antennas, bi-directional communication is possible simply by reversing roles. If one device only has one antenna, however, we posit that secure bi-directional communications may still be possible. In this case, the single-antenna device can alert the multiple-antenna device that it has data to send and the multiple-antenna device initiates jamming on one antenna while listening on its other antenna. The single-antenna device can then monitor the noise floor. When the noise floor rises above a preset threshold, strong jamming is in place and it then transmits its data. In this way, a single-antenna device can bi-directionally communicate with a multiple-antenna device.

This approach, however, has some limitations. If the adversary is able to raise the noise floor above a threshold, the adversary may be able to trick the single-antenna device. The adversary could time his jamming such that after reaching the threshold on the single-antenna device, the adversary stops jamming just as the single-antenna device transmits. In this case the data is transmitted without jamming coverage and could be intercepted. To counter this attack, however, the single-antenna device could wait a random amount of time after the noise threshold is reached before sending the data. This way if the adversary stopped jamming, the single-antenna device would detect it and not transmit.

As noted in Section 5.6, an adversary would need to transmit a great deal of power to raise the noise floor to a level comparable to a nearby JamFi device. It is possible, however, that a formidable adversary with a highly directional antenna and extremely powerful transmitter may be able to raise the noise floor sufficiently.

5.8 Related work

JamFi securely transfers data among devices in close proximity using jamming. Other research has previously looked at accomplishing that goal using a variety of

techniques.

5.8.1 Cryptography

Many approaches involve cryptographic means, such as Diffie-Hellman key exchange. Despite impressive mathematics, Diffie-Hellman and related approaches have been shown to be vulnerable to Man-In-The-Middle attacks [2]. Additionally, because cryptographic methods such as Public Key Infrastructure are expensive to compute and rely on a trusted certificate authority, they may not be suitable for embedded devices common in IoT scenarios. JamFi does not require cryptography to accomplish secure data transfer.

5.8.2 Out-of-band communications

Out-of-band communication systems exchange a secret key between devices over a secondary communication channel that is impervious to observation and interference by an adversary. The devices then bootstrap a secure connection over the primary channel using the information exchanged over the secondary channel. Proposed secondary channels have included visual [71], audio [79], gesture [138], or secondary radios such as NFC or RFID. In each of these cases an additional sensor (light sensor, microphone, accelerometer, or second radio) is required. That required sensor or radio will not be present on many devices. Additionally, NFC has been shown to be vulnerable to interception at much longer distances than originally thought [139]. JamFi uses the in-band Wi-Fi radio and does not require additional sensors, radios, or complicated algorithms. As noted in Section 2.1 though, JamFi could be used to share a secret that can be used to bootstrap a long-term or long-distance session between devices.

Area	*Description*
Time synchronization	Create common reference point
Interference management	Overcome capture effect
Coexistence	Allow different protocols to use same spectrum
Cooperative diversity	Allow users to help each other transmit faster
Waking up a destination	Jam long enough to wake up sleeping nodes
Power control	Jam control channel so devices can determine required transmit power
Sense sharing	Share results of channel use
Channel reservation	Jam control channel to reserve data channel
Contention resolution	Access to a channel in a multi-user environment
Support of QoS	Prioritize some network traffic over others
Solving different problems	Hidden/exposed terminal, deafness, erroneous reservation
Collision detection	Eliminate hidden nodes by jamming
Statistics estimation	Estimate channel performance
Frame transmissions	Transmit data, control, attachment frames
Interrupting an activity	Stop transmissions already in progress
Physical security	Valid receivers get message/prevent injection

Table 5.2: Areas where friendly jamming has been suggested [3].

5.8.3 Jamming

Jamming has been well studied as means of covering in-band communication. While there are many uses for jamming, "friendly jamming" attempts to use jamming to accomplish a specific purpose such as secure data transfer. Recently, Al-Mefleh and Al-Kofahi published a comprehensive survey of friendly jamming. Their review covered 182 academic papers [3]. Table 5.2 summarizes the 16 areas they identified where where friendly jamming has been contemplated.

The most relevant area to our work listed in the survey is physical security. Like JamFi, the physical security papers deal with using jamming to ensure that only legitimate receivers are able to decode a message, or jamming to prevent injection of malicious frames. For example, Kuo et al. proposed a solution for imparting secret keys onto IoT-type devices. They suggest putting devices into a Faraday cage to exchange information and use a jammer to cover any RF leakage from the

cage [70]. This approach may work for small devices but is impractical for large ones. Several papers use cooperating relay nodes to jam and prevent eavesdroppers from decoding network traffic over large distances [8, 26, 37, 83]. Other researchers consider remote jamming, where unlike JamFi, the data source and jammer are located a large distance apart or have a pre-shared key [109, 110, 122, 129, 130, 140].

Another comprehensive friendly jamming survey by Huo et al. segregated approaches on three criteria: (1) non-self cooperative jamming (where additional devices aid the jamming) versus self-cooperative jamming (where the only legitimate devices are "Alice" and "Bob"), (2) uniform (omni-directional) versus directional jamming (jamming is beam formed to keep the receiver free of jamming), and (3) perfect versus imperfect knowledge of eavesdroppers CSI. JamFi does not rely on additional "helper" devices, does not rely on beam forming to keep the receiver away from the jamming, and does not rely on perfect eavesdropper CSI information (impossible with a passive adversary). None of the papers listed in their survey take JamFi's approach. All others rely on either additional helper devices, directional jamming, or perfect eavesdropper knowledge.

Two papers, however, consider jamming where a data source and jammer are in close proximity. In both papers a new jamming device is worn by a medical patient to protect implantable medical devices (IMDs) such as a pacemaker. In the first paper, a necklace-based jamming device called a *Shield* listens for communications specifically directed toward the implanted device and rapidly turns on the jammer to prevent outsiders from communicating with the implanted device [40]. In effect they are jamming the IMD to prevent it from receiving the outsider's communication. This approach differs from JamFi in that JamFi allows other devices to communicate with the target device. JamFi simply protects its own communications and does not render the target otherwise unreachable. Another difference between JamFi and the Shield project is that the Shield and the IMD are assumed to have a

pre-shared key so that the Shield can communicate with the IMD. JamFi does not have this requirement.

Another paper takes a related approach, but requires an ECG reading to create a common key between an implanted device and a jammer [134]. We do not intend for the MIMO device and target device to each have the capability to monitor a person's ECG.

5.8.4 Proximity

Like JamFi, other approaches to secure data transfer rely on close proximity between devices. *ProxiMate*, for instance uses fluctuations in television or FM radio broadcast signals to develop a common key between devices located within one-half wavelength of each other [73]. Separately, our own project called *Wanda* discussed in Chapter 2 exploits the difference in signal strength between two nearby antennas to securely transmit data to a target device [88]. Wanda, however, can only transmit *one bit* with each Wi-Fi packet, whereas JamFi can send a much larger data payload – 2,304 *bytes* in each Wi-Fi packet [57] – making it more than 18,000 times faster than Wanda. Furthermore, JamFi expands Wanda's security protection by adding jamming and by shortening the period of communication. Finally, Wanda requires the target to run a small amount of code to decode the frames sent by the Wand. JamFi does not have this additional code requirement. The target devices does not even need to be aware that the sender is using JamFi's technique. The target simply receives Wi-Fi frames as it would from any other sender.

5.9 Conclusion

As the number of deployed IoT devices grows, devices will increasingly encounter each other on an ad hoc basis and securely transferring data between them will

become an increasingly difficult problem. Manually entering secret keys on each device will become extraordinarily cumbersome. To help alleviate that problem, we developed and evaluated a system called JamFi that leverages MIMO antennas and the Inverse-Square Law to ensure wireless devices in close physical proximity can securely communicate while more distant devices cannot recover the information transmitted. Like Wanda, JamFi and works irrespective of device type or manufacturer and without additional hardware, complicated computation, or manual configuration. Unlike Wanda, however, JamFi does not require the target device to run special-purpose code to decode messages. The target simply receives JamFi frames as it would from any other device.

6

Future work

In this section we briefly describe some ideas for future work. One direction is to apply our Wanda and JamFi techniques to a wearable device such as a watch or bracelet. Another idea is to leverage some of our SNAP work to create a biometric bracelet. We discuss both directions next.

6.1 Wanda or JamFi bracelet

We propose building a wearable version of Wanda and JamFi in bracelet form. We would start with the simpler Wanda techniques because only one antenna transmits at a time and it would be easier to confirm which antenna sent a frame versus the

simultaneously transmitted signals from the data and jamming antennas in JamFi.

The bracelet would be intended to be worn by a single individual and would be less likely to be shared between individuals than other devices like smart phones [53]. Furthermore, the bracelet may be able to identify the wearer using biometric techniques such as those suggested by Cornelius et al. [22] or those described below in Section 6.2. If the identity of the wearer were known, the bracelet could then inform other nearby devices of the user's identity using the techniques described in Chapter 2 and 5.

A bracelet form factor could be useful to identify who is using a particular device. In a multi-person home, for example, several people may use a particular device such as a blood-pressure monitor. In some instances the user can be identified based on the nature of the interaction – existing weight scales can infer from a small population who is standing on the scale based on the weight of the person (e.g., Dad weights much more than Mom who weighs much more than child), but in many interactions simply using the device does not identify the user. The blood-pressure monitor may not be able to accurately determine who is using the device based on blood-pressure readings alone, but a bracelet could identify the user and communicate with the blood-pressure monitor using our techniques to annotate readings with the user's identity. This annotation may help reduce medical diagnosis errors based on improperly attributed health readings.

To work in bracelet form, two potential problems must be investigated. First, there must be enough antenna separation (approximately one-half wavelength) on the wrist to ensure enough difference in signal strength between the two antennas. Second, we must evaluate the signal-occluding properties of the wrist to ensure an adversary cannot receive a signal from only one antenna and compromise our protocols as described in Sections 2.7 and 5.6.

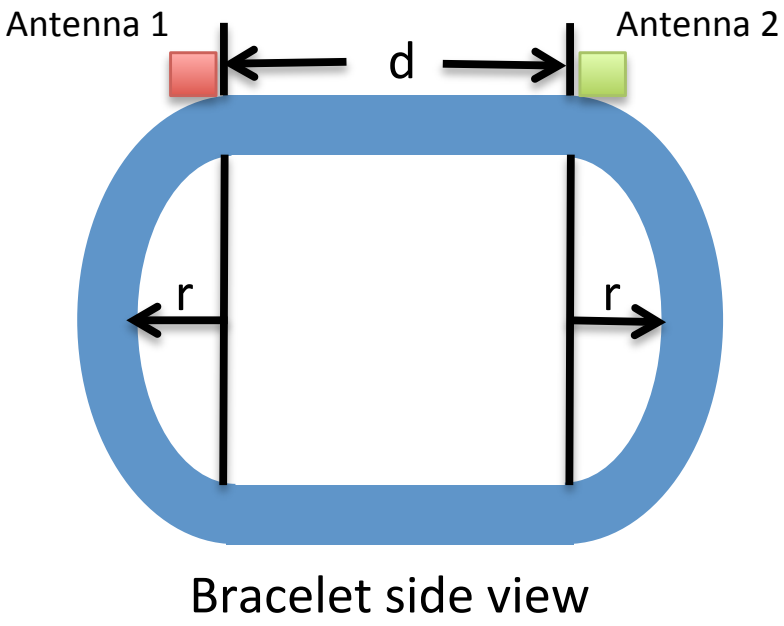

Figure 6.1: Bracelet with antennas stationed d cm apart.

6.1.1 Antenna separation

For our techniques to work in bracelet form, there must be sufficient separation between antennas so that the target device can differentiate between signals from each of the antennas, but not so far that a distant adversary can do the same. We imagine two antennas on top of the wrist as shown in Figure 6.1. Antennas A_1 and A_2 act the same as in Wanda, or antenna A_1 could be the data antenna and antenna A_2 could be the jamming antenna in JamFi. We would like the antenna to be separated by roughly 6.2 cm if using Wi-Fi's 2.4 GHz band and roughly 3.1 cm if using Wi-Fi's 5 GHz band.

If the bracelet's antennas are situated as shown in Figure 6.1, the maximum distance between the antennas would be relative to the size of the wearer's wrist and is $d + 2r$ if the antennas do not extend beyond the wrist. The average circumference of an adult woman's wrist is 19.0 cm, and the average circumference of an adult male's wrist is 20.3 cm [131], but wrist sizes can vary significantly as shown in

Size	*Circumference Women*	*Antenna Spread*	*Circumference Men*	*Antenna Spread*
Small	16.5	6.4	17.8	6.9
Medium	19.0	7.3	20.3	7.9
Large	21.6	8.4	22.9	8.9
X-Large	24.1	9.3	25.4	9.8

Table 6.1: Wrist circumference and estimated antenna spread by gender. Numeric values in cm [131].

Table 6.1.

Using these circumferences, we can estimate the antenna spread on each wrist size. Assuming the bracelet has a semi-circular shape on the left and right side, as shown in Figure 6.1, the total circumference C of the wrist is:

$$C = 2\pi r + 2d \tag{6.1}$$

If we further assume a ratio R that captures the relationship between r and d, then we can solve for d in Equation (6.1) by substituting Rd for r as:

$$\begin{aligned} C &= 2\pi Rd + 2d \\ &= d(2\pi R + 2) \\ d &= C/(2\pi R + 2). \end{aligned} \tag{6.2}$$

Using R equal to 0.5 as a rough estimate (by looking at our own wrist) of the ratio of the r to d, we can estimate the antenna spreads. Without extending beyond the wrist, the maximum antenna spread estimates for both men and women are given in Table 6.1. We see that even for small wrist sizes, there is a relatively large expected spread between the antennas – larger than the 6.2 cm or 3.1 cm needed. The antenna spreads given in Table 6.1 are the estimated maximum spread with the antennas mounted on top of the wrist.

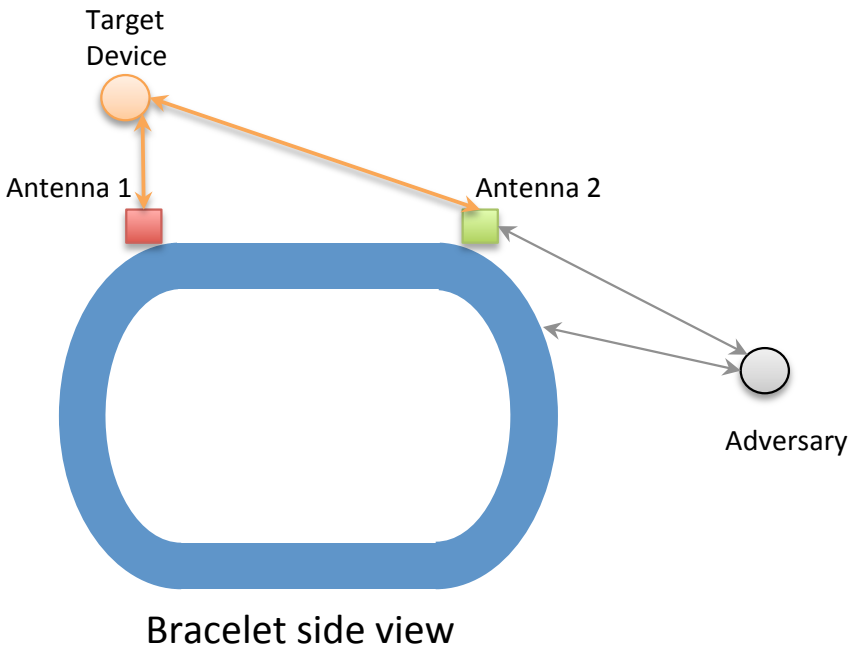

Figure 6.2: An adversary located where one antenna is visible, but the other is occluded, may be able to decode data intended for a legitimate device.

6.1.2 Wrist occlusion

One potential problem with a bracelet is the possibility that an adversary might be located in a position where he could capture frames from one of the bracelet's antennas, but not both, or see a strong difference in RSSI because the wrist might occlude one of the antenna's signals. Figure 6.2 illustrates an adversary located in such a position. In that case, as discussed in Sections 2.7 and 5.6, the adversary may be able to decode the data.

A solution to this problem may be to add secondary transmitting antennas on the opposite side of the bracelet as shown in Figure 6.3. By adding antennas that transmit the same data as the primary antenna (e.g., Antenna 1 and Antenna 1' each transmit the same data at the same time, Antenna 2 and Antenna 2' behave similarly), we may be able to solve the problem where an adversary can see the transmissions from one antenna but not the other. In this case, every location

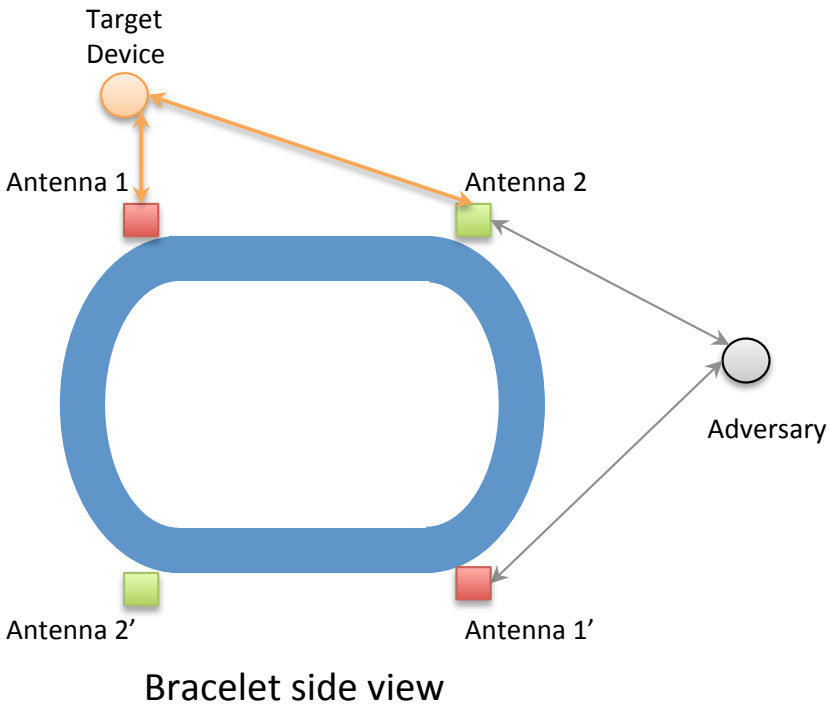

Figure 6.3: Adding antennas on the opposite side of the bracelet may solve the problem where an adversary sees only one of the original two antennas.

around the wrist should have an unobstructed view of at least two antennas.

This area of investigation is predicated on the idea that the wrist effectively blocks RF transmission, but we have not explored this yet. We will also need to ensure that an adversary who can see three antennas (e.g., antenna 1 and 1' plus antenna 2, or antenna 2 and 2' plus antenna 1) gets the same amplitude signal from two antennas of the same type as it does from the single antenna of the opposite type. If there were a signal amplitude difference, the adversary would be able to identify which antenna sent each frame and could then decode the message.

