Physical Layer Attacks with Malicious Full-Duplex Relays and Their Defense Strategies

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Xingya Zhao, B.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2024

Dissertation Committee:

Dr. Kannan Srinivasan Athreya, Adviser

Dr. Jennifer Bogner

Dr. Zhiqiang Lin

Dr. Srinivasan Parthasarathy

Abstract

The widespread adoption of wireless communication technologies underscores the need to ensure security in these systems. Within wireless communications, channel measurement plays a critical role in enabling successful communication. Additionally, the rapid uncorrelation over space of wireless channel makes it an ideal source for various physical layer security applications, such as secret key generation and source authentication protocols. However, existing research has demonstrated that a malicious full-duplex relay, which receives and retransmits signals almost simultaneously at the same frequency band, can manipulate the receivers' channel estimations by actively relaying the pilot signals used for channel measurement.

This thesis aims to investigate novel attacks involving malicious full-duplex relays and explore defenses against these attacks. The thesis focuses on two specific works. The first work concentrates on defending against malicious amplify-and-forward full-duplex relays. To address the emerging threat posed by full-duplex relay attackers to physical-layer wireless security protocols, we propose RelayShield, a system designed to detect such malicious relays and recover the channels manipulated by them. Unlike previous approaches that rely on previously-collected signature channels, RelayShield analyzes signal path information derived from input channels to detect relays and recover channels. RelayShield achieves over 95% detection accuracy with channels collected in two typical indoor environments. The recovered channels can support a wide range of applications.

The second work focuses on the vulnerabilities of the channel estimation process in downlink Multi-User MIMO (MU-MIMO) transmissions. While MU-MIMO technology offers significant benefits, it also opens avenues for potential attacks. In this work, we propose an active eavesdropping attack targeting downlink MU-MIMO transmissions. The attack consists of two phases. First, the attacker sends a forged pilot packet to the victims. After that, the access point transmits streams intended for victims to the attacker, who operates in full-duplex mode and relays the streams to the victims. Compared to existing eavesdropping attacks targeting downlink MU-MIMO transmissions, our proposed attack requires less prior knowledge and coordination from attackers and maximizes eavesdropping opportunities. We evaluate the proposed attack in various settings and prove its effectiveness with multiple victims and partial channel knowledge. Additionally, we explore the use of physical-layer features to detect our proposed attack. Future work about how this attack model can be extended to compromise uplink MU-MIMO transmissions, and how the attackers can potentially adjust their attack strategies to bypass some countermeasures is also discussed.

Dedicated to my family.

Acknowledgments

I am deeply thankful to all those who have supported and guided me throughout the course of my thesis work. First and foremost, I would like to express my deepest gratitude to my advisor Prof. Kannan Srinivasan Athreya. His guidance has been incredibly valuable to me in my research journey. His intellect and kindness have not only supported me throughout this journey but also inspired me to pursue excellence and kindness in my own pursuits. I feel very fortunate to have had the opportunity to work under his mentorship.

I would like to extend my sincere thanks to my thesis committee members Prof. Zhiqiang Lin, Prof. Srinivasan Parthasarathy, and Prof. Jennifer Bogner, as well as Prof. Yang Wang in my candidacy exam committee for their continued support and insightful feedback. Their suggestions have greatly improved the quality of this work.

I am immensely grateful to all my lab fellows, whose unwavering support, collaboration, and friendship have made this journey memorable. I am fortunate to have collaborated with Dr. Avishek Banerjee, Dr. Arjun Bakshi, Dr. Wei-Han Chen, Vishnu Chhabra, Anwesha Roy, and Dr. Jiaqi Xu. This thesis work would never have been possible without their invaluable assistance and the countless hours spent discussing ideas and troubleshooting experiments. Additionally, I extend my appreciation to Dr. Lu Chen, Dr. Ananya Mahanti, Dr. Yifan Mao, Rashid Sowah, Dr. Wei Sun, Ishtiaque Ahmed Showmik, Dr. Fei Wu, and Dr. Ouyang Zhang for creating a supportive and stimulating environment that has greatly contributed to my research experience.

Last but not least, I would like to thank my family and friends for their unwavering support and understanding throughout these years. I owe a special debt of gratitude to my parents Naxin Sun and Baijie Zhao. I cannot thank them enough for their continued love, support, and for instilling in me the values of integrity and resilience. Any of my achievements would not have been possible without them. I would also like to thank my cat Heihei Zhao for being a great cat and for her assistance in my research. While I worked from home during the pandemic, she helped create dynamic multipath-rich environments for some experiments by playing with antennas. She has also helped write research papers and this thesis by occasionally stepping on the keyboard. I cherish every moment we have spent together and hope we can have many more in the future.

Vita

2016 .... B.S. Electronics and Electric Engineering, Shanghai Jiao Tong University, Shanghai, China

Publications

Research Publications

A. Banerjee (co-primary author), X. Zhao (co-primary author), V. Chhabra, K. Srinivasan, S. Parthasarathy “HORCRUX: Accurate cross band channel prediction”. *International Conference on Mobile Computing and Networking (MobiCom)*, 2024

X. Zhao, A. Roy, A. Banerjee, K. Srinivasan “*Fewer demands, more chances*: Active eavesdropping in MU-MIMO systems”. *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2024

X. Zhao, W.-H. Chen, K. Srinivasan “Malicious relay detection and legitimate channel recovery”. *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2023

J. Xu (co-primary author), X. Zhao (co-primary author), A. Bakshi, K. Srinivasan “Learning-based radio fingerprinting for RFID secure authentication scheme”. *IEEE Conference on Communications and Network Security (IEEE CNS)*, 2022

Z. Zhao, J. Wang, X. Zhao, C. Peng, Q. Guo, B. Wu “NaviLight: Indoor localization and navigation under arbitrary lights”. *IEEE International Conference on Computer Communications (INFOCOM)*, 2017

Z. Yang, Y. Bao, C. Luo, X. Zhao, S. Zhu, C. Peng, Y. Liu, and X. Wang “ARTcode: Preserve art and code in any image”. *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2016

C. Yang, X. Zhao, Y. Yao, and B. Xia "Modeling and analysis for cache-enabled cognitive D2D communications in cellular networks". *IEEE Global Communications Conference (GLOBECOM)*, 2016

Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

List of Abbreviations

ACK	ACKnowledgment
AoA	Angle of Arrival
AP	Access Point
CDF	Cumulative Distribution Function
CFO	Carrier Frequency Offset
CP	Cyclic Prefix
CSI	Channel State Information
CTS	Clear-To-Send
EMD	Earth Mover's Distance
FDM	Frequency Division Multiplexing
LoS	Line-of-Sight
LTE	Long-Term Evolution
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Squared Error

MU-MIMO	Multi-User MIMO
MUSIC	MUltiple SIgnal Classification
NDP	Null Data Packet
NLoS	Non-Line-of-Sight
OFDM	Orthogonal Frequency-Division Multiplexing
PDF	Probability Density Function
RF	Radio Frequency
RFID	Radio Frequency IDentification
RSS	Received Signal Strength
RTS	Request-To-Send
SIFS	Short InterFrame Space
SINR	Signal-to-Interference-plus-Noise Ratio
TNR	True Negative Rate
TPR	True Positive Rate

List of Tables

Table **Page**

List of Figures

Figure **Page**

Chapter 1: Introduction

Wireless communications have been used widely and revolutionized the way people connect and interact with each other. However, their widespread use has also given rise to new security challenges and makes the need for robust security measures more critical than ever. Cryptographic techniques, while essential, are not always sufficient to counter the attacks targeting wireless networks. In protocols sensitive to overheads, some packets, such as certain control packets, might not be encrypted for efficiency considerations and are vulnerable to interception. Moreover, security solutions based on current cryptographic methods could be compromised with future advancements in computational power and cryptanalysis.

Physical-layer security offers a promising solution by providing an additional layer of defense that complements traditional cryptographic techniques. Physical layer security leverages the inherent physical properties of the communication medium such as wireless channel measurements. In wireless communications, channel measurement plays a fundamental role in ensuring successful transmissions. Only after correctly estimating the channel from the transmitter, the receiver can remove the effect of wireless channels from its received signals and recover the original signal sent by the transmitter. What's more, the rapid uncorrelation over space of wireless channels makes them an ideal source for various physical layer security applications, such as secret key generation [9, 47, 48, 50, 54] and source authentication [31, 58, 84, 87, 45].

However, existing research has demonstrated that a malicious full-duplex relay is able to manipulate the receiver's estimated channels [73, 60]. A full-duplex relay can receive and retransmit signals almost simultaneously at the same frequency band [21]. If a malicious full-duplex relay is actively relaying the signals when the transmitter sends the pilot signals, the receiver will receive the pilot signals from both the legitimate transmitter and the relay, thus its estimated channels will contain an injected section representing the transmitter-relay-receiver channel. The communications can be easily interrupted if the malicious relay can selectively relay the pilot signals but not the signals for data transmission. Existing works have also applied them to attack physical layer security applications that take wireless channels are inputs [73, 60].

This thesis focuses on the physical layer attacks with malicious full-duplex relays and the defenses against them. In Chapter 2, we first introduce technical backgrounds, including brief introductions to existing full-duplex radio implementations and studies about full-duplex relays. In Chapter 3, we present RelayShield, a system designed to detect such malicious relays and recover the channels manipulated by them. Unlike previous approaches that rely on previously-collected signature channels, RelayShield analyzes signal path information derived from input channels to detect relays and recover channels. In Chapter 4, we introduce an active eavesdropping attack targeting downlink MU-MIMO transmissions that exploits the broadcasted pilot packets during channel sounding in MU-MIMO communication systems. Our attack utilizes a multi-antenna full-duplex device to modify the channel measurements of the victim client and compromise the security of data streams. Compared to existing eavesdropping attacks in MU-MIMO systems, this proposed attack does not require the attacker's devices to join the same communication as the victims and demands less prior knowledge and coordination. In Chapter 5, we discuss about how to extend the attack model proposed in Chapter 4 to compromise uplink MU-MIMO transmissions, and how the attacker can adjust the attack strategies to bypass some countermeasures using physical layer signatures. Chapter 6 concludes this thesis.

Chapter 2: Preliminaries

2.1 Full-Duplex Radio Implementations

Full-duplex devices are able to transmit and receive signals simultaneously at the same frequency bands. One of the main challenges in implementing full-duplex devices is canceling the self-interference caused by the transmitted signal at the receiving side. In recent years, various techniques have been proposed to tackle this challenge. In [29], the authors propose an antenna cancellation method in combination with analog and digital cancellations to cancel the self-interference. Reference [43] improves the analog cancellation technique with Balun transformers. Reference [23] further advances full-duplex implementations by separating outgoing and incoming signals with analog circuits and canceling non-linear distortions in digital cancellation. Figure 2.1 shows the diagram of a full-duplex radio.

While full-duplex communication has previously been associated with single-antenna systems, recent research has shown the feasibility of enabling full-duplex for Multiple-Input Multiple-Output (MIMO) devices. Compared with single-antenna systems, implementing full-duplex on MIMO devices poses an additional challenge of canceling interference across antennas. In [12], the authors propose the first MIMO full-duplex implementation MIDU, which primarily uses antenna cancellation with symmetric placement of transmit and receive

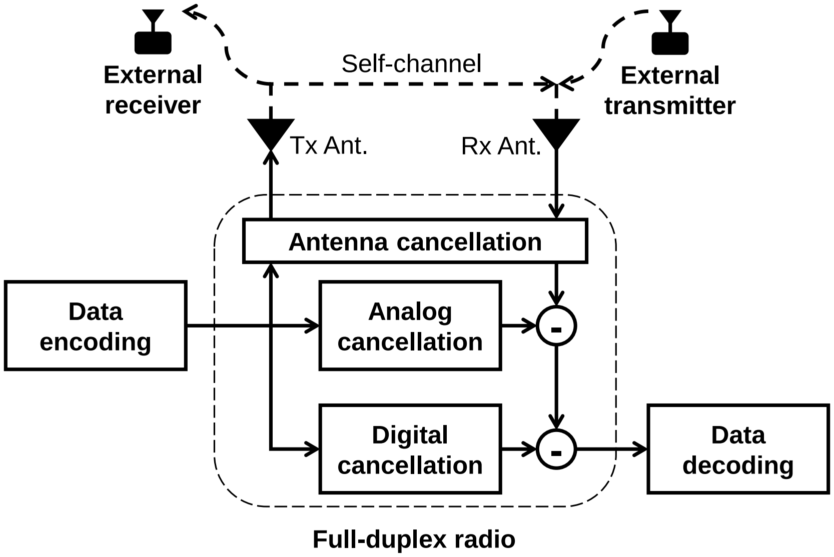


Figure 2.1: Diagram of a full-duplex radio

antennas. Reference [22] further improves MIMO full-duplex implementations which significantly reduce system complexity and achieve near-optimal cancellation results with novel digital estimation and cancellation algorithms.