6.2 Wi-Fi as biometric

Channel State Information has been recently used in a number of interesting ways. For example, Xin et al. recently explored using CSI to biometrically identify individ-

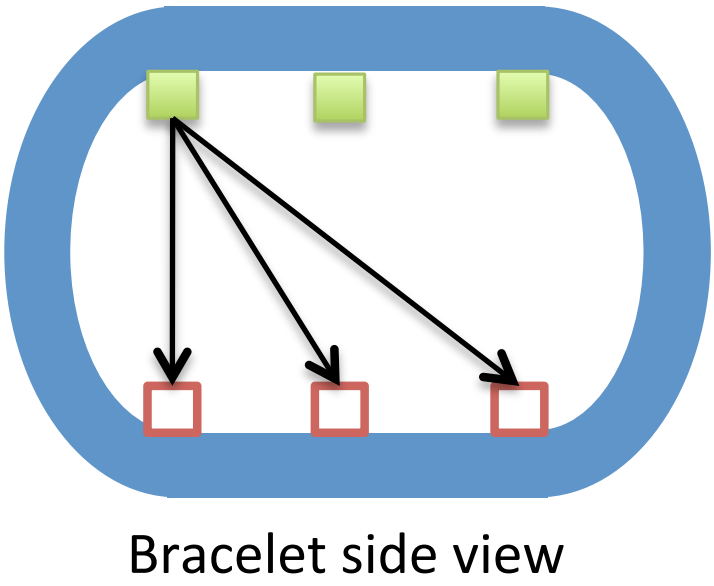

Figure 6.4: Biometric identification from bracelet RF. Multiple transmitting antennas (top) send a signal through the wrist where it is received by multiple receiving antennas (bottom). The physiological make up of the individual's wrist alters the signal while in flight, resulting in a possibly unique CSI reading.

uals [133]. The idea is that each individual is made up of different amounts of blood, fat, bone, and muscle. These physiological factors interact with a radio signal and may result in different CSI measurements for each individual. Xin's work was done in a controlled setting, but we hypothesize that we could use related techniques to identify a person wearing a bracelet.

We imagine a bracelet, similar to the one described above, but with Wi-Fi transmitters along one side of the bracelet band and Wi-Fi receivers located on the opposite side of the band as shown in Figure 6.4. In this configuration the Wi-Fi antennas function as normal Wi-Fi antennas, sending and receiving data as done on the Apple iWatch [6]. Those Wi-Fi signals are also received and analyzed as a biometric, after the signal has traveled through the wrist, by the antennas located on the opposite side of the band. Because each person's wrist has a unique physiology, the CSI of the signal should be different per individual.

Using MIMO antennas could give multiple paths through the wrist as shown in Figure 6.4. Here we see three transmitting antennas sending a signal to three receiving antennas. Because the signal from each transmitting antenna is received by all three receiving antennas, there are a total of nine paths through the wrist in this configuration. Some of these paths will pass through or reflect off from blood, fat, bone, and muscle differently than other paths. We hypothesize that, for reasons similar to the work by Cornelius [22], the result should be a unique CSI signature per individual. Unlike Cornelius, however, we do not require constant contact with the skin, which is a limitation of that work.

Uniquely identifying an individual. Multiple paths through the wrist may allow a system to uniquely identify the wearer better than the single path that would result from a single transmitter and a single receiver. This information may not, however, be accurate enough to identify a person from a large group, but may be enough to identify a person from a small group such as a family.

Placement consistency. One issue that would need to be overcome is placement consistency on the wrist. While wrists are relatively small, it is likely that there will be some inconsistency in the physical location of the antennas each time the bracelet is donned, and the position will likely change as the bracelet shifts position during normal wear.

Near-field effects. Because wrists are relatively small, the antennas will be close together. As noted in Chapter 4, near-field effects may cause problems with CSI readings. In this case, CSI measurements will need to account for rapid changes due to the rotating magnetic and electric fields. Pilot symbols, however, may be able to help by providing a known transmit phase and amplitude.

Attenuation. Finally, this approach assumes a Wi-Fi signal can pass through the wrist. There is some medical data around signal penetration of human tissue, but it is not clear if a Wi-Fi signal will attenuate to a level too low to be detected after passing through the wrist. We conducted some preliminary tests, and were able to receive a signal with micropatch antennas mounted on a bracelet similar to Figure 6.4. It is unclear, however, if the radio signal passed through the wrist or coupled around the wrist. More experimentation is needed.

7

Summary

Like many others, we foresee IoT-type connected devices moving into our daily lives in large numbers. We envision that these devices will gather and share data about us, and some of that data will have privacy or security implications. Securing billions of these devices and the sensitive information they will hold will be difficult. Devices will need to be configured for their local environment, must be managed to stay properly configured as the environment changes, and because they may encounter dozens of unfamiliar devices each day, devices will need a way to securely communicate. We have developed three specific and related solutions to help deal with the impending arrival of billions of these IoT devices.

7.1 Approach

At a high level, we adopt the approach advocated by Balfantz and others [10] where we rely on users to *demonstratively identify* devices with which to interact. In our case, we rely on users to expressly bring specific devices in close physical proximity. Once in proximity, we rely on *location-limited channels* to transfer data between proximate devices. We do not require devices to have already been introduced nor even that they mutually trust a common third party such as a certificate authority. Our techniques can be used to impart configuration information (such as the SSID and password of a Wi-Fi access point) onto a device, can be used to ensure the information came from a nearby device and not a distant imposter, and can facilitate secure short-range communications.

7.2 Wanda

Our first solution, called *Wanda*, is designed to help ease the burden of introducing devices into new environments. It uses a two-antenna hardware device called the Wand and a novel radio signal strength communication technique to impart data onto new devices. In addition to academic acceptance at INFOCOM [88] and a demo at MobiSys [87], Wanda's approach has been favorably reported in the popular press, receiving coverage in over 200 newspapers, radio, and television stations including: the New York Times, Washington Post, NPR, and an invitation to appear on the TBS television show, *America's Greatest Makers.*

Wanda makes the following five contributions:

1. a consistent, fast, easy, and secure method to impart any kind of information onto commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the device;

2. protocols for imparting information onto new devices (such as a Wi-Fi SSID and password), introducing two devices so they can establish a secure and user-intended connection, and imparting cloud identity and credentials into a new device;

3. a prototype implementation and experimental evaluation;

4. a security analysis of the system; and

5. the results of a 36-person user study.

While Wanda has been favorably received, it has specific short-comings that we address in our other work. With Wanda, the single-antenna target device has no way on its own to determine if it is in close physical proximity with the Wand. As discussed in Chapter 2, the Wand can use its two antennas to independently measure signal strength to determine proximity with the target device. A single-antenna target device, however, cannot use the same two-antenna technique. Because radio waves are invisible, it could be the case that an adversary located more than about 9 cm away from the target device modulates its power in an attempt to mimic the Wand and trick the single-antenna target into accepting a malicious payload. Wanda defends against this attack by listening for rogue *Message* frames that the Wand did not send and calling a halt to the data transfer if the Wand detects one of those frames. It would better, however, if the target device had its own way of determining whether a signal originated nearby, rather than relying on the Wand's vigilance.

7.3 SNAP

Our second solution addresses the proximity detection problem for single-antenna devices. Called *SNAP*: SiNgle Antenna Proximity, our method uses the repeating

nature of Wi-Fi physical layer preambles and changes caused by a signal in the near-field region of a transmitter to reliably determine when devices are closer than roughly 9 cm. As discussed in Chapter 4, other researchers have used the preamble to hide a few indicator bits, but our work is the first to use the preamble and the near field to determine proximity. With SNAP, the single-antenna device no longer needs to rely on the Wand for proximity detection and can reject malicious frames on its own.

We focused our testing on dipole and micropatch antennas because they are the most common antennas used in consumer electronics, but there are a myriad of other types of antennas we have not yet tested. We believe, however, that our work outlined in this thesis is a starting point and opens an important new area of research that warrants further investigation.

SNAP makes the following contributions:

1. a novel method for a single-antenna device to quickly determine when it is in close proximity with a transmitting device;

2. a reference Wi-Fi implementation that performs the same frame decoding steps *any* Wi-Fi device must perform; and

3. an experimental evaluation of the technique using several popular types of antennas.

7.4 JamFi

Our third solution, *JamFi*, uses friendly jamming with MIMO antennas and leverages the Inverse-Square Law to ensure data is transferred securely between nearby devices. Wanda was also aimed at secure data transfer between nearby devices, but requires the target device to run a small amount of code to interpret the message

sent by the Wand. JamFi does not require the target device to run additional code, or *even be aware that JamFi's technique is being used by the sender.* With JamFi, the target simply receives Wi-Fi frames as it normally does with any other sender.

We presented a preliminary version of JamFi at the S3 Workshop at MobiCom 2017, and showed that JamFi's jamming cover enables a nearby device to receive a data signal despite the presence of jamming that thwarts more distant adversaries [90]. In this thesis we present a more fully evaluated version of JamFi and tested its technique with several unmodified commercial-off-the-shelf Wi-Fi receivers. We found those receivers were able to recover BPSK 1/2 signals with high probability when devices were located roughly 5 cm apart. Additionally, because it can use the entire data-carrying capacity of a Wi-Fi frame, rather than just sending one bit per frame, we show that JamFi is up to 18,000 times faster than Wanda. This could create new opportunities to securely send large amounts of data.

JamFi makes the following contributions:

1. a consistent, fast, easy, and secure method to transfer any kind of information between commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the devices;

2. a theoretical analysis of jamming at close range to facilitate data transfer; and

3. an experimental evaluation using several different commercial off-the-shelf Wi-Fi receivers.

7.5 Conclusion

The promise of the Internet-of-Things is intriguing, but the challenges are many. We are skeptical that current approaches such as a smartphone application for each type of device will scale and we expect those approaches will quickly become impractical

when there are billions of IoT devices to configure, manage, and secure. We believe, however, that our three related solutions offer a secure path forward.

A

Wanda Appendix

A.1 RSSI Ratio in the near field

In Chapter 2 we model the received power on each of the Wand's two antennas using the RSSI Ratio derived from the log-normal shadow model. This model is appropriate when antennas are in the far field, but not when antennas are in the near field. Here we show that, even though the predicted received power is different in the near field from the far field, the RSSI Ratio is the same in both fields.

As noted in Chapter 5, Balanis gives an approximation of the average power for a thin-wire (radius $r \ll l$) finite-length dipole that is valid everywhere except on

the surface of the antenna [9]:

$$\mathbf{W}_{av} = \eta \frac{|I_0|^2}{8\pi^2 d^2} \left[\frac{\cos\left(\frac{kl}{2}\cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta} \right]^2 \tag{A.1}$$

where $\eta = 120\pi$ is the intrinsic impedance of free space, I_0 is the current applied to the transmitter, $k = 2\pi/\lambda$ is the wavenumber, l is the length of the transmitting antenna, d is the distance from the transmitting antenna, and θ is the vertical angle between the transmitter and receiver (below we assume $\theta = \pi/4$).

For notational simplicity let ρ represent Equation (A.1) without the distance d between antennas

$$\rho = \eta \frac{|I_0|^2}{8\pi^2} \left[\frac{\cos\left(\frac{kl}{2}\cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta} \right]^2 . \tag{A.2}$$

Together with distance d, Equation (A.1) can be reconstructed as:

$$\mathbf{W}_{av} = \frac{\rho}{d^2}. \tag{A.3}$$

We can convert Equation (A.3) into the decibel scale as follows:

$$\mathbf{W}_{decibel} = 10 \log_{10}(\frac{\rho}{d^2}). \tag{A.4}$$

Assuming identical antennas on the Wand, the power, P, captured by each antenna in dBm will be proportional to Equation (A.4), after accounting for antenna gain and any loss in the receiver. We now calculate the RSSI Ratio between antenna

A_1 at distance d_1 and antenna A_2 at distance d_2 as:

$$\begin{aligned} P_1 - P_2 &= 10\log_{10}(\frac{\rho}{d_1^2}) - 10\log_{10}(\frac{\rho}{d_2^2}) \\ &= 10\log_{10}(\rho) - 10\log_{10}(d_1^2) - 10\log_{10}(\rho) + 10\log_{10}(d_2^2) \\ &= -20\log_{10}(d_1) + 20\log_{10}(d_2) \\ &= -20\log_{10}(\frac{d_1}{d_2}). \end{aligned} \tag{A.5}$$

In free space $\alpha = 2$. We see that Equation (A.5) is the same as Equation (2.6) with $\alpha = 2$, suggesting the RSSI Ratio is the same in the near field and the far field.

A.2 Wanda user study participant comments

As part of the 36-person Wanda user study, we asked participants for general comments on using Wanda versus methods they had previously used to configure devices. Their answers are shown in the following table.

#	*Comment*
1.	Very easy to use.
2.	Very easy to use. I got it right away.
3.	Straight forward.
4.	Seems clear, easy to use, very satisfactory experience.
5.	Very easy, with the written instructions.
6.	Easy as pie.
7.	Very simple, explains itself, which is a good thing, I'm not very good when it comes to computers.
8.	I'm excited because it is easy, it feels magic. I love the concept because my goal is not to connect the device, my goal is to use the device.
9.	More intuitive and simpler than the normal way. I like that I don't have to type in my password.
10.	Why can't everything be that easy? Can I take it home?
11.	This is pretty cool. It's super easy compared with how I normally use my password, which is not easy to remember. I am surprised more devices aren't this simple to use.
12.	I would not trust it. I would worry that my information would be compromised. This would be less safe than entering the password myself. I had identity theft once on my computer, so I'm very careful now.
13.	Seems a lot easier to me. I can't see any downside or hang ups to trip people up that aren't computer savvy.

14. Makes it thoughtless to use. Very nice.
15. This is much easier and much simpler. I feel like anybody could do this.
16. 100% of people could do this, even the somewhat mentally impaired could probably do it.
17. If life was that easy I'd be extremely happy!
18. Super easy and intuitive. I would trust my grandmother with this.
19. I think it is quicker and simpler than traditional methods.
20. Really simple. I don't have a very steady hand, but it managed to connect. That's pretty impressive. It was fast and I don't have to think of an app.
21. This was much easier. It was seamless and fun.
22. Easier, less complicated.
23. I would prefer to connect through this method. This is much less complicated and the same every time. My grandfather could do this, he could not otherwise.
24. It was great. It seems like it would need a universal docking port. Completely usable as it is though.
25. This is so straightforward; it's perfect.
26. This goes beyond medical devices, this would be a great way to hook up all kinds of things. A really good idea.
27. Let me know when this is on the market so I can get one.
28. Very easy to use. I thought it was pretty fun.
29. It's easy, it's good.
30. Simple and straightforward. Foolproof.
31. Pretty intuitive. The wand is an intuitive proxy between the router and some device you'd like to connect with it.
32. Super easy to use. I could do it, my Mom could as well.
33. Wanda is a much simpler and much more likely to be used.

34.	It would be very easy for older people.
35.	I love it! Its easy and fun to use. When can I get one?
36.	I'm looking forward to this being on the market. It's so easy and quick.

Table A.1: Study participant comments.

B

SNAP Appendix

B.1 Distribution of preamble errors by antenna type

The distribution of E_j for 1,000 Wi-Fi frames sent at distances from 2 cm to 3 m is shown by antenna type in Figure B.1, Figure B.2, Figure B.3, and Figure B.4 for a half-wavelength, quarter-wavelength, micropatch, and Panda USB adapter.

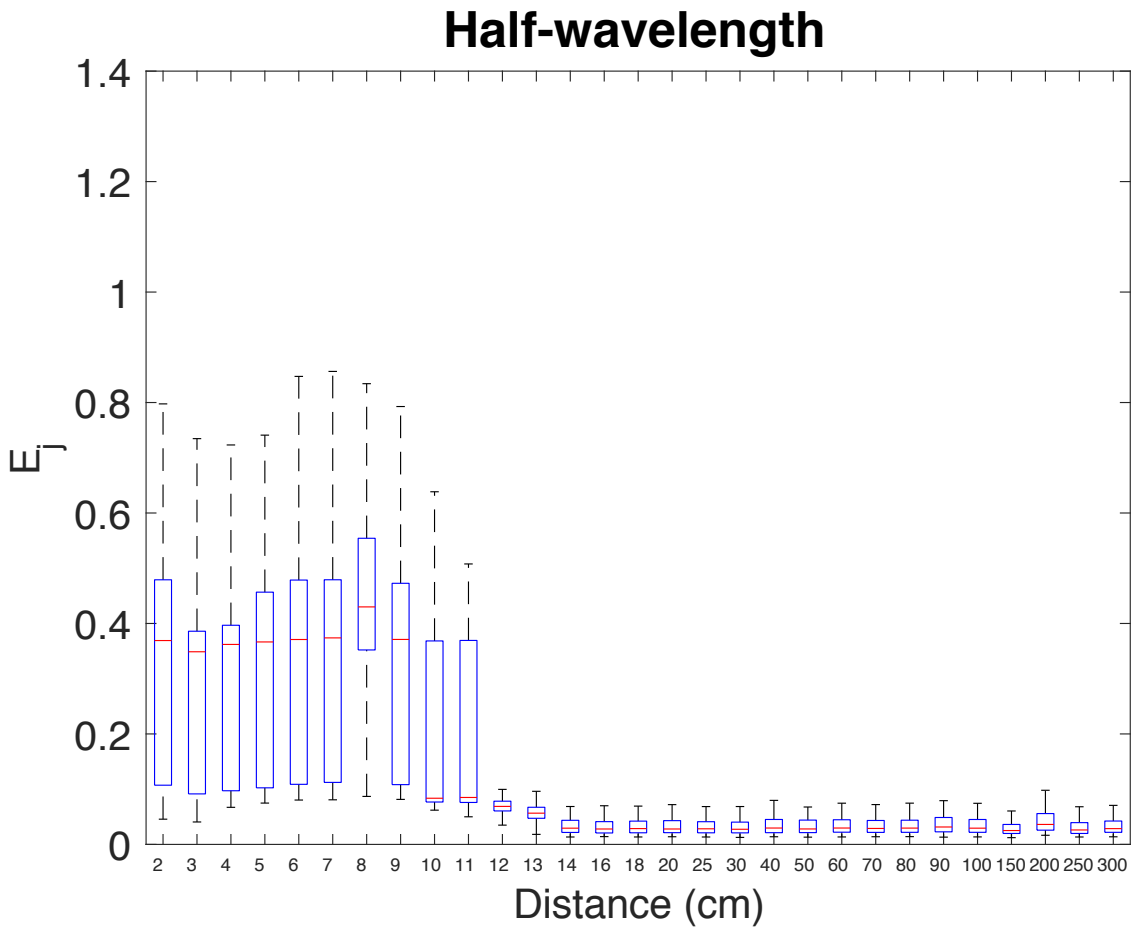

Figure B.1: Distribution of sum of phase differences for a half-wavelength antenna. Sum of differences of phase and amplitude between T_1 and T_2 in the Long Training Field portion of the Wi-Fi preamble over 64 subcarriers as calculated by Equation (4.15). Results are from 1,000 Wi-Fi frames transmitted at each distance. The red line indicates the median value, the box indicates the 75th and 25th percentile, and the whiskers indicate the maximum and minimum value.

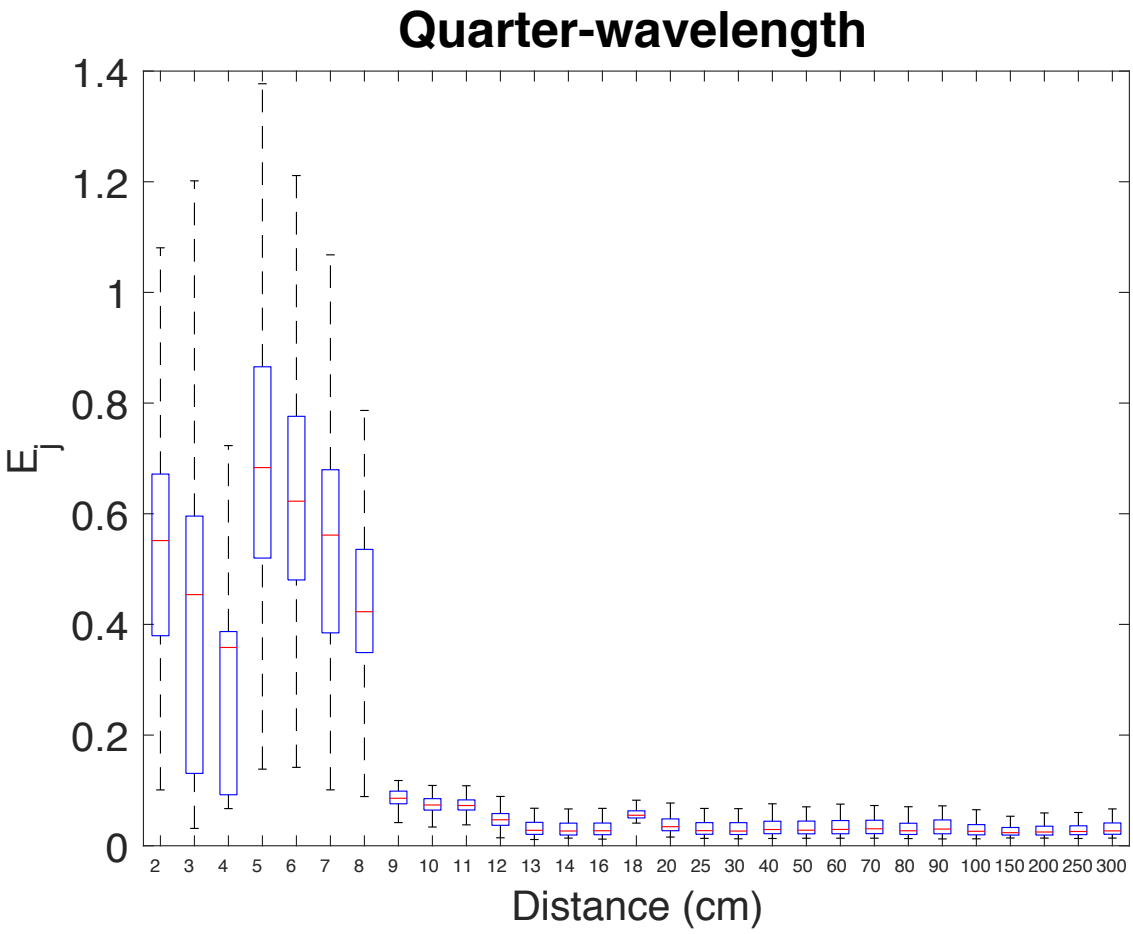

Figure B.2: Distribution of sum of phase differences for a quarter-wavelength antenna. Sum of differences of phase and amplitude between T_1 and T_2 in the Long Training Field portion of the Wi-Fi preamble over 64 subcarriers as calculated by Equation (4.15). Results are from 1,000 Wi-Fi frames transmitted at each distance. The red line indicates the median value, the box indicates the 75th and 25th percentile, and the whiskers indicate the maximum and minimum value.

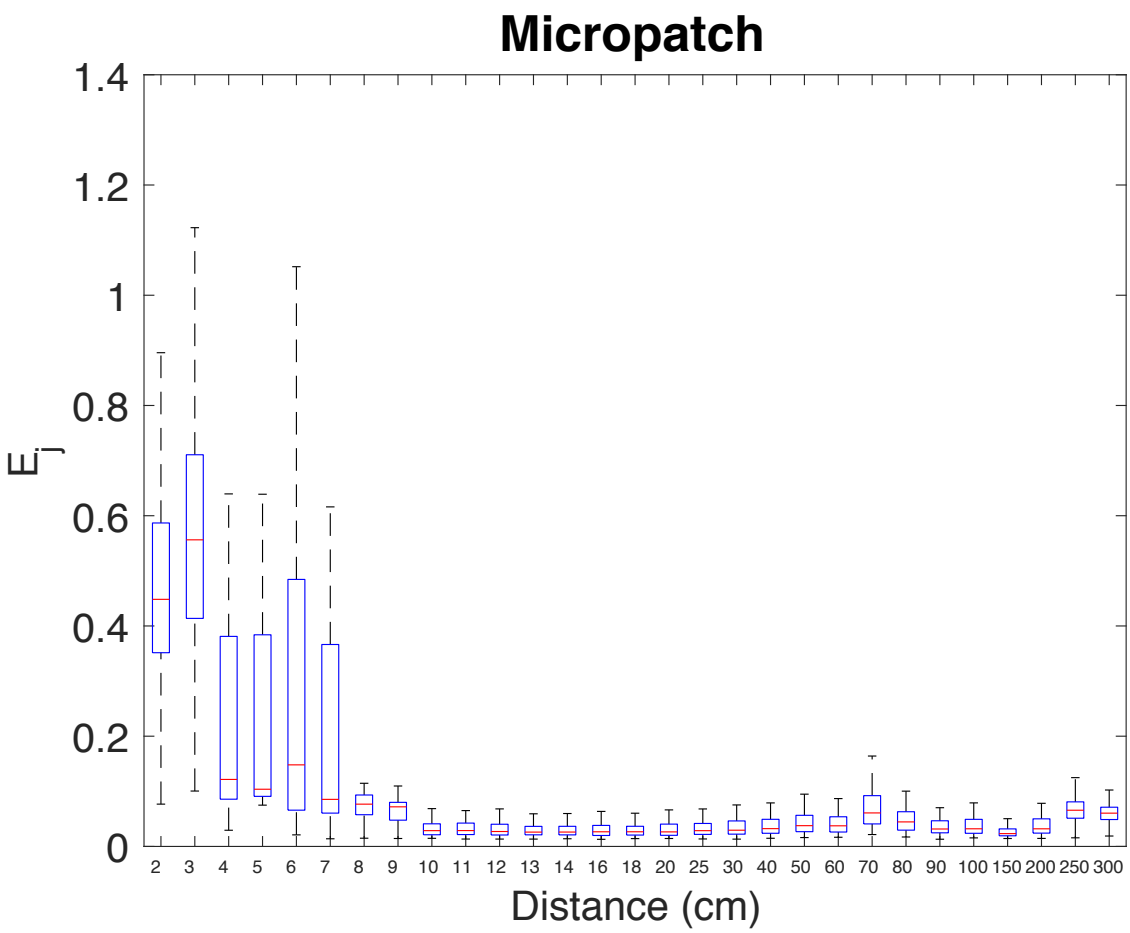

Figure B.3: Distribution of sum of phase differences for a micropatch antenna. Sum of differences of phase and amplitude between T_1 and T_2 in the Long Training Field portion of the Wi-Fi preamble over 64 subcarriers as calculated by Equation (4.15). Results are from 1,000 Wi-Fi frames transmitted at each distance. The red line indicates the median value, the box indicates the 75th and 25th percentile, and the whiskers indicate the maximum and minimum value.

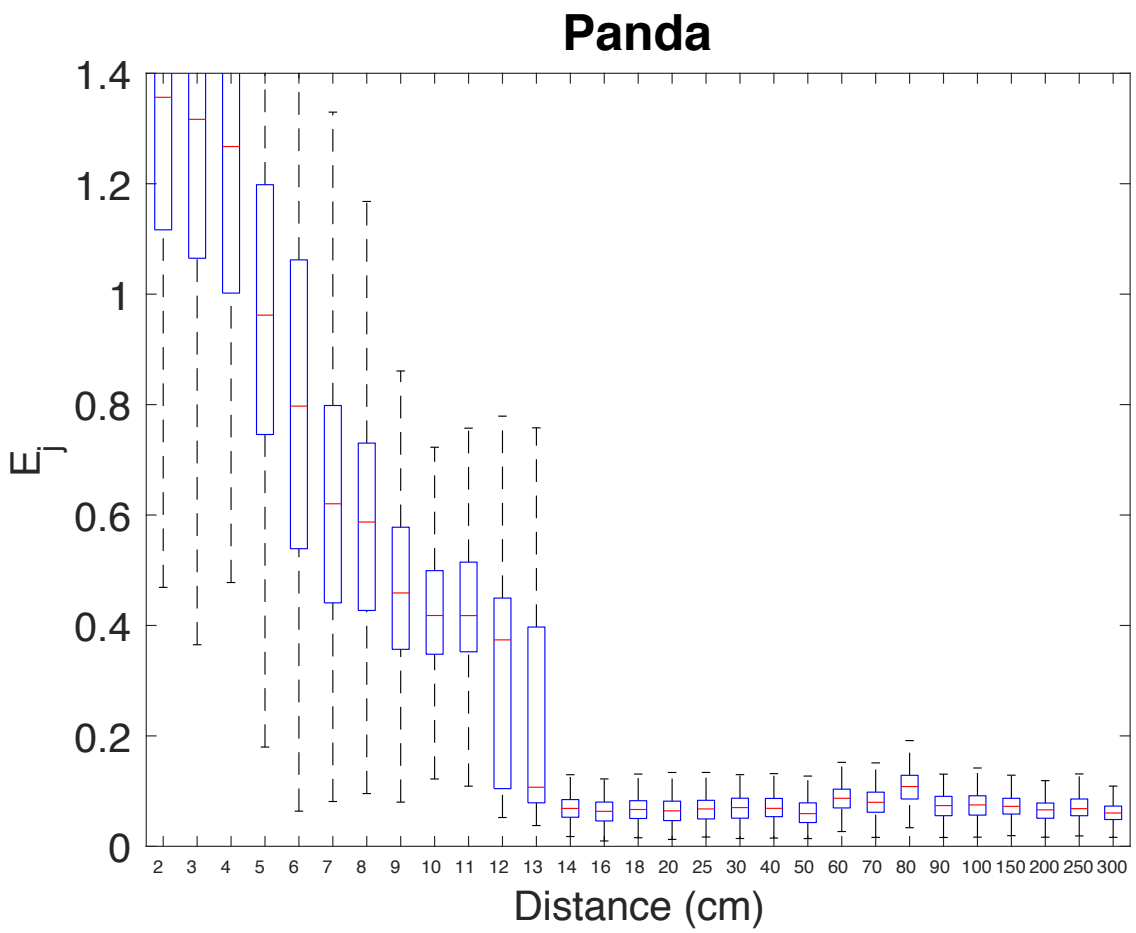

Figure B.4: Distribution of sum of phase differences for a Panda Ultra Wireless N USB adapter. Sum of differences of phase and amplitude between T_1 and T_2 in the Long Training Field portion of the Wi-Fi preamble over 64 subcarriers as calculated by Equation (4.15). Results are from 1,000 Wi-Fi frames transmitted at each distance. The red line indicates the median value, the box indicates the 75th and 25th percentile, and the whiskers indicate the maximum and minimum value.

B.2 Likelihood of detecting proximity by antenna, distance, and threshold

The likelihood of declaring proximity based on a Monte Carlo simulation of measure preamble differences and tranmiting antenna type and thresholds are shown in Figures B.5 – B.16 for a half-wavelength, quarter-wavelength, micropatch, and Panda Ultra N Wireless USB adapter.

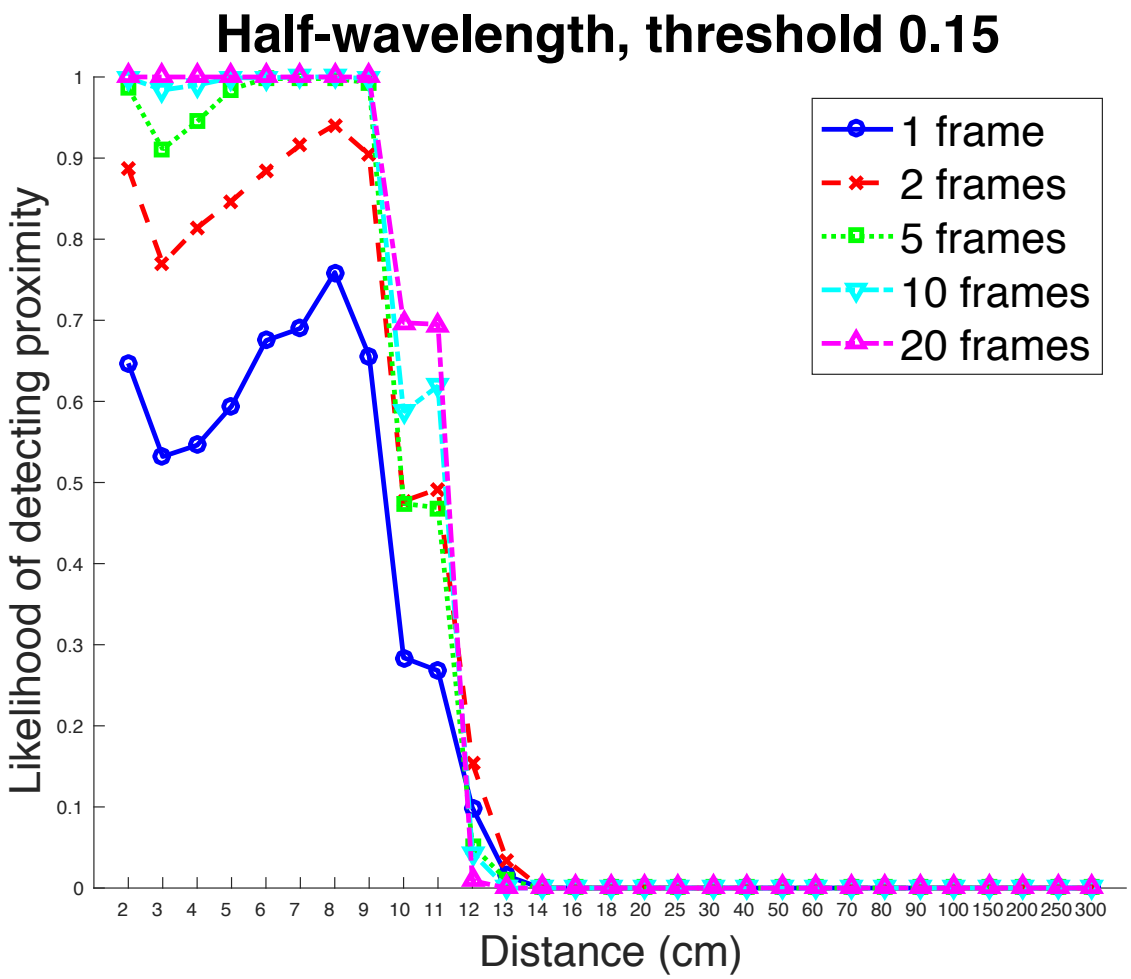

Figure B.5: Likelihood of declaring proximity at various distances with a half-wavelength antenna and $\tau = 0.15$.

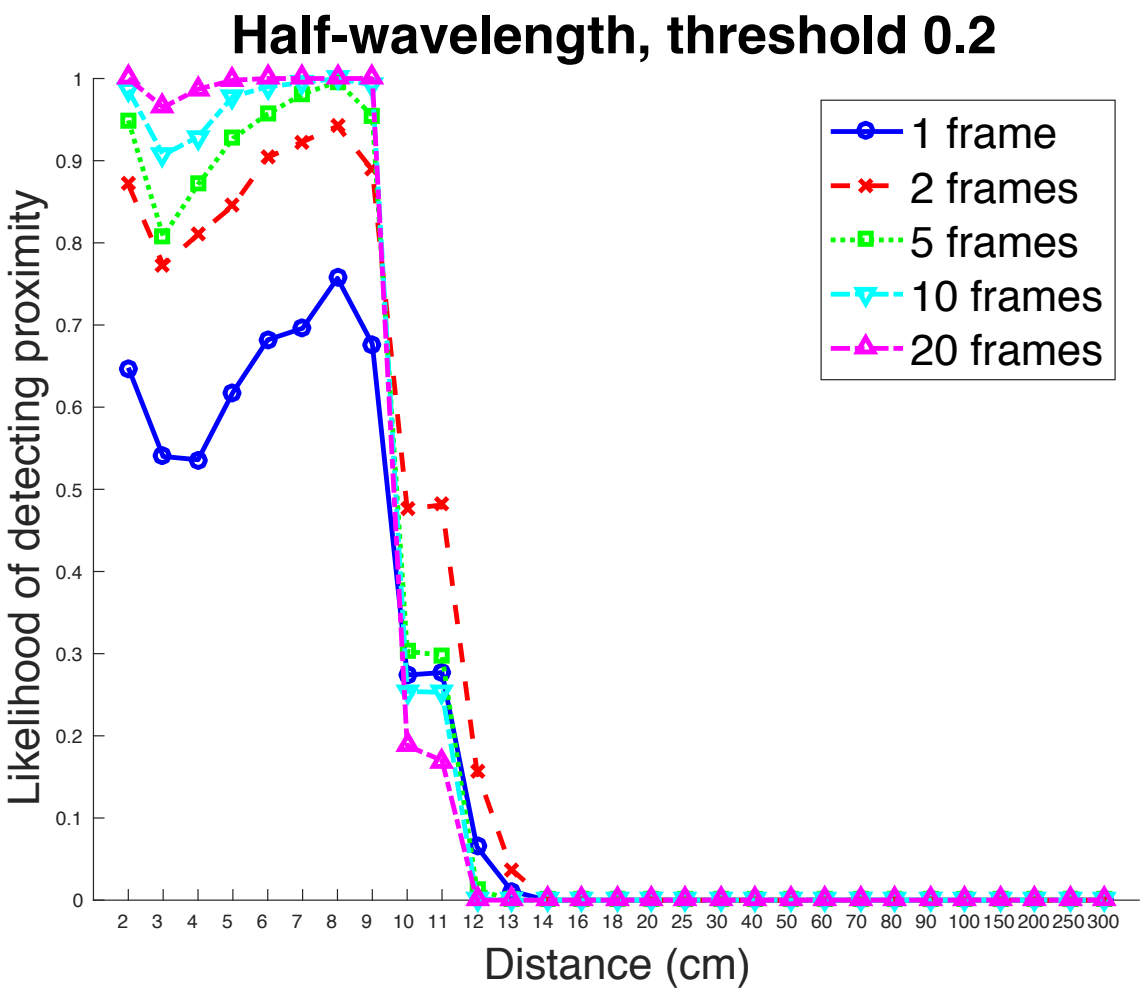

Figure B.6: Likelihood of declaring proximity at various distances with a half-wavelength antenna and $\tau = 0.2$.