2.2 Full-Duplex Relays

The advancements in full-duplex implementations have made full-duplex relays feasible [21, 27, 42, 61, 3, 28]. In order to utilize a full-duplex device as a relay, it is necessary for the device to amplify and retransmit the signals it has just received in the same frequency band. By employing efficient self-interference cancellation methods, the relaying process can be completed within a very short time, resulting in minimal latency during the forwarding process, typically a few sample durations [21]. Unlike regular full-duplex devices, a full-duplex relay sends its data stream from its receiving Radio Frequency (RF) chains instead of the internal logic, as shown in Figure 2.2.

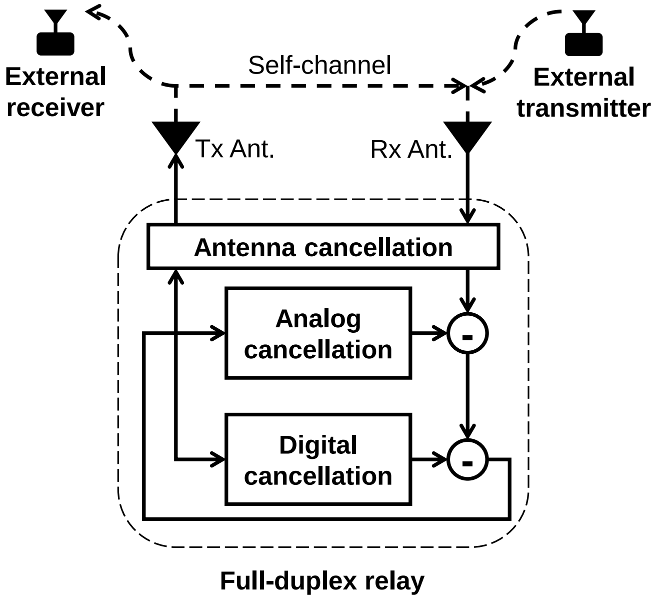


Figure 2.2: Diagram of a full-duplex relay

One major application of full-duplex relays is to enhance wireless communication performance. In [21], the authors design a relay system based on MIMO full-duplex devices. By employing a construct-and-forward filter, the relayed signal received at the destination can be constructively combined with signals directly from the transmitter, thereby extending the coverage area of wireless communication. In [27], the authors propose a system that enables in-band wireless cut-through transmission using multiple full-duplex relay devices and investigate the cancellation of inter-relay interference in the physical layer. Furthermore, in [28], the authors introduce a cluster of multiple full-duplex relays to enable end-to-end full-duplex communications and demonstrate the throughput improvement achieved by their system through extensive evaluations.

Apart from communication systems, full-duplex relays have found applications in other scenarios as well. For instance, they have been employed in protecting wireless sensing privacy [61], enabling untethered virtual reality [3], and reading remote Radio Frequency IDentification (RFID) tags with drones [51].

Chapter 3: Malicious Amplify-and-Forward Full-Duplex Relay Detection and Legitimate Channel Recovery

Full-duplex devices can compromise the integrity of wireless channel measurements through signal relaying and several attacks have been proposed based on this vulnerability. Existing source authentication methods relying on previously-collected signatures face significant challenges in detecting these attacks because a relay attacker can gradually inject the channels so that the manipulated channels will fall within the tolerance range of the authentication methods and are mistaken as new signatures. In this chapter, we introduce RelayShield, a system for detecting malicious relays and recovering the legitimate transmitter-receiver channels from the manipulated channels. RelayShield requires only one channel measurement at the receiver. It analyzes signal path information resolved from input channels to detect relays and recover channels. RelayShield achieves over 95% detection accuracy with channels collected in two typical indoor environments. The recovered channels can support a wide range of applications, including secret generation protocols and sensing systems.

3.1 Motivation

Physical-layer wireless security has been an emerging field due to the widespread use of wireless communication and its potential to enhance the security of communication systems. Wireless channels are considered to be uncorrelated at locations more than half a wavelength away because of the multipath propagation. The unique and random nature of wireless channels make them an ideal source for various physical layer security applications, such as secret key generation [9, 47, 48, 50, 54] and source authentication [31, 58, 84, 87, 45].

However, a new type of attack has been proposed that compromises these link-based security protocols with full-duplex relays [73, 60]. The relayed packets contain the same preambles as the legitimate packets, so the receiver will include their channels in its calculation of the channel responses. In these attacks, a full-duplex relay attacker actively relays legitimate packets while the receiver is collecting them for channel measurements. As a result, the receiver will include the relayed packets in its channel measurements and potentially allow the channel through attacker to dominate the measurement results when the relaying gain is large enough. After this channel injection phase, the attacker can replay the injected channels for identification attacks or use the injected channels to infer link-based shared secrets.

Existing physical-layer source authentication methods face difficulty in detecting the relay attackers. Both link-based methods [31, 58, 84, 87, 45] and hardware-based methods [32, 25, 41, 10] need to first measure signals from legitimate transmitters to build profiles. But it is unknown if injections have already started during the profile-building process. Additionally, to accommodate noise and environment changes, many authentication methods would have a tolerance for differences between input channels and signatures in profile when making authentication decisions, and they update the signatures periodically with the latest accepted channels. A relay attacker can take advantage of this mechanism by gradually increasing its amplification gain from zero so that the injected channels can pass the checks and be taken as new signatures.