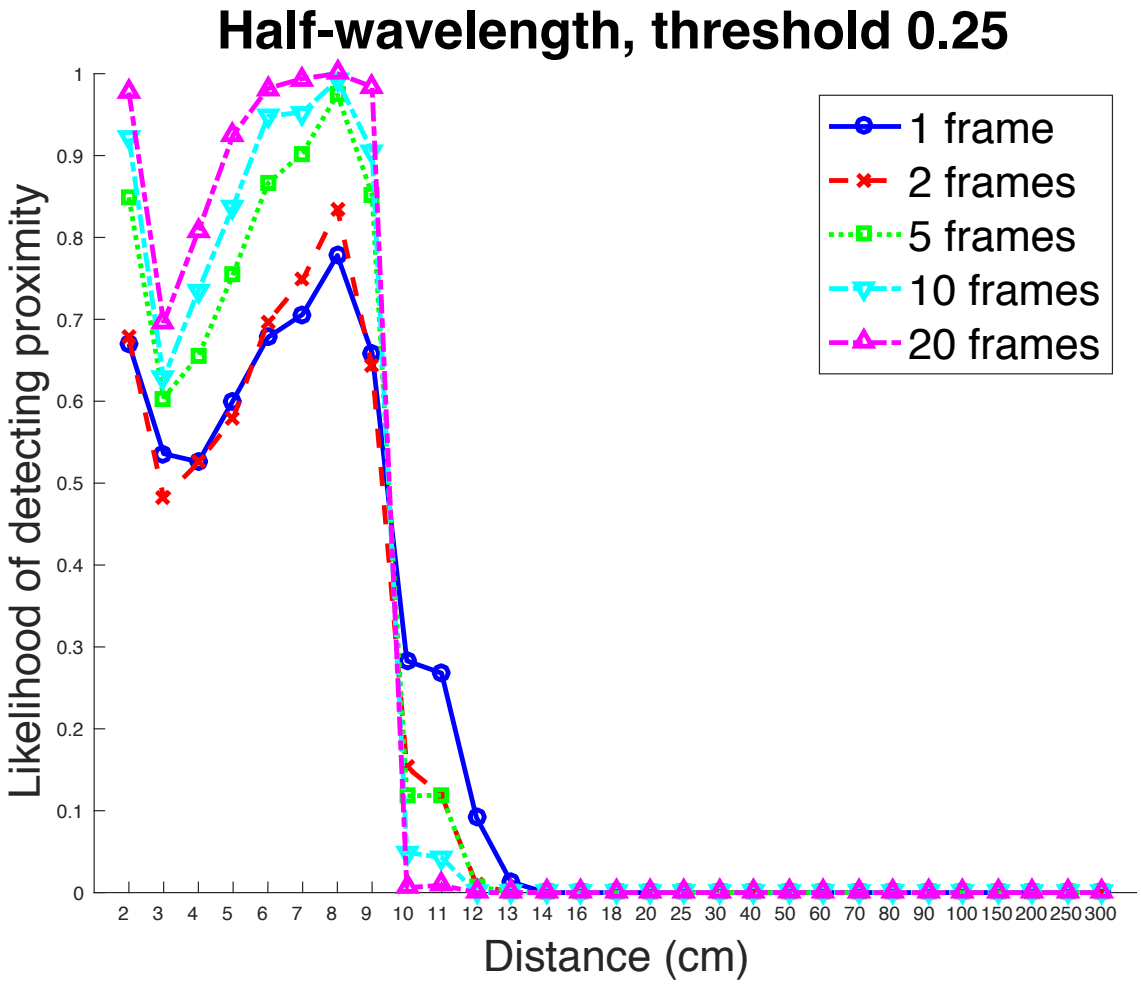

Figure B.7: Likelihood of declaring proximity at various distances with a half-wavelength antenna and $\tau = 0.25$.

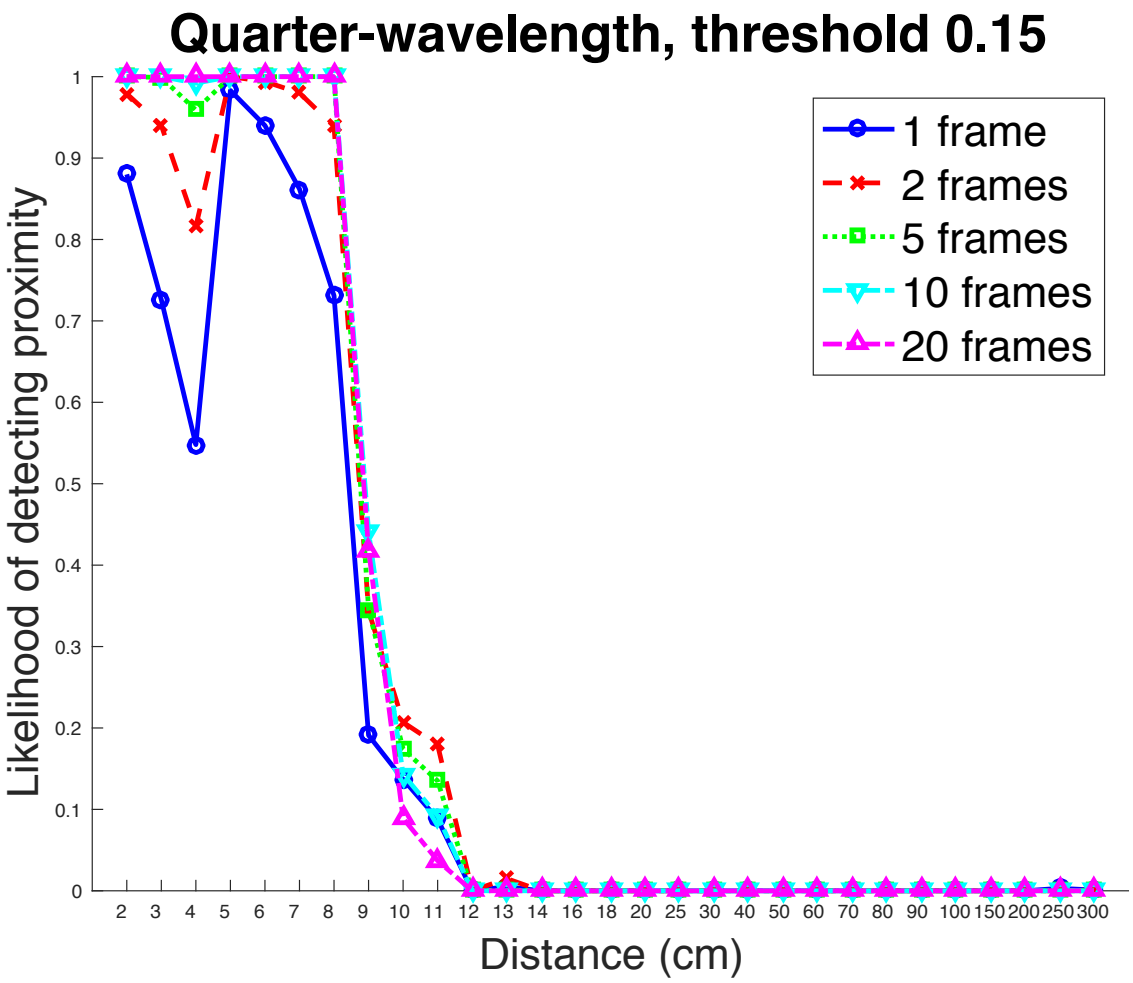

Figure B.8: Likelihood of declaring proximity at various distances with a quarter-wavelength antenna and $\tau = 0.15$.

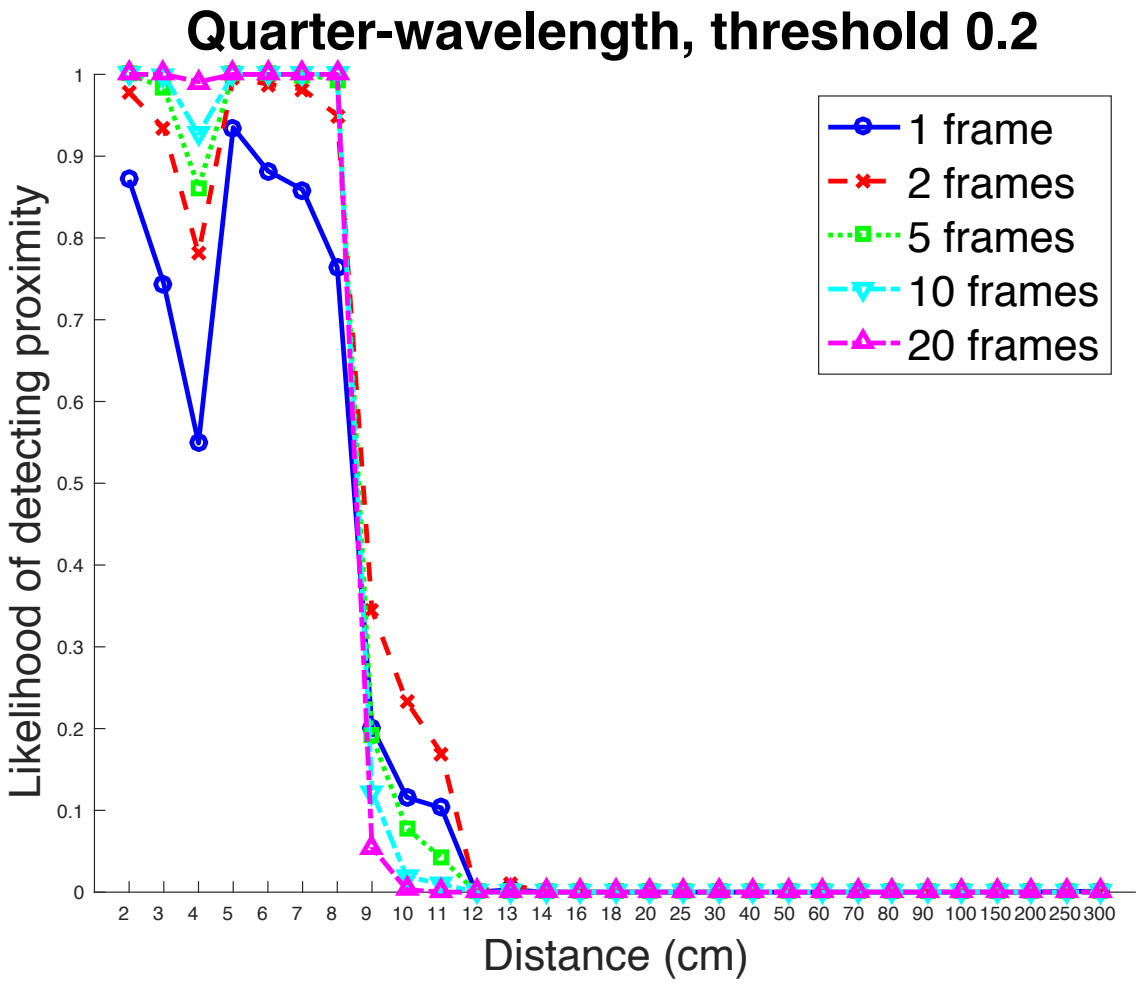

Figure B.9: Likelihood of declaring proximity at various distances with a quarter-wavelength antenna and $\tau = 0.2$.

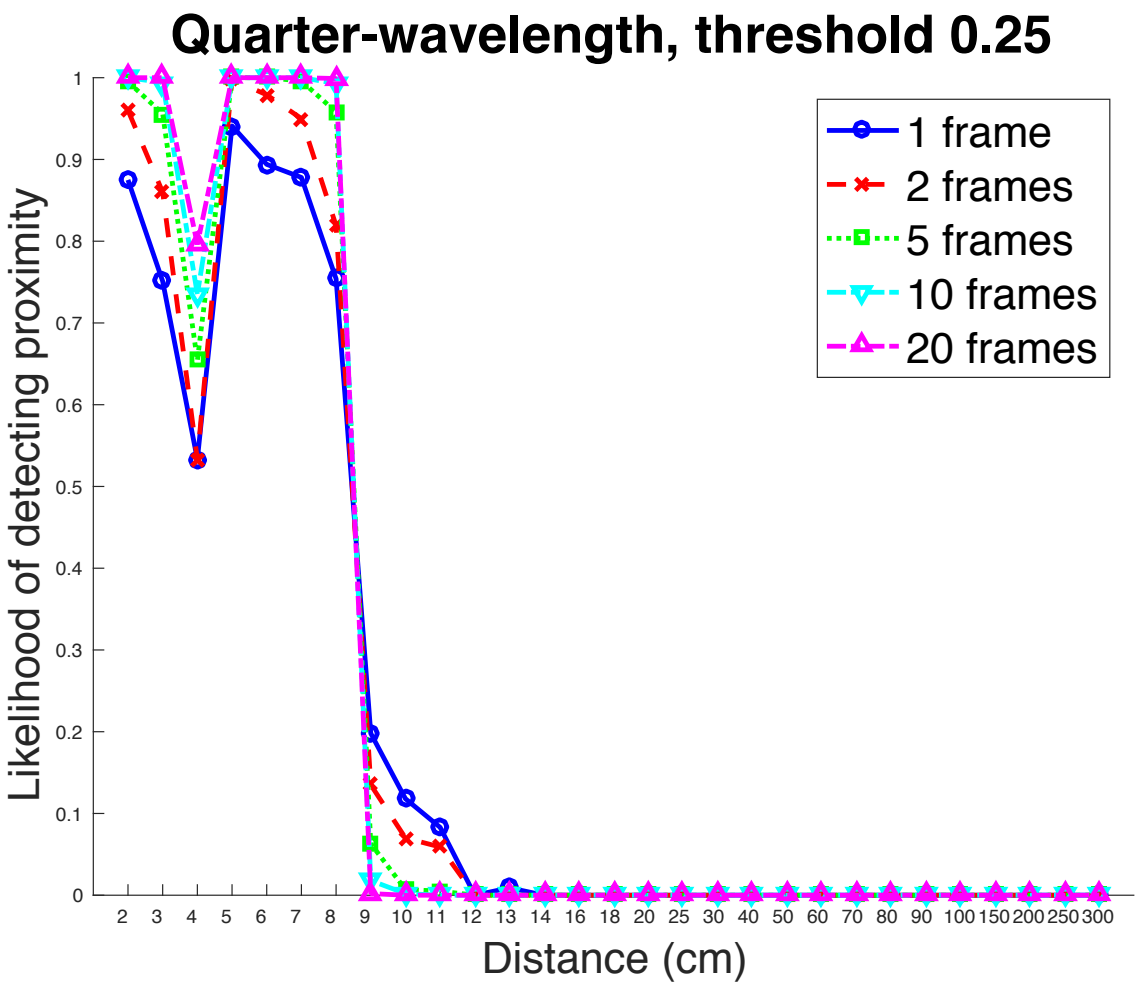

Figure B.10: Likelihood of declaring proximity at various distances with a quarter-wavelength antenna and $\tau = 0.25$.

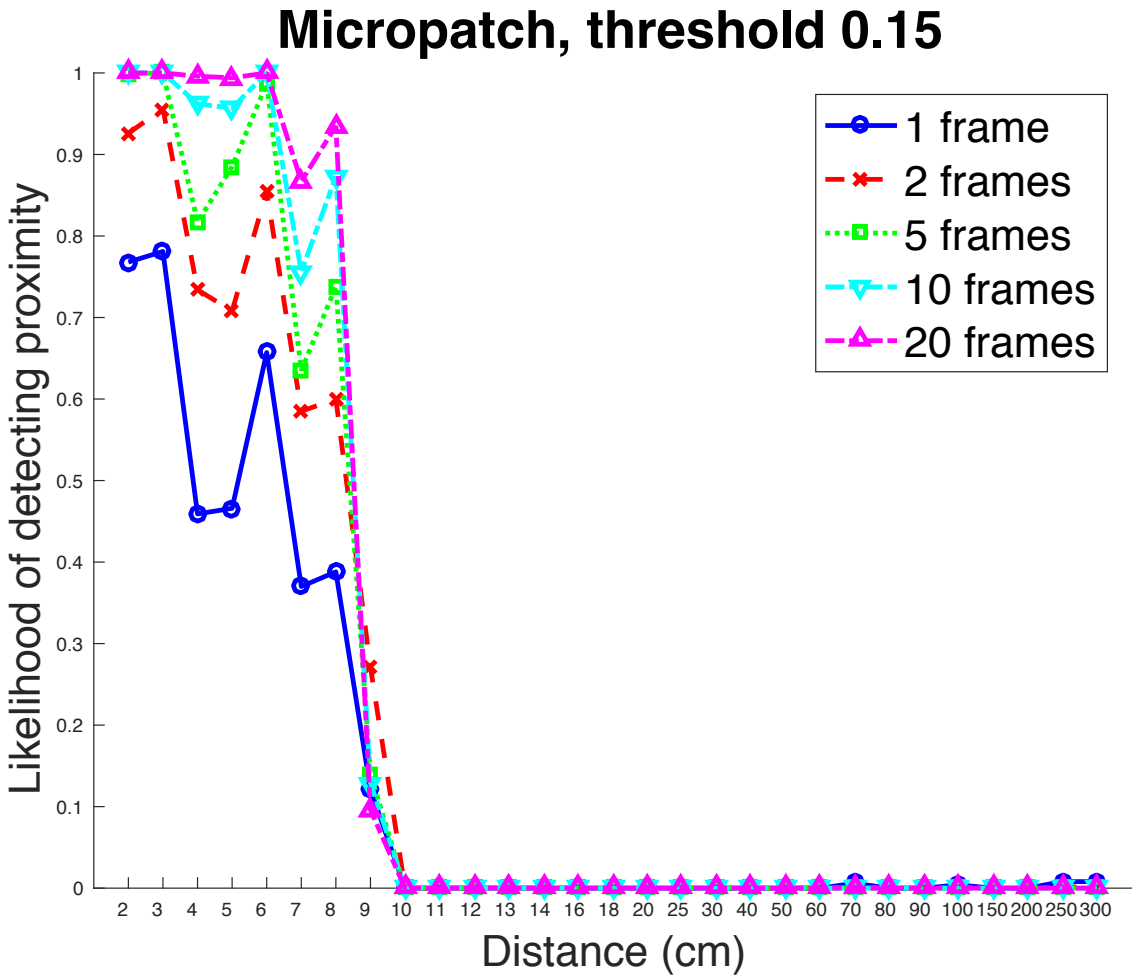

Figure B.11: Likelihood of declaring proximity at various distances with a micropatch antenna and $\tau = 0.15$.

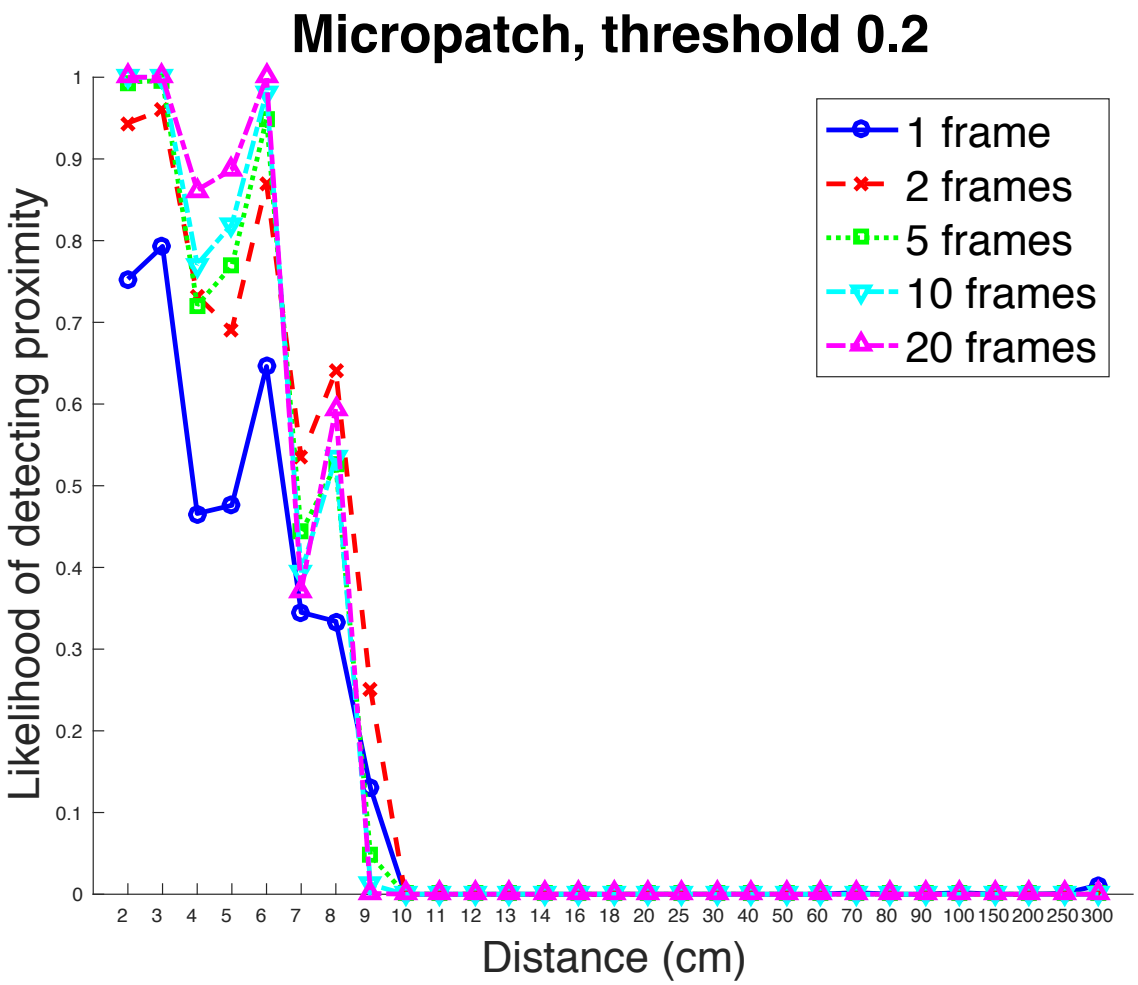

Figure B.12: Likelihood of declaring proximity at various distances with a micropatch antenna and $\tau = 0.2$.

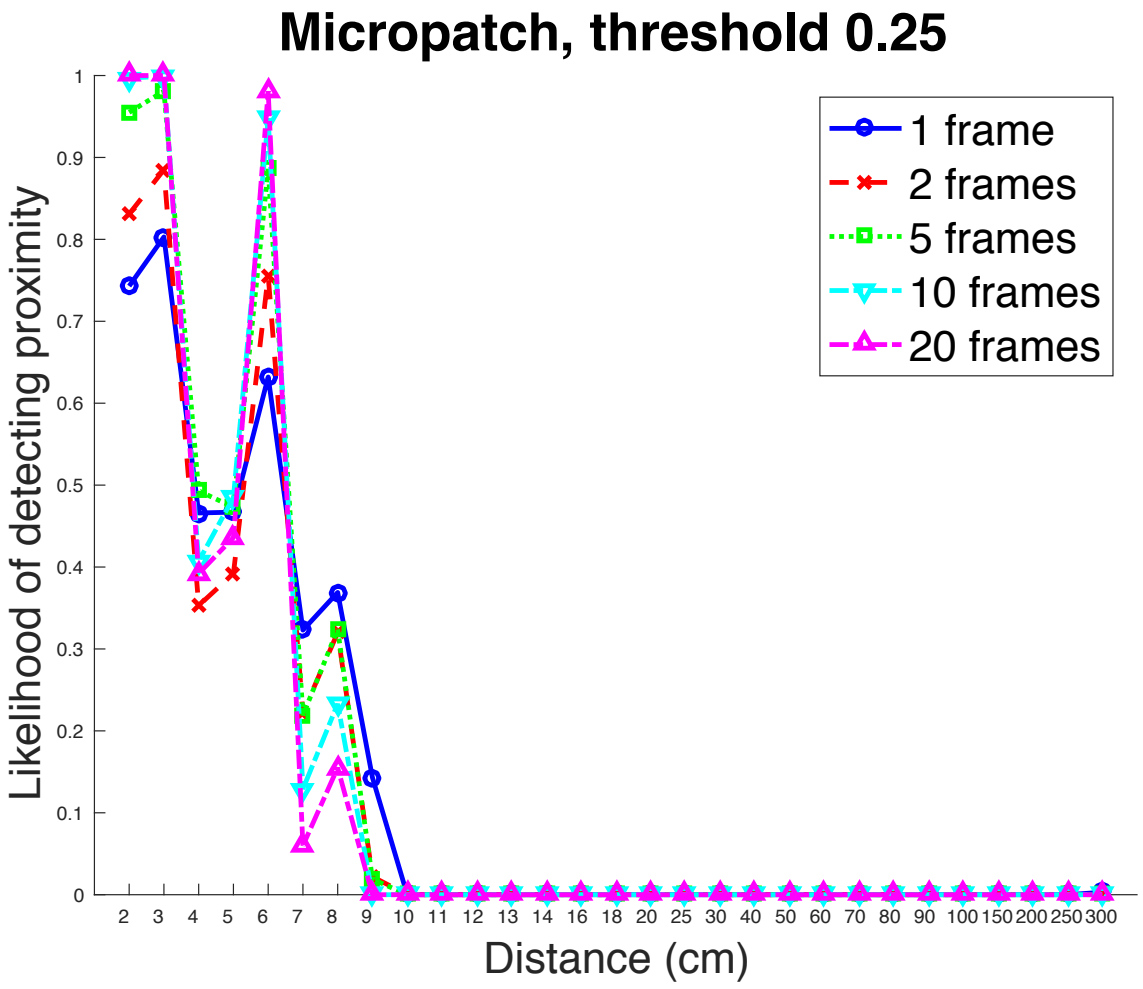

Figure B.13: Likelihood of declaring proximity at various distances with a micropatch antenna and $\tau = 0.25$.

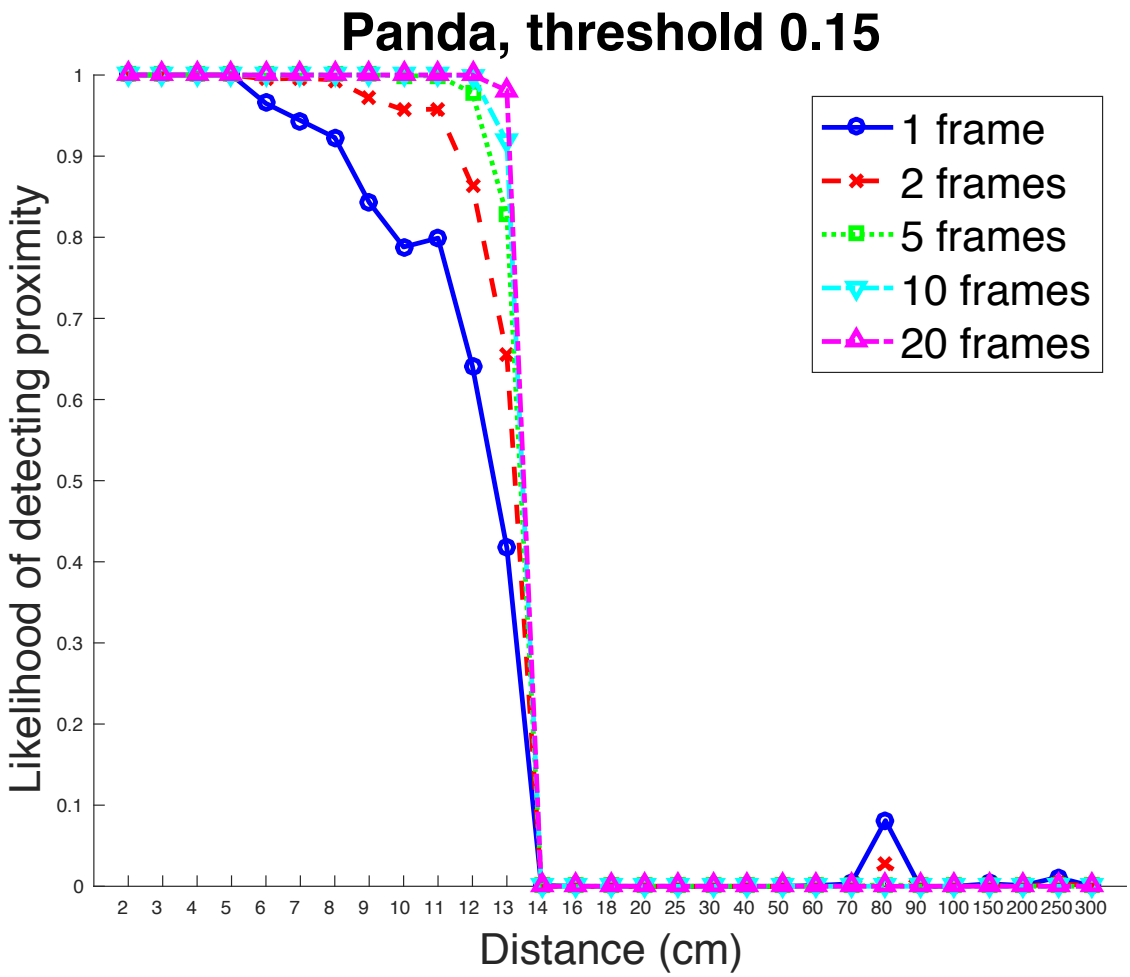

Figure B.14: Likelihood of declaring proximity at various distances with a Panda Ultra N Wireless USB antenna and $\tau = 0.15$.

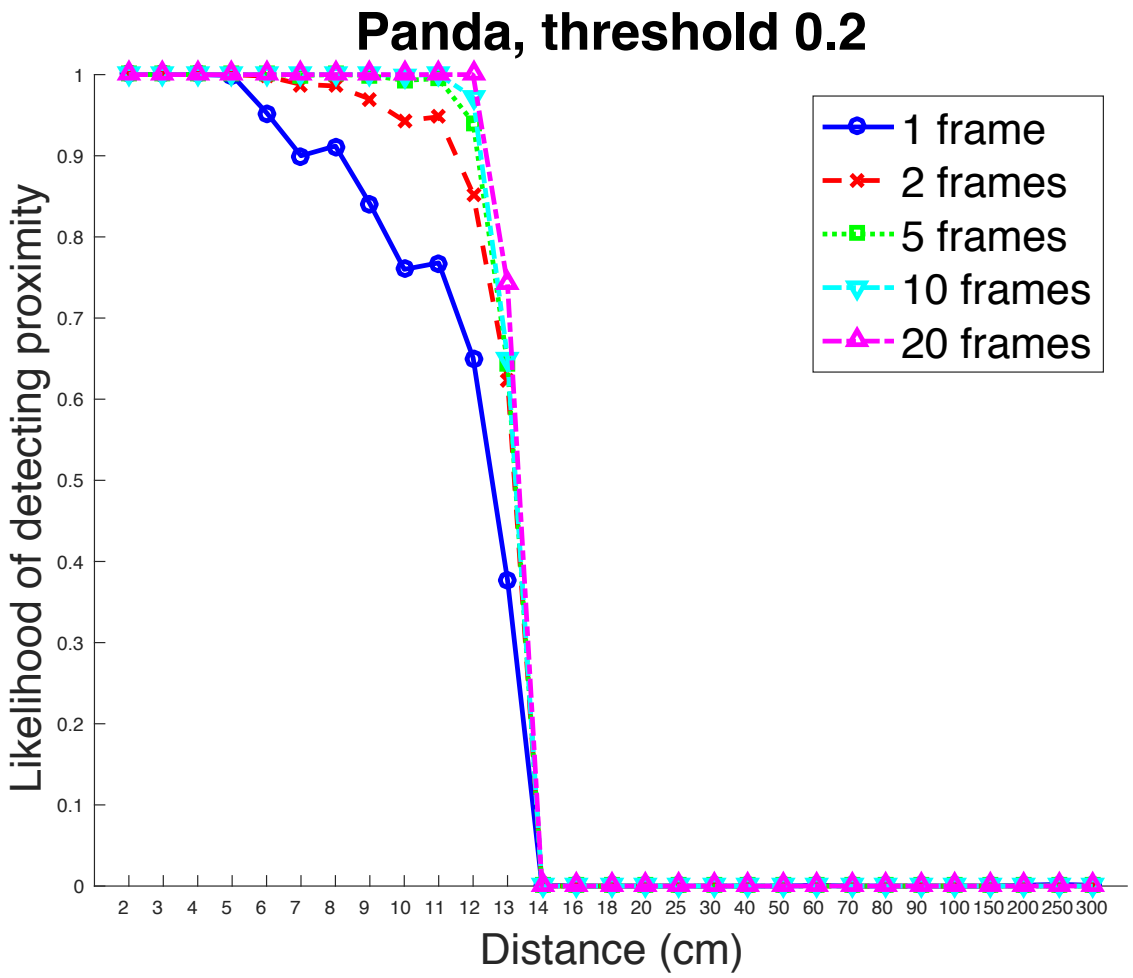

Figure B.15: Likelihood of declaring proximity at various distances with a Panda Ultra N Wireless USB antenna and $\tau = 0.2$.

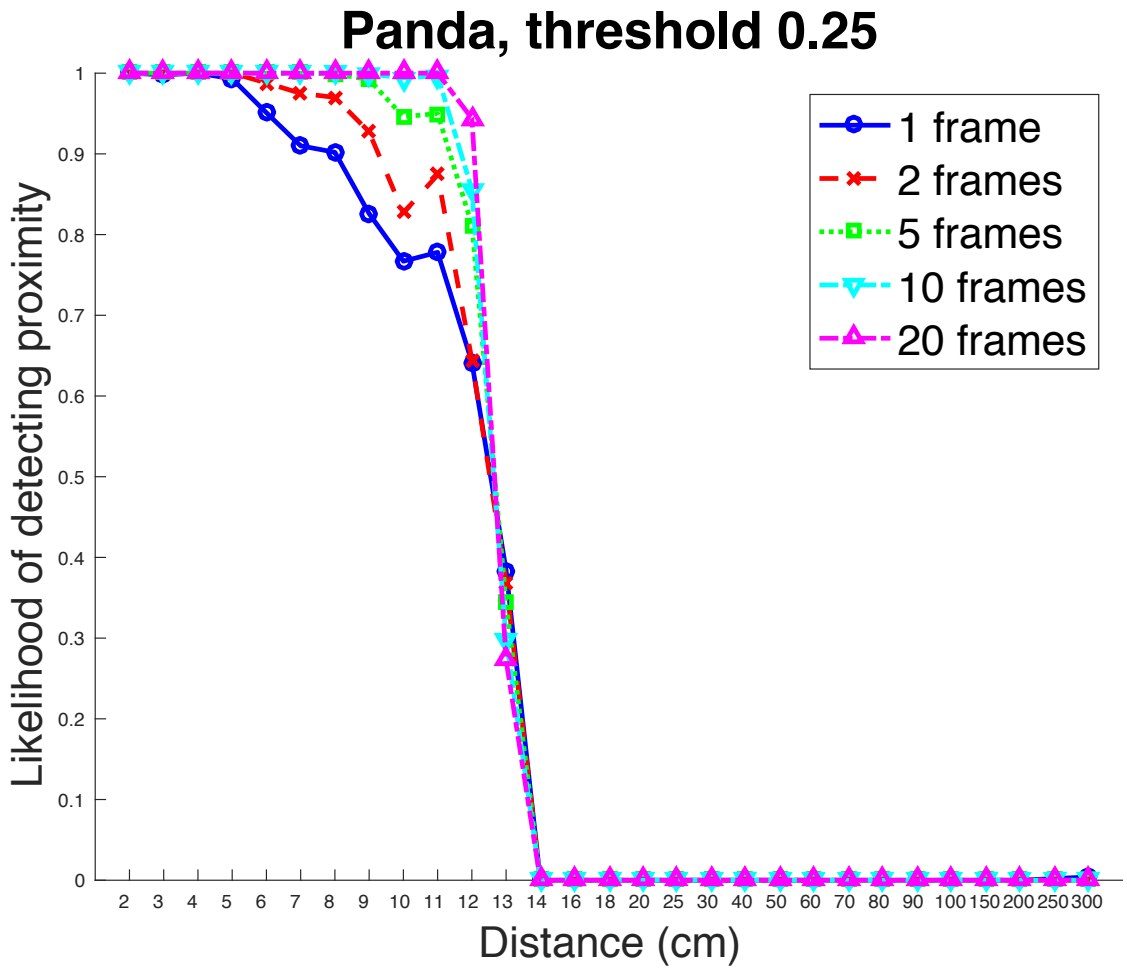

Figure B.16: Likelihood of declaring proximity at various distances with a Panda Ultra N Wireless USB antenna and $\tau = 0.25$.

C

JamFi Appendix

This appendix provides the Frame Reception Ratio for each of the Panda, Alfa, Intel, and Edimax adapters tested in Chapter 5 with jamming, with $P_j = 4$ dBm and with $P_j = 8$ dBm.