To effectively defend against full-duplex attackers, it is necessary to address two questions: first, how to detect the existence of a relay attacker without relying on any previously-collected channel signatures; second, if a malicious relay is manipulating channels, is there a way to recover the legitimate channels instead of simply discarding the measurements and pausing all link-based security protocols or applications. To address these challenges, we propose RelayShield, a system for malicious relay detection and legitimate channel recovery. RelayShield takes advantage of the expected difference in delay and power loss between signals from the legitimate transmitter and through the relay attacker. It contains a relay detection module and a channel recovery module. The relay detection module uses a neural network to produce real-time results. The channel recovery module resolves multipath components that represent signal paths from input channels and reconstructs the legitimate channels with components from the legitimate transmitter. We conclude our contributions as follows.

- We propose a relay detection method without reliance on previously-collected signature channels. It achieves an accuracy of over 95% and can detect gradual channel injections.
- We propose a method to recover legitimate channels from measurements manipulated by relay attackers. The recovered channels are proven to support various applications.
- We improve the channel-to-signal-path techniques and apply them to enhance physical-layer wireless security.

3.2 Background

3.2.1 Attacks Employing Full-Duplex Relays

Full-duplex relays have garnered attention as a tool for attackers in compromising link-based security protocols due to their ability to manipulate the channel measurements perceived by the receiver. A typical attack scenario is depicted in Figure 3.1. Let us denote the genuine packet intercepted by the relay by x, the amplification factor by w, the carrier

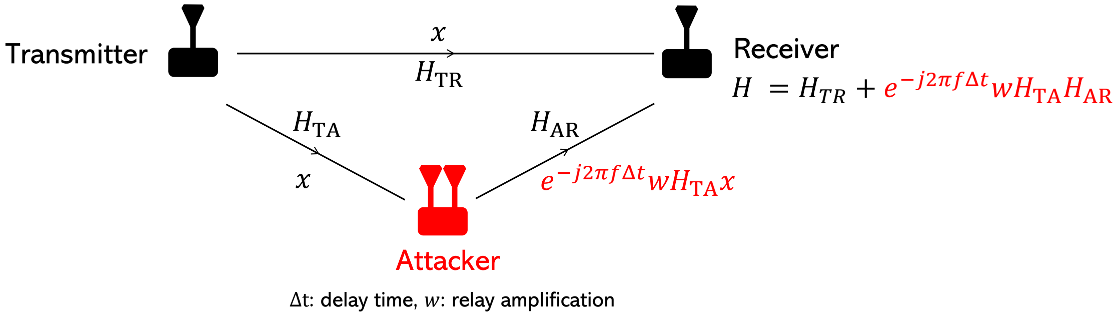

Transmitter
x
H_{TR}
Receiver
$H = H_{TR} + e^{-j2\pi f\Delta t} w H_{\text{TA}} H_{\text{AR}}$
H_{TA}
x
H_{AR}
$e^{-j2\pi f\Delta t} w H_{\text{TA}} x$
Attacker
Δt: delay time, w: relay amplification

Figure 3.1: General attack model with a malicious full-duplex relay.

frequency by f, the delay time introduced by the relay by Δt, and the transmitter-receiver, transmitter-relay, and relay-receiver channels by H_{TR}, H_{TA}, and H_{AR}, respectively.

In this scenario, the relayed signals carry the same preambles as the original signals, causing the receiver to interpret them as originating from the transmitter. As a result, the receiver calculates the channel as $H = H_{TR} + e^{-j2\pi f\Delta t} w H_{TA} H_{AR}$ with noise neglected, where $e^{-j2\pi f\Delta t} w H_{TA} H_{AR}$ is the component injected by the attacker, as highlighted in red in Figure 3.1. If w is sufficiently large, the injected component can dominate H. Given that the attacker can probe H_{TA} and H_{AR} from legitimate nodes, the injected component is now under the control of the attacker.

For instance, in [73], the authors describe a man-in-the-middle attack against link-based source identification protocols that employs relay attackers. Such protocols consist of two phases: training and identification. During the training phase, the receiver collects legitimate transmitter-receiver channels and saves them as signatures. In the identification phase, the source of a packet is determined by comparing its channel to the signatures collected during the training phase. To execute the man-in-the-middle attack, the relay attacker first injects its channels during the training phase. Later, to fabricate a packet with payload y, the attacker transmits $e^{-j2\pi f\Delta t} w H_{TA} y$. The receiver receives $e^{-j2\pi f\Delta t} w H_{TA} H_{AR} y$ and calculates the channel as $e^{-j2\pi f \Delta t} w H_{TA} H_{AR}$, which can be similar to the signature H given appropriate relay settings.

Similarly, in [60], the authors propose an attack against shared secret generation protocols, where two nodes in a communication system measure channels between them and independently generate secrets based on channel values. Under ideal conditions, the secrets generated on both sides should be the same due to channel reciprocity. The proposed attack involves the injection of the attacker's channels while the legitimate users are collecting channels for secret generation. The relay then estimates the secrets using the injected component $e^{-j2\pi f \Delta t} w H_{TA} H_{AR}$.

It is important to note that these attacks can be successful only when signals received from the attacker are comparable or stronger than the signal from the legitimate transmitter and the injected component dominates the channel. This can be achieved by placing the attacker node close to the receiver or using a high amplification power at the attacker node.

To avoid detection from source authentication systems or sudden changes in Received Signal Strength (RSS), authors of both works suggest that attackers can gradually increase their channel injection from a low amplification level.