C.1 Panda

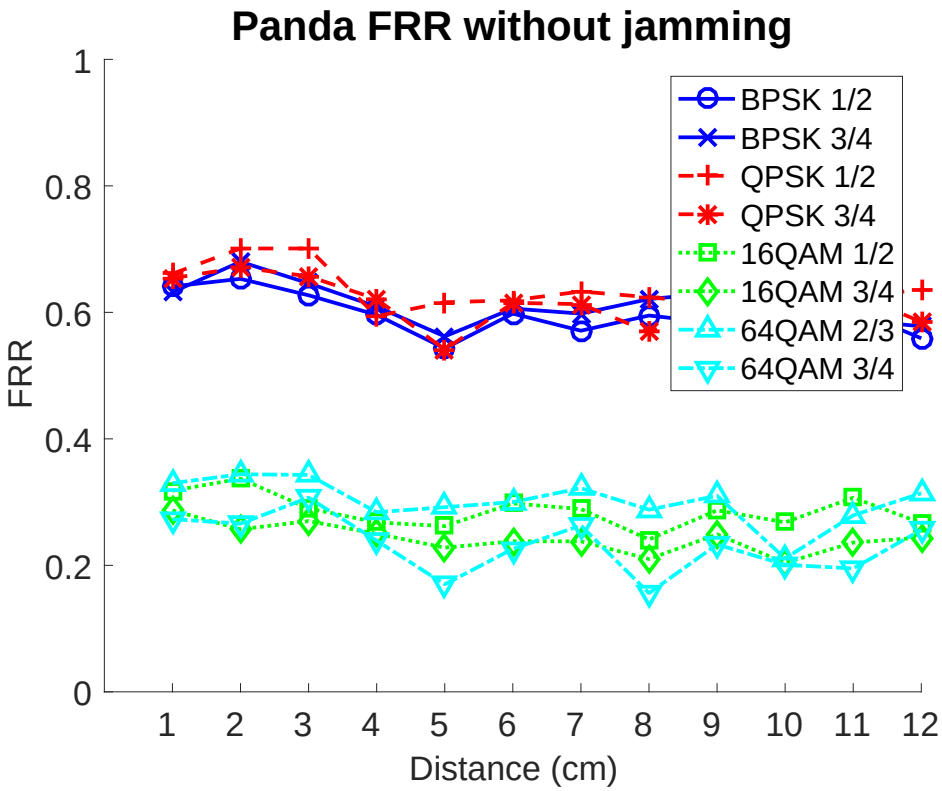

Figure C.1: Panda FRR with no jamming.

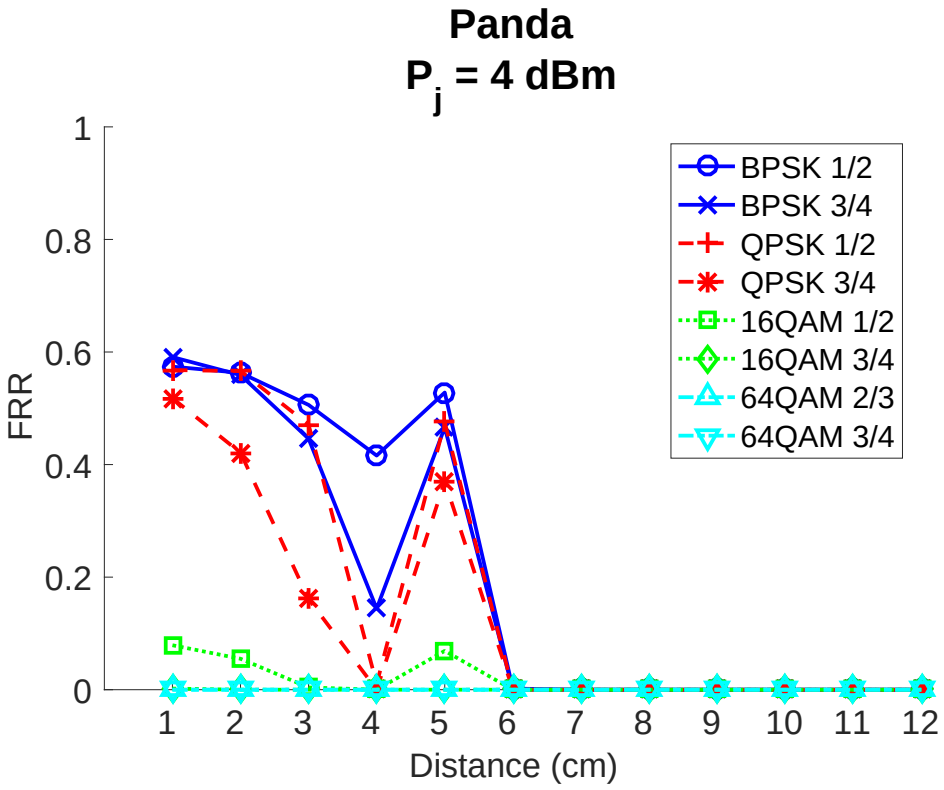

Figure C.2: Panda FRR with $P_j = 4$ dBm.

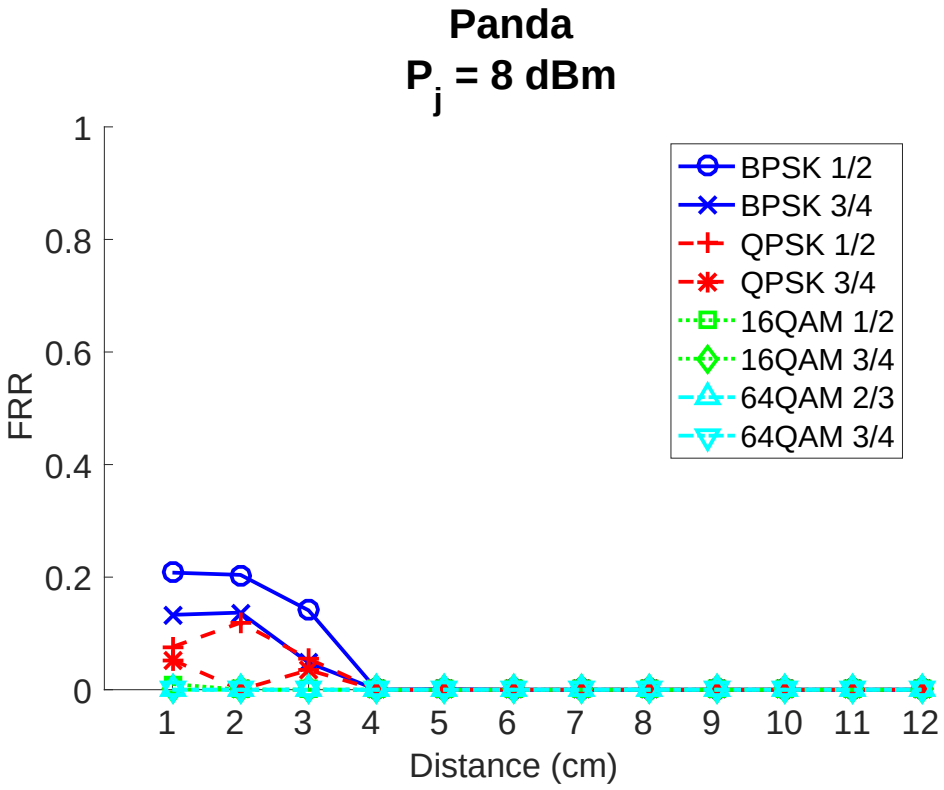

Figure C.3: Panda FRR with $P_j = 8$ dBm.

C.2 Alfa

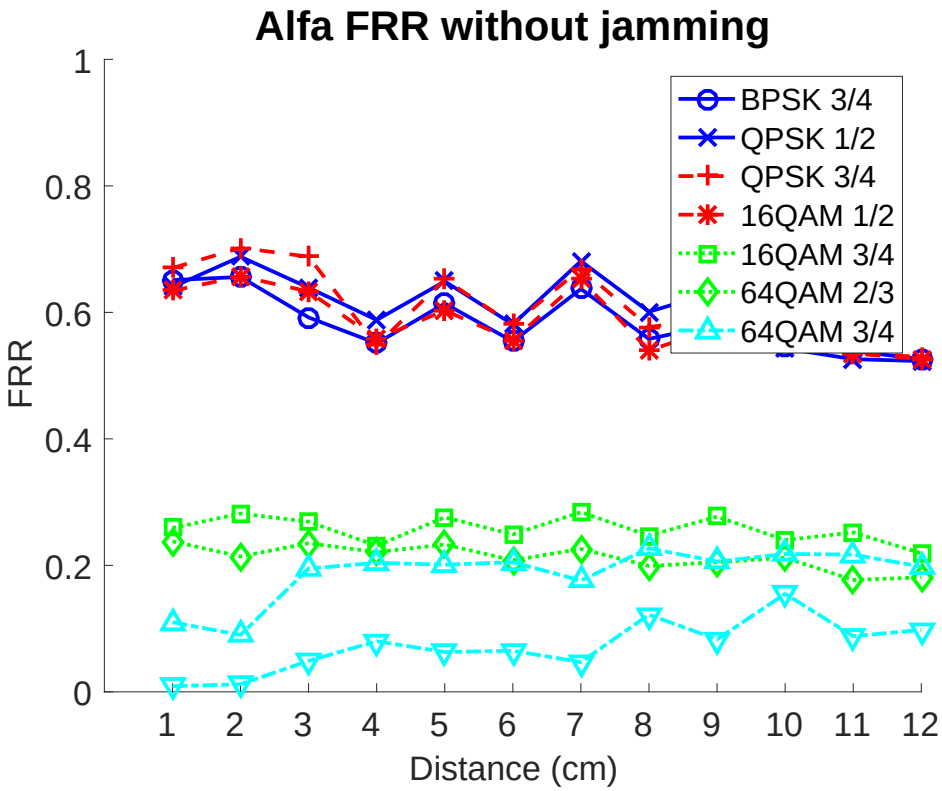

Figure C.4: Alfa FRR with no jamming.

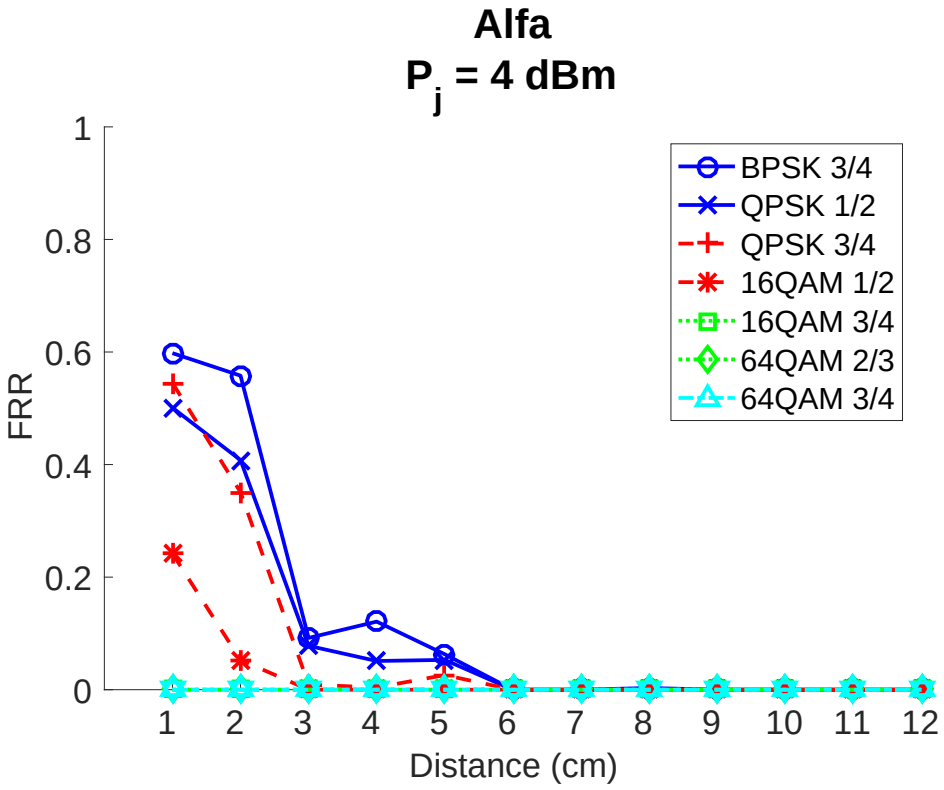

Figure C.5: Alfa FRR with $P_j = 4$ dBm.

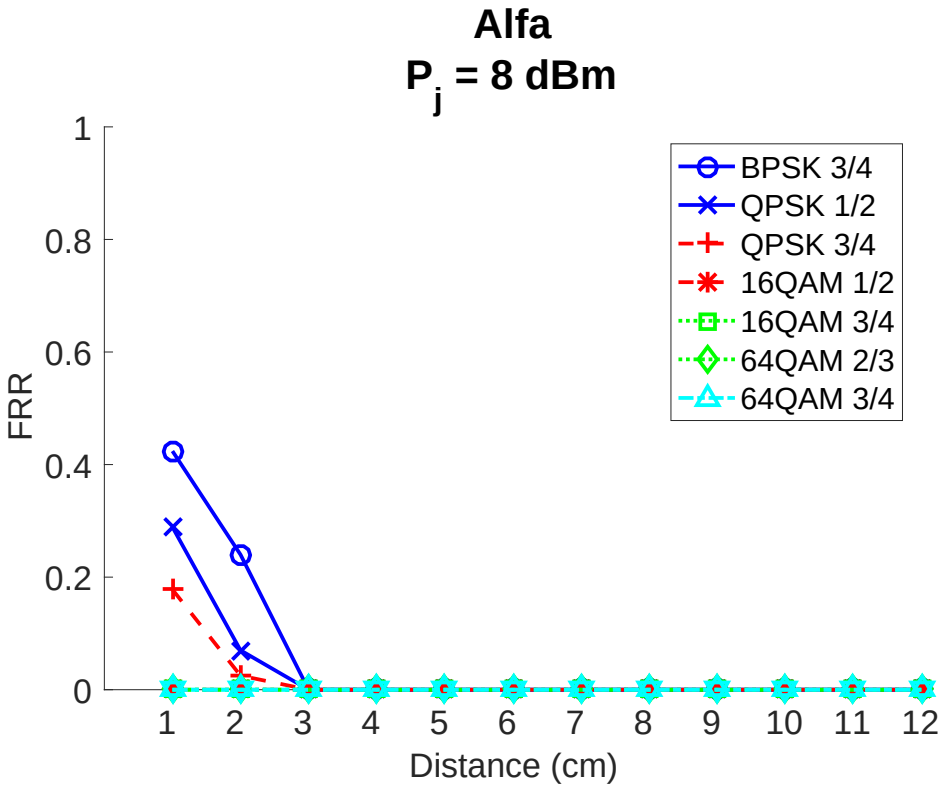

Figure C.6: Alfa FRR with $P_j = 8$ dBm.

C.3 Intel

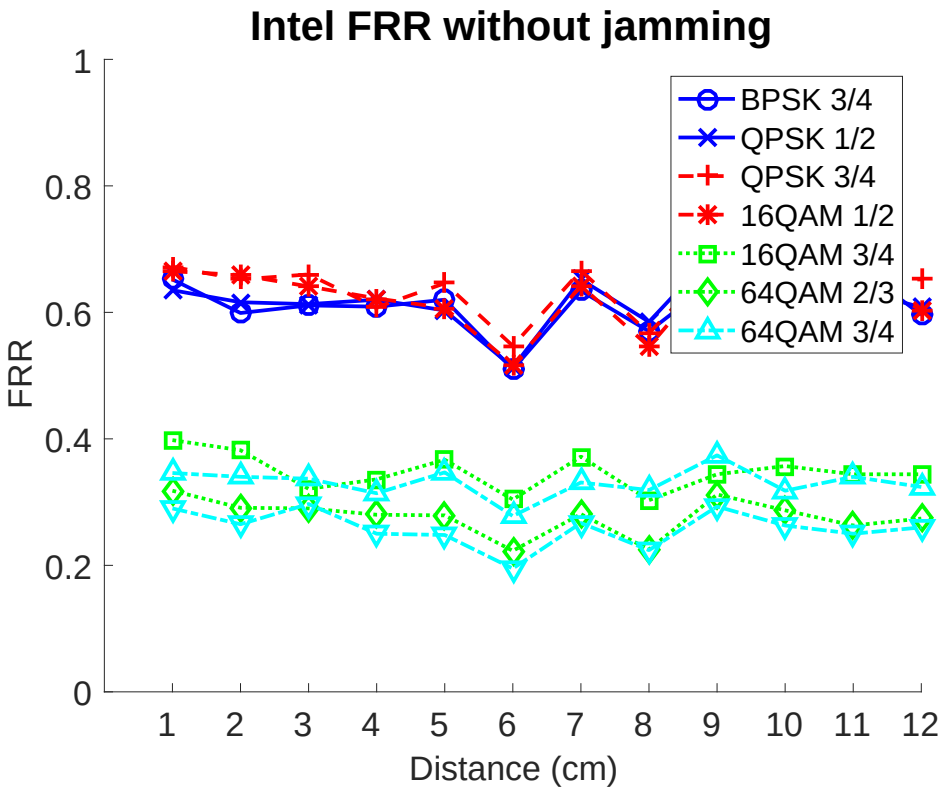

Figure C.7: Intel FRR with no jamming.

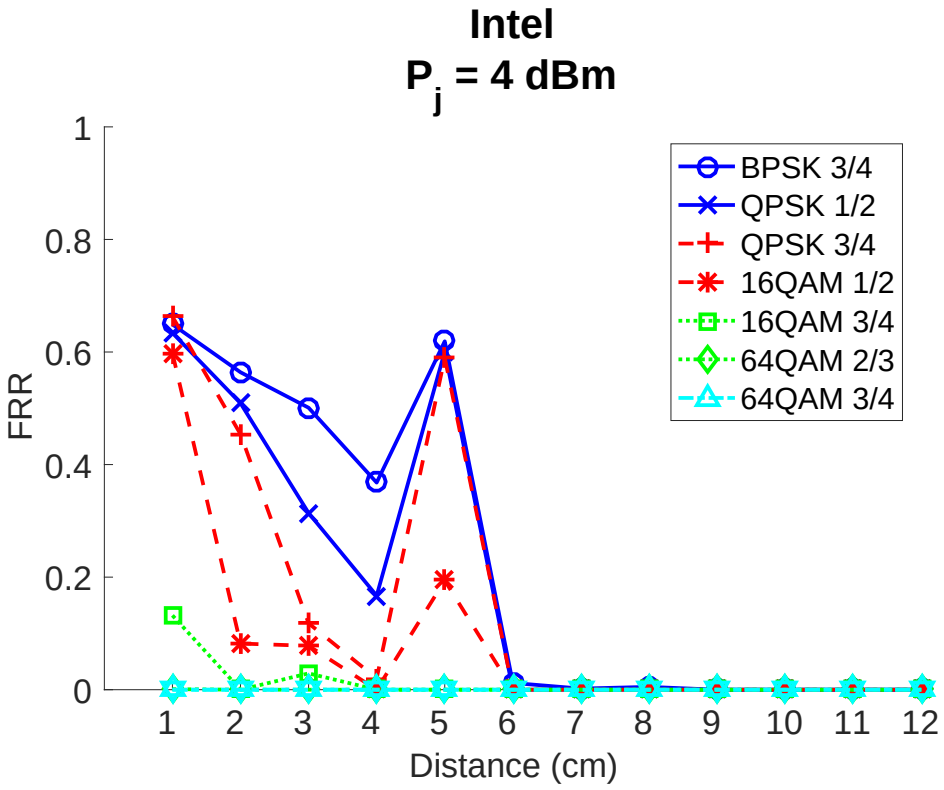

Figure C.8: Intel FRR with $P_j = 4$ dBm.