3.2.2 Resolving Multipath Components from Channels

Signal paths are susceptible to various forms of attenuation and distortion when traveling to the receiver. For a single path with traveling distance d, attenuation parameter a, and phase distortion ϕ, the channel at frequency f can be described as:

$$h_f = a e^{-j2\pi \frac{df}{c} + j\phi} \tag{3.1}$$

In a multipath-rich environment, the received signal is a combination of multiple delayed and attenuated copies of the original signal that have traveled through different paths. Let d_i, a_i, ϕ_i be the traveling distance, attenuation parameter, and phase distortion of the i-th path, respectively, and N_p represent the total number of paths. The channel at frequency f can be described as:

$$h_f = \sum_{i}^{N_p} a_i e^{-j2\pi \frac{d_i f}{c} + j\phi_i} \tag{3.2}$$

To resolve the multipath components from channels, we can utilize observations at different frequencies, such as different subcarriers in Orthogonal Frequency-Division Multiplexing (OFDM) signals. The parameters of each signal path can be estimated through an optimization problem, where $H_{observed}$ represents the channel measurement, H represents the calculated channel, N_f is the number of subcarriers, and f_1-f_{N_f} are their frequencies.

$$\begin{array}{rl} \text{min} & \|H_{observed} - H\| \\ \text{s.t.} & H = [h_{f_1}\ h_{f_2}\ \dots\ h_{f_{N_f}}] \\ & \forall k \in [1, N_f], h_{f_k} = \sum_i^{N_p} a_i e^{-j2\pi \frac{d_i f_k}{c} + j\phi_i} \\ & \forall i \in [1, N_p], d_i > 0, a_i > 0, -\pi < \phi_i \leq \pi \end{array} \tag{3.3}$$

This concept has been utilized in several existing works for different purposes. For example, R2F2 [74] and OptML [18] focus on Long-Term Evolution (LTE) cross-band channel prediction. They resolve multipath components from channel observations at one band to estimate channels at a different band. mD-Track [86] adds the frequency shifts caused by the Doppler effect to the optimization problem and resolves multipath parameters to localize and track moving targets with Wi-Fi. To reduce the runtime of the optimization process, authors of aforementioned works have proposed various methods to find suitable initial values for the optimization parameters, particularly for the traveling distance which has a greater impact on the results than other parameters. In R2F2, the authors estimate the probability of the existence of signal paths and pick paths with high probability as initial values. In mD-Track, the authors define a similar probability estimation function and iteratively cancel the path with the highest probability until the remaining signal contains only noise. In OptML, the authors train a neural network to produce the probability distribution of path existence with the channel as input, then pick high-probability paths as initial values.

3.3 Related Work

3.3.1 Physical-Layer Source Authentication

Various physical-layer device identification and authentication methods have been proposed to improve wireless communication security. Link-based methods use wireless signal features, such as signal strength, Channel State Information (CSI), and Angle of Arrival (AoA) to identify and authenticate wireless devices. In [31], the authors propose to defend against Sybil attacks in sensor networks using RSS ratios from multiple receivers. Reference [58] proposes using CSI signatures over time to detect an attacker impersonating transmitters in static environments. Reference [87] proposed SecureArray, which utilizes a multi-antenna Access Point (AP) to profile the AoA of clients to identify each source.

Hardware-based methods extract features caused by unique hardware imperfections from received signals. The authors of [32] propose to distinguish among unique devices through timing analysis of 802.11 probe request frames. In [25], the authors propose PARADIS, where differentiating artifacts of individual wireless frames are measured in the modulation domain to identify devices. Reference [41] utilizes time-varying carrier frequency offset caused by oscillators for device authentication. The authors of [10] propose to use clock skew measurement as fingerprints of wireless devices. Recently, researchers have introduced machine learning techniques to help authentication [65, 7].

We believe that the existing methods cannot always be effective in defending against attacks using full-duplex relay devices. This is because both types of methods rely on previously collected signals, and some of them also require periodic updates to account for noise, environment changes, or errors in initial signatures. As a result, these methods can fail to detect channel injections if the attack has already begun before the defense system is employed, or if the attacker injects signals from a low amplification level and gradually increases it.

3.3.2 Relay Attacks and Countermeasures

Relay attacks enable an attacker to impersonate a participant in an authentication protocol by using one or more devices to relay authenticating messages between two parties. Such attacks have been proposed for various systems, such as near-field communications [34, 35, 64] and Bluetooth systems [40, 69, 20]. In most relay attack scenarios, the legitimate transmitter and receiver are located outside their intended communication ranges and these ranges are extended because of the attackers. As a result, relay attacks can be detected using distance-bounding-based methods, where authentication requests are rejected if the two parties are farther apart than expected [24, 33, 38, 39, 63].

In our targeted attacks utilizing full-duplex relays, the transmitter and receiver can directly communicate with each other. We believe that this difference from typical relay attacks makes them hard to detect for distance-bounding-based protocols. Specifically, since the transmitter and receiver are still within each other's communication range, they can pass the distance checks. Additionally, since the full-duplex relay does not inject any new messages, the exchanged information remains as expected and can pass cryptographic checks.

3.4 Insight

The design of RelayShield is based on the techniques of resolving multipath components from channel observations. By using prior knowledge of the transmit power and resolved multipath components, it is possible to differentiate between signal paths from the legitimate transmitter and those through an attacker, thus detecting the presence of active malicious relays and recovering the original channels.

Assuming the transmitter's transmit power is known at the receiver as prior knowledge and remains constant during the measurement period. This assumption is reasonable for typical wireless communication networks, such as home Wi-Fi networks, where the transmit power usually remains unchanged after the initial system setup. According to the free space propagation model, the received signal power P_r of a Line-of-Sight (LoS) path at a distance d from the transmitter would be

$$P_r = \frac{P_t \lambda^2}{(4\pi d)^2} \tag{3.4}$$

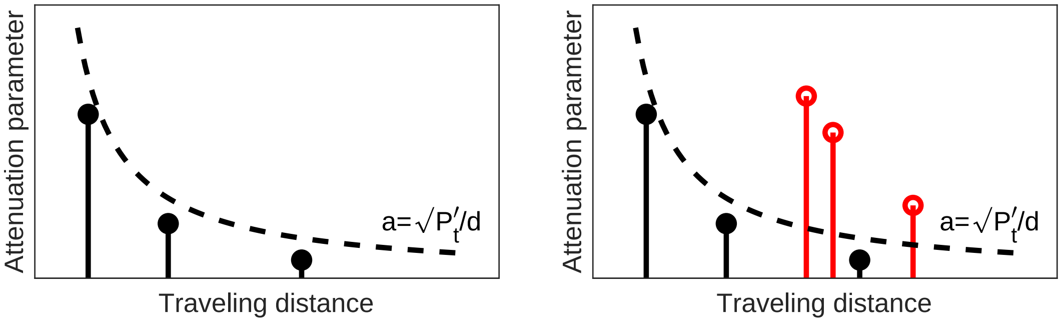
where P_t is the transmit power and λ is the signal wavelength. Compared with LoS paths, Non-Line-of-Sight (NLoS) paths from the same transmitter can experience greater attenuation due to reflections, scattering, and shadowing in the environment. As a result, signals through NLoS paths will be weaker than those through LoS paths of the same distance. Therefore, we can conclude that for N_p signal paths from the same source, we have:

$$\forall i \in [1, N_p], a_i \leq \frac{\sqrt{P_t'}}{d_i} \tag{3.5}$$

where a_i is the attenuation parameter of path i, which is equal to the square root of the received signal power. d_i is the traveling distance of path i, and we define $P_t' = \frac{P_t \lambda^2}{(4\pi)^2}$. For any channel, if we know the signal is from a single source and the transmit power of this source, we can include this constraint in the optimization problem in Equation (3.3) to improve the results of channel analysis.

The above constraint can also be used to detect malicious relays. For this purpose, it is necessary to understand the nature of the multipath components of paths through relays. Consider a signal path that passes through a relay attacker. Let a_{TA}, d_{TA}, and ϕ_{TA} denote the parameters of the transmitter-attacker section h_{TA}; let a_{AR}, d_{AR}, and ϕ_{AR} denote the parameters of the attacker-receiver section h_{AR}; and let w and Δt denote the amplification factor and delay of the relay. The channel at frequency f is:

$$\begin{aligned} h_f' &= e^{-j2\pi f \Delta t} w h_{TA} h_{AR} \\ &= e^{-j2\pi f \Delta t} w a_{TA} e^{-j2\pi \frac{d_{TA} f}{c} + j\phi_{TA}} a_{AR} e^{-j2\pi \frac{d_{AR} f}{c} + j\phi_{AR}} \\ &= w a_{TA} a_{AR} e^{-j2\pi \frac{(d_{TA} + d_{AR} + c\Delta t) f}{c} + j(\phi_{TA} + \phi_{AR})} \end{aligned} \tag{3.6}$$



(a) Without a relay

(b) With a relay

Figure 3.2: Examples of multipath components

The result obtained above can be compared with Equation (5.8). This comparison reveals that the signal path through the relay attacker will be resolved as a component with parameters $a' = wa_{TA}a_{AR}$, $d' = d_{TA} + d_{AR} + c\Delta t$, and $\phi' = \phi_{TA} + \phi_{AR}$. Considering that a relay attacker tends to use a large amplification factor w to ensure the success of attacks and the delay Δt can be a few sampling intervals in state-of-the-art full-duplex implementations [21], the resolved multipath component will exhibit an abnormally large attenuation parameter, a long traveling distance, and is highly likely to violate the constraint in Equation (3.5). Therefore, by checking if any of the resolved multipath components violate this constraint, it is possible to determine the existence of an active relay. The legitimate channels can then be recovered by excluding any suspicious components and reconstructing the channel using only the remaining components.

In Figure 3.2, we present resolved multipath components from two example channels. To detect the presence of a relay, we first resolve all paths from the channel, then compare their powers with the maximum possible received powers at their corresponding traveling distances. This comparison is made against the constraint in Equation (3.5), which is visualized as dashed lines in Figure 3.2. If all components comply with the constraint, as seen in Figure 3.2(a), it suggests that the channel is more likely from a single source, which in our case is the transmitter. Conversely, if one or more components violate the constraint, as seen with the red components in Figure 3.2(b), it indicates the presence of a relay attacker during the channel measurement.

It is possible for some signal paths through the relay to comply with the constraint as well. However, since these paths are resolved as having long traveling distances, their attenuation parameters would be extremely small. Neglecting these components would therefore have a minimal effect on the recovered results.

It is worth noting that our method is based on the assumption that we can perfectly resolve the multipath components of each signal path, which is not always achievable in real-world scenarios. For more reliable results and improved efficiency, we propose modifications to these steps in Section 3.5 and present their details.

3.5 System Design

The RelayShield system consists of two components: a relay detection module and a channel recovery module. As depicted in Figure 3.3, a channel measurement taken at the receiver is first passed through the relay detection module to determine if an active relay was present when the measurement was taken. If no relay is detected, the channel is considered safe for use. Otherwise, the measurement is further processed by the channel recovery module, which resolves the multipath components of the signal and reconstructs the legitimate channel by selecting the appropriate components.

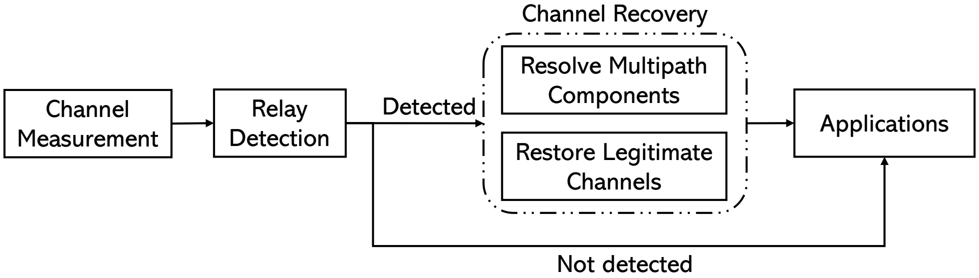


Figure 3.3: An overview of RelayShield

In this section, we will describe the design and implementation of both the relay detection and channel recovery modules. Additionally, we will discuss the generation of simulated training datasets for the neural networks used in these modules.

3.5.1 Relay Detection

In Section 3.4, we have discussed the expected signal path parameters for channels from a single source. We observe that while optimizing Equation (3.3) with the additional constraint in Equation (3.5) yields good fits for channels from the transmitter only, it usually fails for channels collected with an active relay. An obvious relay detection solution involves solving the optimization problem for every measurement and checking whether a good fit is achieved. However, this approach is computationally expensive and impractical for real-time detection.