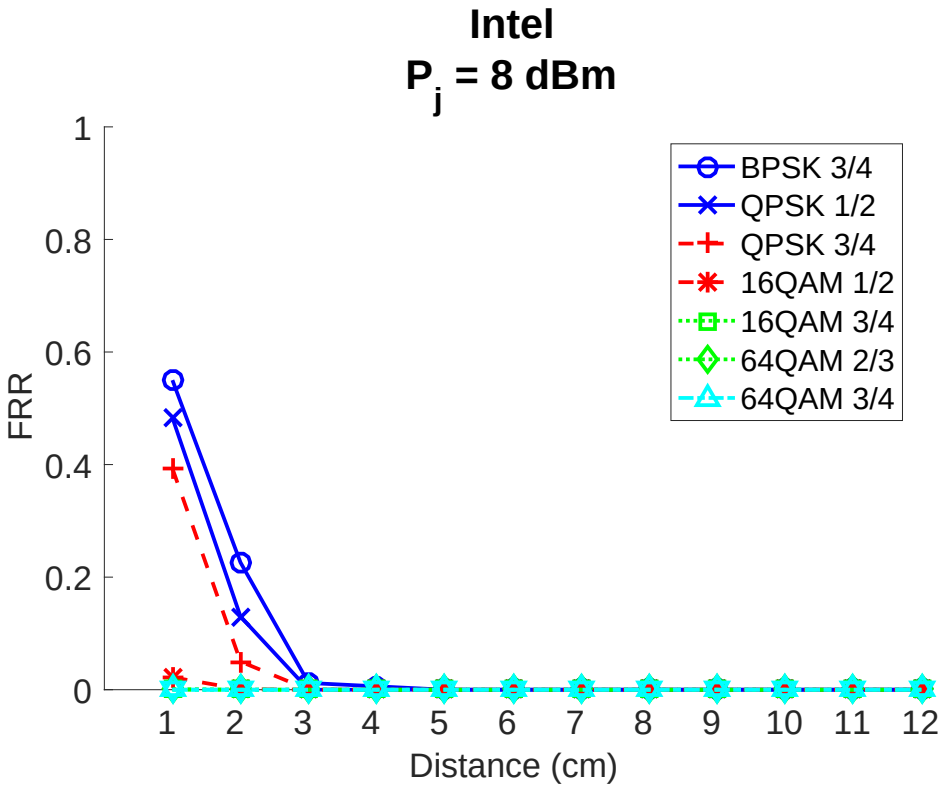

Figure C.9: Intel FRR with $P_j = 8$ dBm.

C.4 Edimax

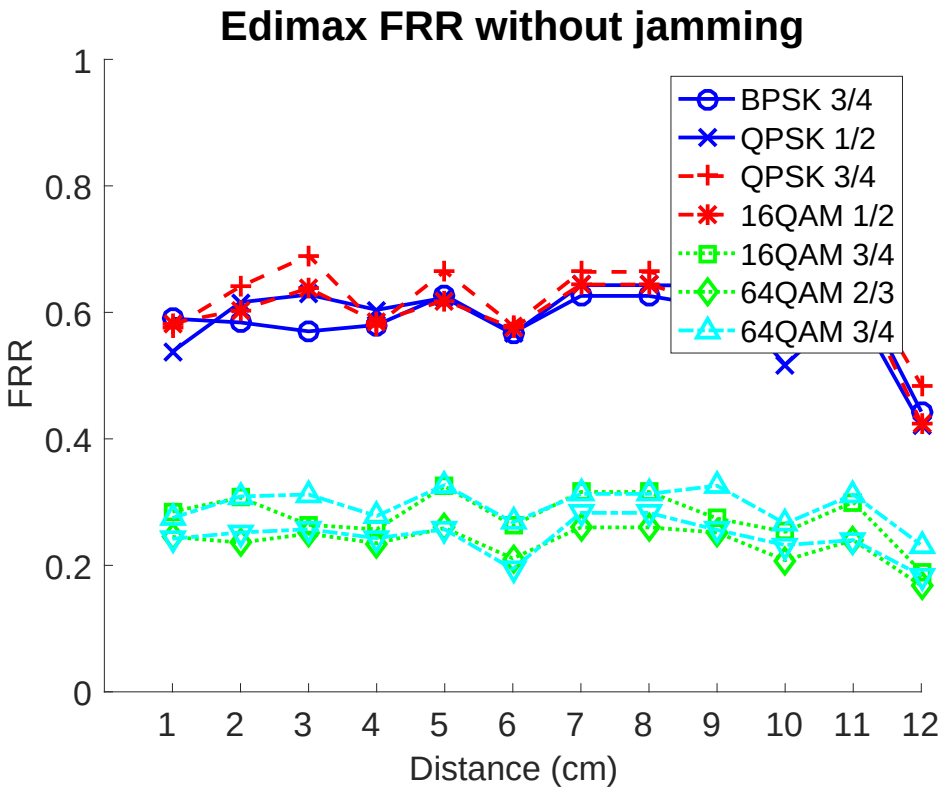

Figure C.10: Edimax FRR with no jamming.

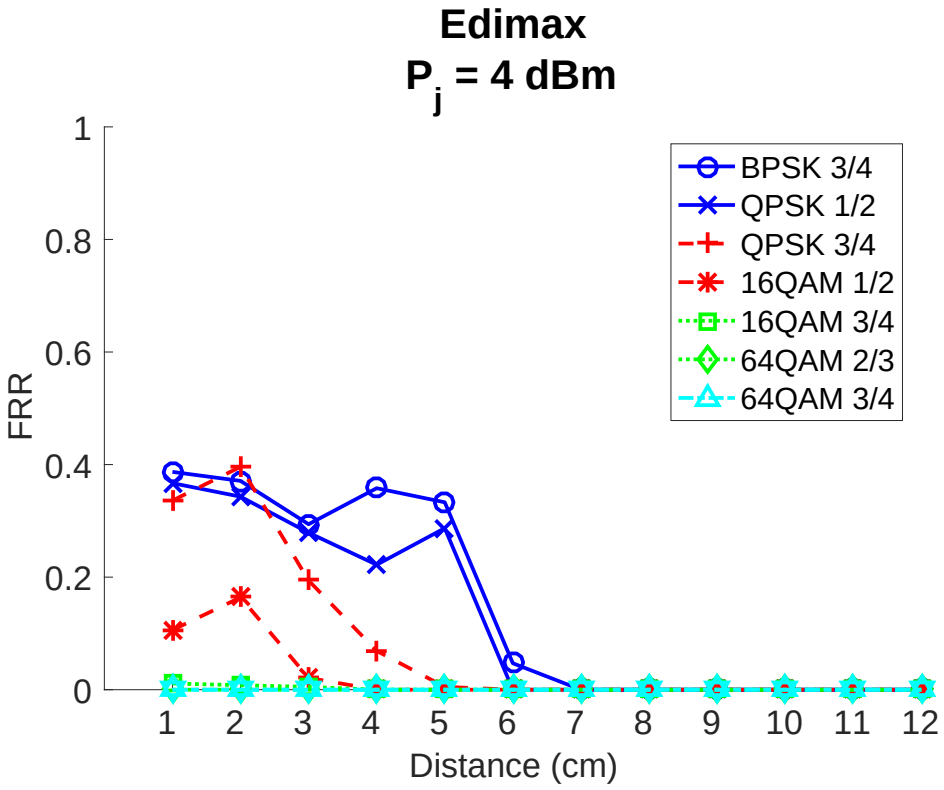

Figure C.11: Edimax FRR with $P_j = 4$ dBm.

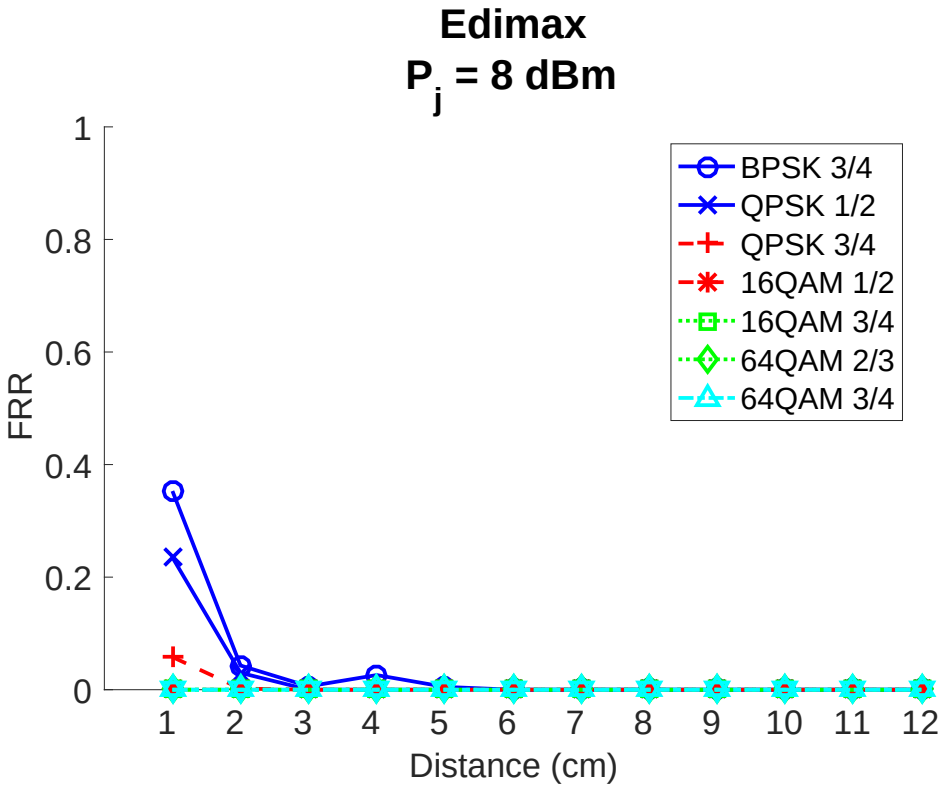

Figure C.12: Edimax FRR with $P_j = 8$ dBm.

Bibliography

[1] A & D Medical. A & D Medical UA-767PC blood pressure monitor. Online at https://medical.andonline.com/home, visited 4/15/2018. Citation on page 38.

[2] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Z. Béguelin, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 5–17. ACM, 2015. DOI 10.1145/2810103.2813707. Citation on page 168.

[3] H. Al-Mefleh and O. Al-Kofahi. Taking advantage of jamming in wireless networks: A survey. *Computer Networks*, 99:99–124, 2016. DOI doi.org/10.1016/j.comnet.2016.02.011. Citation on page 169.

[4] Alfa Networks. Alfa Networks AWUS036H Wi-Fi adapter. Online at http://www.alfa.com.tw, visited 4/15/2018. Citation on pages 22, 38, 61, and 157.

[5] Altelix LLC. 2.4 GHz 15 dBi WiFi Parabolic Antenna. Online at http://www.altelix.com/2-4-GHz-15-dBi-Grid-Antenna-p/ag24g15.htm, visited 3/22/2018. Citation on page 131.

[6] Apple Inc. iWatch. Online at https://www.apple.com/watch/, visited 3/22/2018. Citation on page 179.

[7] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz. On the reliability of wireless fingerprinting using clock skews. In *Proceedings of the ACM Conference on Wireless Network Security (WiSec)*, Mar. 2010. DOI 10.1145/1741866.1741894. Citation on page 134.

[8] A. Araujo, J. Blesa, E. Romero, and O. Nieto-Taladriz. Cooperative jam technique to increase physical-layer security in CWSN. In *International Conference on Advances in Cognitive Radio (COCORA)*, pages 11–14, 2012. Citation on page 170.

[9] C. A. Balanis. *Antenna Theory: Analysis and Design*. Wiley, third edition, 2005. Citation on pages 46, 47, 103, 105, 106, 107, 109, 110, 118, 143, 144, and 189.

[10] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Symposium on Network and Distributed Systems Security (NDSS)*, 2002. Citation on pages 6 and 183.

[11] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988. Citation on page 59.

[12] Berg Insight. Wireless IoT connectivity technologies and markets. Online at http://www.berginsight.com/ReportPDF/ProductSheet/bi-wirelessIoT-ps.pdf, visited 3/25/2018. Citation on page 3.

[13] P. Biondi. Scapy. Online at http://www.secdev.org/projects/scapy, visited 4/15/2018. Citation on pages 22 and 120.

[14] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. An IEEE 802.11a/g/p OFDM receiver for GNU Radio. In *Proceedings of the Workshop on Software Radio Implementation Forum (SRIF)*, pages 9–16. ACM, 2013. DOI 10.1145/2491246.2491248. Citation on pages 114 and 157.

[15] Bluetooth Special Interest Group. Bluetooth Secure Simple Pairing. Online at https://www.bluetooth.org/, visited 4/15/2018. Citation on page 54.

[16] A. Brown, R. Mortier, and T. Rodden. Multinet: Reducing interaction overhead in domestic wireless networks. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, Apr. 2013. DOI 10.1145/2470654.2466208. Citation on pages 5, 58, and 69.

[17] A. J. B. Brush, J. Jung, R. Mahajan, and J. Scott. Homelab: Shared infrastructure for home technology field studies. In *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp)*, pages 1108–1113. ACM, 2012. Citation on page 53.

[18] L. Cai, K. Zeng, H. Chen, and P. Mohapatra. Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas. In *Proceedings of Network and Distributed Systems Security Symposium (NDSS)*, 2011. Citation on pages 16, 47, and 60.

[19] Q. Chen, K. Ozawa, Q. Yuan, and K. Sawaya. Antenna characterization for wireless power-transmission system using near-field coupling. *IEEE Antennas and Propagation Magazine*, 54(4):108–116, Aug. 2012. DOI 10.1109/map.2012.6309161. Citation on page 106.

[20] W. Cheng, K. Tan, V. Omwando, J. Zhu, and P. Mohapatra. RSS-Ratio for enhancing performance of RSS-based applications. In *Proceedings of IEEE*

International Conference on Computer Communications (INFOCOM), pages 3075–3083. IEEE, 2013. Citation on page 25.

[21] J. Classen, M. Schulz, and M. Hollick. Practical covert channels for WiFi systems. In *Conference on Communications and Network Security (CNS)*, pages 209–217. IEEE, Sept. 2015. DOI 10.1109/cns.2015.7346830. Citation on page 133.

[22] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz. A wearable system that knows who wears it. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (Mobisys)*, pages 55–67. ACM, 2014. Citation on pages 174 and 180.

[23] V. Coskun, B. Ozdenizci, and K. Ok. A survey on Near Field Communication (NFC) technology. *Wireless Personal Communications*, 71(3):2259–2294, 2013. DOI 10.1007/s11277-012-0935-5. Citation on page 135.

[24] B. Danev, D. Zanetti, and S. Capkun. On physical-layer identification of wireless devices. *ACM Computing Surveys (CSUR)*, 45(1):6, 2012. Citation on page 37.

[25] DigitSole. DigitSole Smart Shoes. Online at http://www.digitsole.com, visited 4/15/2018. Citation on page 3.

[26] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor. Cooperative jamming for wireless physical layer security. In *IEEE Workshop on Statistical Signal Processing*, pages 417–420. IEEE, 2009. Citation on page 170.

[27] Dyn Corporation. Oracle + Dyn. Online at https://dyn.com/, visited 4/15/2018. Citation on page 8.

[28] Edimax Techology Company, Ltd. EW-7811Un. Online at http://us.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/us/wireless_adapters_n150/ew-7811un/, visited 3/22/2018. Citation on page 157.

[29] W. K. Edwards, R. E. Grinter, R. Mahajan, and D. Wetherall. Advancing the state of home networking. *Communications of the ACM*, 54(6):62–71, 2011. Citation on page 5.

[30] Ettus Research. USRP N210 Software Defined Radio. Online at https://www.ettus.com/product/details/UN210-KIT, visited 3/18/2018. Citation on pages 112 and 157.

[31] Federal Communications Commission. Section 15.247 Operation within the bands 902-928 MHz, 2400-2485.3 MHz, and 5725-5850 MHz. Online at https://www.gpo.gov/fdsys/pkg/CFR-2013-title47-vol1/pdf/CFR-2013-title47-vol1-sec15-247.pdf, visited 3/22/2018. Citation on page 130.

[32] J. Finkle. J & J warns diabetic patients: Insulin pump vulnerable to hacking. Online at https://www.reuters.com/article/us-johnson-johnson-cyber-insulin-pumps-e/jj-warns-diabetic-patients-insulin-pump-vulnerable-to-hacking-idUSKCN12411L, visited 1/24/2018. Citation on page 4.

[33] Fitbit. Fitbit Aria Smart Scale. Online at https://www.fitbit.com/aria2, visited 4/8/2018. Citation on page 55.

[34] M. Fomichev, F. Alvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick. Survey and systematization of secure device pairing. *IEEE Communications Surveys & Tutorials*, 2017. Citation on page 10.

[35] Fora Care. D30 blood pressure monitor. Online at http://www.foracare.com/Blood-Pressure-D30.html, visited 4/15/2018. Citation on page 61.

[36] I. Gelfand and M. Saul. Trigonometric identities. In *Trigonometry*, pages 139–171. Springer, 2001. Citation on page 78.

[37] S. Goel and R. Negi. Guaranteeing secrecy using artificial noise. *IEEE Transactions on Wireless Communications*, 7(6):2180–2189, 2008. Citation on page 170.

[38] A. Goldsmith. *Wireless communications*. Cambridge University Press, 2005. Citation on pages xiv, 86, 87, 88, 146, 147, 149, 150, and 151.

[39] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In *Proceedings of the USENIX Security Symposium*, 2011. Citation on page 59.

[40] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. *SIGCOMM Computer Communication Review*, 41(4):2–13, Aug. 2011. DOI 10.1145/2018436.2018438. Citation on pages 165 and 170.

[41] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 10–10. IEEE, 2006. Citation on page 58.

[42] D. Goodwin. 9 baby monitors wide open to hacks that expose users' most private moments. Online at http://arstechnica.com/security/2015/09/9-baby-monitors-wide-open-to-hacks-that-expose-users-most-private-moments/, visited 4/15/2018. Citation on page 5.

[43] D. Goodwin. Record-breaking DDoS reportedly delivered by >145k hacked cameras. Online at https://arstechnica.com/information-technology/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/, visited 4/15/2018. Citation on page 8.

[44] Google. Share content by NFC with Android Beam. Online at https://support.google.com/nexus/answer/2781895?hl=en, visited 3/22/2018. Citation on page 135.

[45] GoPro. GoPro cameras. Online at http://gopro.com/, visited 4/15/2018. Citation on page 55.

[46] G. M. Graff. How a dorm room Minecraft scam brought down the Internet. Online at https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/, visited 3/22/2018. Citation on page 8.

[47] Great Scott Gadgets. HackRF One. Online at https://greatscottgadgets.com/hackrf/, visited 3/22/2018. Citation on page 130.