We tackle this issue by training a neural network on simulated channels. We find that such a neural network can provide comparably accurate detection results with much less time than solving the optimization problem. The training dataset consisted of two cases: legitimate channels from transmitters only, and channels measured from mixed signals from transmitters and through relays. Each channel of n_f subcarriers can be represented as n_f complex values. To ensure compatibility with the neural network libraries used for implementation, we separated the real and imaginary parts of each value and represented each channel as an array of $2n_f$ real numbers. The output of the neural network is a binary value representing whether an active relay is present. To remove the effect of transmit power, input channels are normalized by power before feeding into the neural network.

While training a neural network model with a large number of channels is time-consuming, once trained, it can output results quickly. Furthermore, since only four environment-specific parameters - the minimum and maximum of total path numbers, and the minimum and maximum of possible traveling distances - are required to generate simulated channels in the training datasets, it's possible to apply a model trained for one environment to another if they share similar features, such as room dimensions and reflectors.

3.5.2 Legitimate Channel Recovery

The channel recovery module is to address the scenario where a malicious relay is detected but the channel must still be used for security or other purposes. It first resolves the multipath components and their sources, and then uses components from the transmitter to restore the legitimate channel.

We resolve the multipath components in two steps. First, the channel is fed to a neural network to generate initial estimates of the traveling distances. Next, we apply the optimization to refine the initial estimates and determine the other parameters. We observe that the initial values of traveling distances have a more pronounced impact on the results than other parameter types, owing to the larger search ranges of traveling distances and the non-convex nature of the problem. As a neural network cannot provide adequate initial estimates for all parameters, we train the neural network to only provide estimates for traveling distances and the sources of corresponding signal paths (i.e., whether they are from the transmitter or through the relay attacker).

The neural network is trained using simulated input channels following the same format as described in Section 3.5.1. The output is defined as an array of size $2N_{p,max}$, where $N_{p,max}$ is the maximum number of multipath components considered from a source. The first $N_{p,max}$ values in the output array represent the traveling distances of components from the legitimate transmitter, and the second $N_{p,max}$ values represent the traveling distances of components through the relay. When there are less than $N_{p,max}$ signal paths from one source, we use a placeholder value d_{null} to denote the absence of a path. For example, if $N_{p,max} = 3$, the output $[d_{TR,1}, d_{TR,2}, d_{null}, d_{A,1}, d_{A,2}, d_{A,3}]$ indicates that there are two multipath components

($d_{TR,1}$-$d_{TR,2}$) from the transmitter and three multipath components ($d_{A,1}$-$d_{A,3}$) through the relay attacker. To avoid confusion, d_{null} is set to a value greater than the maximum considered traveling distance in the implementation. During processing, we interpret out-of-range values as nonexistent paths and discard them before the optimization. Simulated channels in training datasets and input channel measurements are normalized before feeding into the model.

$$
\begin{array}{rl}
\min & \|H_{observed} - H_{TR} - H_A\| - \alpha \sum_m^M a_{A,m} \\
\text{s.t.} & H_{TR} = [h_{TR,f_1}\ h_{TR,f_2}\ \ldots\ h_{TR,f_{N_f}}] \\
& H_A = [h_{A,f_1}\ h_{A,f_2}\ \ldots\ h_{A,f_{N_f}}] \\
& \forall k \in [1, N_f], h_{TR,f_k} = \sum_n^N a_{TR,n} e^{-j2\pi \frac{d_{TR,n} f_k}{c} + j\phi_{TR,n}} \\
& \forall k \in [1, N_f], h_{A,f_k} = \sum_m^M a_{A,m} e^{-j2\pi \frac{d_{A,m} f_k}{c} + j\phi_{A,m}} \\
& \forall m \in [1, M], a_{A,m} > 0, -\pi < \phi_{A,m} < \pi \\
& \forall m \in [1, M], \|d_{A,m} - d_{A,m,init}\| < r_d \\
& \forall n \in [1, N], a_{TR,n} > 0, -\pi < \phi_{TR,n} < \pi \\
& \forall n \in [1, N], \|d_{TR,n} - d_{TR,n,init}\| < r_d \\
& \forall n \in [1, N], a_{TR,n} \leq \frac{\sqrt{P_t'}}{d_{TR,n}}
\end{array}
\tag{3.7}
$$

We formulate the optimization problem as shown in Equation (3.7). The input observed channel is denoted as $H_{observed}$, and our objective is to identify the multipath components that provide the best fit, which is a combination of the transmitter-receiver channel H_{TR} and the transmitter-relay-receiver channel H_A. All channels contain data for N_f subcarriers. h_{TR,f_k} and h_{A,f_k} represent the transmitter-receiver or transmitter-relay-receiver channel of the subcarrier at frequency f_k. M and N are the numbers of multipath components in the transmitter-relay-receiver and transmitter-receiver channels, respectively. The attenuation parameter, traveling distance, and phase shift of the m-th transmitter-relay-receiver signal path are denoted as $a_{A,m}$, $d_{A,m}$, and $\phi_{A,m}$, and those of the n-th transmitter-receiver signal path are denoted as $a_{TR,n}$, $d_{TR,n}$, and $\phi_{TR,n}$. We restrict the search range for traveling distances, denoted as r_d, around the initial guesses ($d_{A,m,init}$ or $d_{TR,n,init}$), while searching for all possible values of other parameters. P_t' denotes the transmit power factor as described in Section 3.4.

In this optimization, we notice a specific type of overfitting that can result in multipath components with extremely large attenuation parameters for channels through the relay. In such instances, the signals through each path can be more than 10 dB stronger than the total received power of observed signals, and they cancel each other out in the result channels. To prevent such outcomes, we add a penalty term $\alpha \sum_m^M a_{A,m}$ to the objective function. Since the objective function is non-convex, this optimization problem is susceptible to converging at local minima. Therefore, we employ the basin-hopping optimization algorithm to obtain better results.

With multipath components resolved, we can recover the legitimate transmitter-receiver channel as:

$$\begin{aligned} H_{TR} &= [h_{TR,f_1}\ h_{TR,f_2}\ \dots\ h_{TR,f_{N_f}}] \\ h_{TR,f_k} &= \sum_n^N a_{TR,n} e^{-j2\pi \frac{d_{TR,n} f_k}{c} + j\phi_{TR,n}} \end{aligned} \tag{3.8}$$

3.5.3 Training Dataset Generation

The relay detection and channel recovery modules both contain neural networks that require channels as training inputs. We used simulated channels to train these models because the ground truth traveling distances of signal paths are required to train the neural network in the recovery module, but they are hard to obtain due to limited sampling rates of most commercial hardware.