[48] A. Greenberg. Hackers remotely kill a Jeep on the highway – with me in it. Online at http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, visited 4/15/2018. Citation on page 5.

[49] R. E. Grinter, W. K. Edwards, M. Chetty, E. S. Poole, J.-Y. Sung, J. Yang, A. Crabtree, P. Tolmie, T. Rodden, C. Greenhalgh, and S. Benford. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Transactions on Computer-Human Interaction (CHI)*, 16(2), June 2009. DOI 10.1145/1534903.1534905. Citation on page 5.

[50] R. E. Grinter, W. K. Edwards, M. W. Newman, and N. Ducheneaut. The work to make a home network work. In *Proceedings of the European Conference on Computer-Supported Cooperative Work (ECSCW)*, pages 469–488. Springer, 2005. DOI 10.1007/1-4020-4023-7_24. Citation on page 5.

[51] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. 802.11 with multiple antennas for dummies. *SIGCOMM Computer Communication Review*, 40(1):19–25, Jan. 2010. DOI 10.1145/1672308.1672313. Citation on pages 21, 87, 152, and 156.

[52] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with Channel State Information. *SIGCOMM Computer Communication Review*, 41(1):53, Jan. 2011. DOI 10.1145/1925861.1925870. Citation on pages 90, 92, and 112.

[53] J. Hester, T. Peters, T. Yun, R. Peterson, J. Skinner, B. Golla, K. Storer, S. Hearndon, K. Freeman, S. Lord, R. Halter, D. Kotz, and J. Sorber. Amulet: An energy-efficient, multi-application wearable platform. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 216–229. ACM Press, Nov. 2016. DOI 10.1145/2994551.2994554. Citation on pages 3 and 174.

[54] Hexoskin. Hexoskin Wearable Body Metrics. Online at https://www.hexoskin.com/, visited 4/15/2018. Citation on page 3.

[55] S. Hilton. Dyn analysis summary of Friday October 21 attack. Online at https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/, visited 11/17/2017. Citation on page 8.

[56] Institute of Electrical and Electronics Engineers. 802.11ac standard. Online at http://standards.ieee.org/about/get/802/802.11.html, visited 3/22/2018. Citation on page 140.

[57] Institute of Electrical and Electronics Engineers. 802.11n standard. Online at http://standards.ieee.org/findstds/standard/802.11n-2009.html, visited 3/22/2018. Citation on pages 85, 96, 98, 112, 140, 141, and 171.

[58] Intel. Intel Ultimate N Wi-Fi Link 5300: Product Brief. Online at http://www.intel.com/content/www/us/en/wireless-products/ultimate-n-wifi-link-5300-brief.html, visited 4/15/2018. Citation on pages 92, 119, and 157.

[59] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the International Conference on Mobile Computing and Networking (Mobicom)*, pages 321–332. ACM, 2009. Citation on pages 21, 23, and 59.

[60] C. Javali, G. Revadigar, L. Libman, and S. Jha. SeAK: Secure authentication and key generation protocol based on dual antennas for wireless body area networks. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pages 74–89. Springer, 2014. Citation on page 60.

[61] I. R. Jenkins, R. Shapiro, S. Bratus, T. Goodspeed, R. Speers, and D. Dowd. Speaking the local dialect: exploiting differences between IEEE 802.15.4 receivers with commodity radios for fingerprinting, targeted attacks, and WIDS evasion. In *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, pages 63–68, 2014. Citation on page 37.

[62] M. O. Jewell, E. Costanza, and J. Kittley-Davies. Connecting things to the Internet: an evaluation of four configuration strategies for Wi-Fi devices with minimal user interfaces. In *Proceedings of the ACM International Joint Conference*

on Pervasive and Ubiquitous Computing (UbiComp), pages 767–778. ACM, 2015. Citation on page 58.

[63] JR Raphael. Android nostalgia: 13 once-trumpeted features that quietly faded away. Online at https://www.computerworld.com/article/3239864/android/android-nostalgia-old-features.html, visited 3/22/2018. Citation on page 136.

[64] L. Jun, J. H. Andrian, and C. Zhou. Bit error rate analysis of jamming for OFDM systems. In *Wireless Telecommunications Symposium*, pages 1–8. IEEE, Apr. 2007. DOI 10.1109/wts.2007.4563327. Citation on page 147.

[65] F. Kawsar and A. B. Brush. Home computing unplugged: Why, where and when people use different connected devices at home. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 627–636. ACM, 2013. Citation on page 53.

[66] S. L. Keoh, S. S. Kumar, and H. Tschofenig. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal*, 1(3):265–275, June 2014. DOI 10.1109/jiot.2014.2323395. Citation on page 5.

[67] F. Koehler, S. Winkler, M. Schieber, U. Sechtem, K. Stangl, M. Böhm, H. Boll, S. S. Kim, K. Koehler, S. Lücke, M. Honold, P. Heinze, T. Schweizer, M. Braecklein, B.-A. Kirwan, G. Gelbrich, S. D. Anker, and on behalf of the TIM-HF Investigators. Telemedical Interventional Monitoring in Heart Failure (TIM-HF), a randomized, controlled intervention trial investigating the impact of telemedicine on mortality in ambulatory patients with heart failure: Study design. *European Journal of Heart Failure*, pages 1354–1362, 2010. Citation on page 52.

[68] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84, 2017. DOI 10.1109/mc.2017.201. Citation on page 8.

[69] B. Krebs. Krebs On Security hit with record DDoS. Online at https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/, visited 11/17/2017. Citation on page 8.

[70] C. Kuo, M. Luk, R. Negi, and A. Perrig. Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 233–246. ACM, 2007. Citation on page 170.

[71] X. Liang, T. Yun, R. Peterson, and D. Kotz. LightTouch: Securely connecting wearables to ambient displays with user intent. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9, May 2017. DOI 10.1109/INFOCOM.2017.8057210. Citation on page 168.

[72] C.-H. Liu. On the design of OFDM signal detection algorithms for hardware implementation. In *Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 596–599. IEEE, Dec. 2003. DOI 10.1109/glocom.2003.1258308. Citation on pages 114 and 117.

[73] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam. ProxiMate: Proximity-based secure pairing using ambient wireless signals. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 211–224. ACM, 2011. DOI 10.1145/1999995.2000016. Citation on pages 59 and 171.

[74] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, 2009. Citation on page 58.

[75] M. L. Mazurek, J. P. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. F. Cranor, G. R. Ganger, and M. K. Reiter. Access control for home data sharing: Attitudes, needs and practices. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 645–654. ACM, 2010. DOI 10.1145/1753326.1753421. Citation on page 5.

[76] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proceedings of the IEEE Symposium on Security and Privacy (S & P)*, pages 110–124. IEEE, 2005. Citation on page 58.

[77] S. Mennicken and E. M. Huang. Hacking the natural habitat: An in-the-wild study of smart homes, their development, and the people who live in them. In *Pervasive Computing*, pages 143–160. Springer, 2012. Citation on page 53.

[78] P. Middleton, P. Kjeldsen, and J. Tully. Forecast: The Internet of Things, Worldwide, 2013. *Gartner Research*, https://www.gartner.com/doc/2625419/forecast-internet-things-worldwide, visited 4/15/2018. Citation on page 3.

[79] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 880–891. ACM, Nov. 2014. DOI 10.1145/2660267.2660334. Citation on page 168.

[80] MIT Technology Review. Finding Insecurity in the Internet of Things. Online at https://www.technologyreview.com/s/545661/finding-insecurity-in-the-internet-of-things/, visited 4/15/2018. Citation on page 5.

[81] National Instruments. What is I/Q Data? Online at http://www.ni.com/tutorial/4805/en/, visited 4/15/2018. Citation on pages 79 and 80.

[82] J. Nazario. The anatomy of an IoT botnet attack. Online at https://www.fastly.com/blog/anatomy-an-iot-botnet-attack/, visited 4/15/2018. Citation on page 9.

[83] R. Negi and S. Goel. Secret communication using artificial noise. In *IEEE Vehicular Technology Conference*, page 19. IEEE, 2005. Citation on page 170.

[84] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *IEEE Communications Surveys & Tutorials*, 3(3):2–12, 2000. Citation on page 21.

[85] C. Neumann, O. Heen, and S. Onno. An empirical study of passive 802.11 device fingerprinting. In *International Conference on Distributed Computing Systems Workshops*, pages 593–602. IEEE, June 2012. DOI 10.1109/icdcsw.2012.8. Citation on page 134.

[86] Panda Wireless. Panda Ultra Wireless N USB Wi-Fi adapter. Online at http://www.pandawireless.com/, visited 4/15/2018. Citation on pages 37, 120, and 157.

[87] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz. Demo: Wanda, securely introducing mobile devices. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, page 113. ACM Press, June 2016. DOI 10.1145/2938559.2938581. Citation on pages 13 and 183.

[88] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz. Wanda: securely introducing mobile devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, Apr. 2016. DOI 10.1109/INFOCOM.2016.7524366. Citation on pages 13, 171, and 183.

[89] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz. Wanda: securely introducing mobile devices – extended version. Technical Report TR2016-789, Dartmouth Computer Science, Feb. 2016. Expanded version of the INFOCOM 2016 paper by the same title., Online at http://www.cs.dartmouth.edu/reports/abstracts/TR2016-789/. Citation on page 13.

[90] T. J. Pierson, R. Rawassizadeh, R. Peterson, and D. Kotz. Secure information transfer between nearby wireless devices. *MobiCom S3 Workshop*, 90:100, 2017. Citation on pages 13 and 186.

[91] E. S. Poole. Is living with others a barrier to technical literacy? In *Proceedings of the International Conference on Supporting Group Work*, pages 178–181. ACM, 2014. Citation on page 53.

[92] E. S. Poole, M. Chetty, T. Morgan, R. E. Grinter, and W. K. Edwards. Computer help at home: Methods and motivations for informal technical support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 739–748. ACM, 2009. Citation on page 53.

[93] E. S. Poole, W. K. Edwards, and L. Jarvis. The home network as a socio-technical system: Understanding the challenges of remote home network problem diagnosis. *Computer Supported Cooperative Work (CSCW)*, 18(2-3):277–299, 2009. Citation on page 53.

[94] G. Press. Internet Of Things (IoT) Predictions. *Forbes Inc.*, https://www.forbes.com/sites/gilpress/2016/01/27/internet-of-things-

iot-predictions-from-forrester-machina-research-wef-gartner-idc/, visited 4/15/2018. Citation on page 1.

[95] H. Rahbari and M. Krunz. Exploiting frame preamble waveforms to support new physical-layer functions in OFDM-based 802.11 systems. *IEEE Transactions on Wireless Communications*, 16(6):3775–3786, June 2017. DOI 10.1109/twc.2017.2688405. Citation on page 134.

[96] T. S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 2002. Citation on pages 19, 21, and 29.

[97] Raspberry Pi Foundation. Raspberry Pi. Online at http://www.raspberrypi.org, visited 4/15/2018. Citation on pages 9 and 37.

[98] R. Rawassizadeh, T. J. Pierson, R. Peterson, and D. Kotz. NoCloud: Exploring network disconnection through on-device data analysis. *IEEE Pervasive Computing*, 17(1):64–74, 2018. DOI 10.1109/MPRV.2018.011591063. Citation on page 4.

[99] A. D. Rayome. DDoS attacks increased 91% in 2017 thanks to IoT. Online at https://www.techrepublic.com/article/ddos-attacks-increased-91-in-2017-thanks-to-iot, visited 4/15/2018. Citation on page 9.

[100] K. Ren, Q. Wang, D. Ma, and X. Jia. Securing emerging short range wireless communications: the state of the art. *IEEE Wireless Communications*, 21(6):153–159, Dec. 2014. DOI 10.1109/mwc.2014.7000983. Citation on page 135.

[101] R. Riemsma, I. C. Ramos, R. Birnie, N. Büyükkaramikli, N. Armstrong, S. Ryder, S. Duffy, G. Worthy, M. Al, J. Severens, and Others. Integrated sensor-augmented pump therapy systems for managing blood glucose levels in

type 1 diabetes: A systematic review and economic evaluation. *Health technology assessment*, 20(17):1, 2016. Citation on pages 2 and 4.

[102] K. Rose, S. Eldridge, and L. Chapin. The Internet of Things (IoT): An Overview – Understanding the issues and challenges of a more connected world, Oct. 2015. Online at https://www.internetsociety.org/resources/doc/2015/iot-overview. Citation on pages 3 and 4.

[103] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *Proceedings of the IEEE Symposium on Security and Privacy (S & P)*, pages 6–pp. IEEE, 2006. Citation on page 58.

[104] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel: Design and usability study. *Proceedings of the IEEE Symposium on Information Forensics and Security*, 6(1):28–38, March 2011. Citation on page 58.

[105] J. N. Sayles, G. W. Ryan, J. S. Silver, C. A. Sarkisian, and W. E. Cunningham. Experiences of social stigma and implications for healthcare among a diverse population of HIV positive adults. *Journal of Urban Health*, 84(6):814–828, 2007. Citation on pages 4 and 54.

[106] T. M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for OFDM. *Transactions on Communications*, 45(12):1613–1621, Dec. 1997. DOI 10.1109/26.650240. Citation on page 114.

[107] B. Schneier. The Internet of Things is wildly insecure—and often unpatchable. *Schneier on Security*, June, 2014. Citation on page 5.

[108] M. Sethi, E. Oat, M. Di Francesco, and T. Aura. Secure bootstrapping of cloud-managed ubiquitous displays. In *Proceedings of the ACM International Joint*

Conference on Pervasive and Ubiquitous Computing (Ubicomp), pages 739–750. ACM, 2014. Citation on page 58.

[109] A. Sheikholeslami, D. Goeckel, and H. Pishro-Nik. Jamming based on an ephemeral key to obtain everlasting security in wireless environments. *IEEE Transactions on Wireless Communications*, 14(11):6072–6081, Nov 2015. Citation on page 170.

[110] W. Shen, P. Ning, X. He, and H. Dai. Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time. In *IEEE Symposium on Security and Privacy (S & P)*, pages 174–188, May 2013. DOI 10.1109/SP.2013.22. Citation on page 170.

[111] Y. Shen and E. Martinez. Channel estimation in OFDM systems. *Freescale semiconductor application note*, pages 1–15, 2006. Citation on pages 110 and 112.

[112] L. Shi, M. Li, S. Yu, and J. Yuan. BANA: Body area network authentication exploiting channel characteristics. *IEEE Journal on Selected Areas in Communications*, 31(9):1803–1816, 2013. Citation on page 59.

[113] L. Shi, J. Yuan, S. Yu, and M. Li. ASK-BAN: Authenticated secret key extraction utilizing channel characteristics for body area networks. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, pages 155–166. ACM, 2013. Citation on page 59.

[114] Shodan. Shodan.io. Online at https://www.shodan.io/, visited 4/15/2018. Citation on page 9.

[115] S. W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, CA, 1997. Citation on page 75.

[116] S. W. Smith. *The Internet of Risky Things: Trusting the devices that surround us*. O'Reilly, 2017. Online at http://www.worldcat.org/isbn/9781491963623. Citation on page 5.

[117] C. Soriente, G. Tsudik, and E. Uzun. BEDA: Button-enabled device association. *UbiComp Workshop Proceedings: International Workshop on Security for Spontaneous Interaction (IWSSI)*, 2007. Citation on page 58.

[118] C. Soriente, G. Tsudik, and E. Uzun. HAPADEP: human-assisted pure audio device pairing. *Information Security*, pages 385–400, 2008. Citation on page 58.

[119] F. Stajano and R. Anderson. The Resurrecting Duckling: security issues for ubiquitous computing. *Computer*, 35(4):22–26, Apr. 2002. DOI 10.1109/mc.2002.1012427. Citation on pages 7 and 58.

[120] Symantect Security Response. IoT devices being increasingly used for DDoS attacks. Online at https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks, visited 4/15/2018. Citation on page 9.

[121] Symantect Security Response. Mirai: what you need to know about the botnet behind recent major DDoS attacks. Online at https://www.symantec.com/connect/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-attacks, visited 4/15/2018. Citation on page 8.

[122] X. Tang, R. Liu, P. Spasojević, and H. V. Poor. Interference assisted secret communication. *IEEE Transactions on Information Theory*, 57(5):3153–3167, 2011. Citation on page 170.

[123] TechTarget. Distributed Denial of Service attack definition. Online at http://searchsecurity.techtarget.com/definition/

distributed-denial-of-service-attack, visited 4/15/2018. Citation on page 8.

[124] Tektronix. What's your IQ – about quadrature signals. Online at http://www.tek.com/blog/whats-your-iq-about-quadrature-signals, visited 4/15/2018. Citation on page 81.

[125] The GNU Radio Foundation. GNURadio. Online at https://www.gnuradio.org/, visited 3/22/2018. Citation on page 112.

[126] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun. On limitations of friendly jamming for confidentiality. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 160–173, May 2013. Citation on pages 165 and 166.

[127] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, 2005. Citation on pages 76, 90, 91, 99, 101, 102, 113, 129, 163, 164, and 165.

[128] B. Ur, J. Jung, and S. Schechter. The current state of access control for smart devices in homes. In *Proceedings of the Workshop on Home Usable Privacy and Security (HUPS)*, 2013. Citation on page 5.

[129] J. P. Vilela, M. Bloch, J. Barros, and S. W. McLaughlin. Friendly jamming for wireless secrecy. In *IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2010. Citation on page 170.

[130] J. P. Vilela, M. Bloch, J. Barros, and S. W. McLaughlin. Wireless secrecy regions with friendly jamming. *IEEE Transactions on Information Forensics and Security*, 6(2):256–266, 2011. Citation on page 170.

[131] watchcases.com. What is the average wrist size and how do I measure my wrist? Online at http://www.watchcases.com/wrist-watches--wrist-size.html, visited 11/17/2017. Citation on pages 175 and 176.

[132] Wireshark. How to decrypt 802.11. Online at https://wiki.wireshark.org/HowToDecrypt802.11, visited 4/8/2018. Citation on page 54.

[133] T. Xin, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou. FreeSense: Indoor human identification with Wi-Fi signals. In *Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2016. DOI 10.1109/GLOCOM.2016.7841847. Citation on page 179.

[134] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1862–1870, April 2011. Citation on page 171.

[135] Q. Xu, R. Zheng, W. Saad, and Z. Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, Jan. 2016. DOI 10.1109/comst.2015.2476338. Citation on page 134.

[136] D. Young. Radiotap. Online at http://www.radiotap.org/, visited 11/17/2017. Citation on page 22.

[137] K. Zeng, D. Wu, A. Chan, and P. Mohapatra. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, 2010. Citation on page 59.

[138] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang. Proximity based IoT device authentication. *IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2017. Citation on page 168.

[139] R. Zhou and G. Xing. nShield: A noninvasive NFC security system for mobile devices. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 95–108. ACM, 2014. DOI 10.1145/2594368.2594376. Citation on pages 135 and 168.

[140] X. Zhou and M. R. McKay. Physical layer security with artificial noise: Secrecy capacity and optimal power allocation. In *International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–5. IEEE, 2009. Citation on page 170.