We use Algorithm 1 to simulate channels from transmitters. First, we randomly generate signal path parameters to simulate the channels in different environments. The attenuation parameters are then adjusted according to the traveling distances to make the generated signal path parameters satisfy the constraint in Equation (3.5), assuming a uniform transmit power of $P = 1$. Finally, we add up each path's channel response to get the channel. If the attenuation parameter of a signal path is smaller than a threshold, we exclude this path because it brings little change to the channel and might confuse the neural network.

Algorithm 1: Generate channels from transmitters

```
Input: $\vec{f}$: frequencies of subcarriers
       $N_{min}$, $N_{max}$: the minimum and maximum number of signal paths
       $d_{min}$, $d_{max}$: the minimum and maximum traveling distance of signal paths
       $a_{min}$: the minimum considered attenuation parameter of signal paths
Output: A channel $\vec{h}$ that satisfies Equation (3.5)
$N_f$ = the size of $\vec{f}$
$\vec{h}$ = a zero array of size $N_f$
$N_p$ = a random number between $N_{min}$ and $N_{max}$
$\vec{d} = N_p$ random numbers between $d_{min}$ and $d_{max}$
$\vec{\phi} = N_p$ random numbers between $-\pi$ and $\pi$
$\vec{a} = N_p$ random numbers between 0 and 1 element-wisely divided by $\vec{d}$
foreach integer $i$ between 1 and $N_f$ do
    foreach integer $j$ between 1 and $N_p$ do
        if $a_j > a_{min}$ then
            $h_i = h_i + a_j e^{-j2\pi \frac{d_j f_i}{c} + j\phi_j}$
Add Gaussian white noise to $\vec{h}$
```

Algorithm 2: Generate channels manipulated by relays

```
Input: $\vec{f}$: frequencies of subcarriers
       $\vec{h_{tr}}$: simulated transmitter-receiver channel generated using Algorithm 1
       $\vec{h_a}$: simulated transmitter-relay-receiver channel
       $\Delta t_{min}$, $\Delta t_{max}$: the minimum and maximum possible delay of the relay
       $r_{min}$, $r_{max}$: the minimum and maximum considered received signal power
       ratio through the relay attacker vs. from the transmitter
Output: A simulated channel $\vec{h}$ manipulated by the relay attacker
$\Delta t$ = a random number between $\Delta t_{min}$ and $\Delta t_{max}$
$\vec{h_a} = e^{-j2\pi\Delta t}\vec{h_a}$
$r$ = a random number between $r_{min}$ and $r_{max}$
$\vec{h} = \vec{h_{tr}} + \frac{\|\vec{h_{tr}}\|}{\|\vec{h_a}\|} r \vec{h_a}$
```

Channels through relays are generated in a similar manner, with the exception that the attenuation parameters are not divided by traveling distances. After obtaining channels from transmitters and through relays, their power levels are tuned and a random delay is added to the channels through relays. These channels are then added up with channels from transmitters, as described in Algorithm 2, to simulate measurements during attacks.

3.6 Implementation

We implement the relay detection and channel recovery modules in Python. We define the neural networks in both modules as fully-connected neural networks with exponential linear unit activation function with Keras. The neural networks have 10 hidden layers and 100 neurons in each layer. The optimization problem in the channel recovery module is solved using the basin-hopping method in SciPy. We limit the search ranges of traveling paths to 5 m around the initial guesses and set the penalty rate $\alpha = 0.1$.

We train the neural networks with simulated channels generated as in Section 3.5.3. In the detection module, the neural network is trained with 200,000 channels, half of them are from transmitters only, and the other half are mixes of channels from transmitters and through relays. In the recovery module, the neural network is trained with 200,000 channels, which are mixes of channels from transmitters and through relays. According to observations of the experiment environments, one channel is generated to have 2-4 signal paths, each with a traveling distance of 1-120 m. We assume delays at the relays are between 1 and 5 sampling intervals and discard signal paths with attenuation parameters less than 0.1. For mixed channels, we consider the received signal power ratios through the relay vs. from the transmitter from -3 dB to 6 dB.

3.7 Evaluation

In this section, we will first introduce the channel collection process, then present the experiment designs and results.

3.7.1 Data Collection

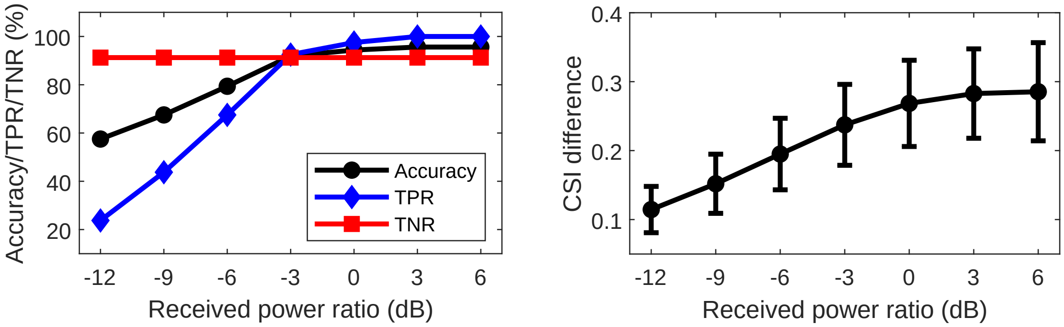
We use WARP v3 software-defined radios to collect channels in two typical indoor environments. Because of a μs-level latency from the receiving RF chain to the transmitter RF chain, we could not use our devices to implement an attacker with a delay time of a few sampling intervals as the state-of-art full-duplex relay implementation [21]. Instead, we emulate channels with active relay attackers as follows.

1. The transmitter transmits packet x, the relay and receiver receive y_a and $y_{r,1}$.
2. The relay transmits y_a received in step 1 with amplification, and the receiver receives $y_{r,2}$.
3. We calculate the transmitter-receiver channel from $y_{r,1}$ and the transmitter-relay-receiver channel from $y_{r,2}$, then combine them to emulate the injected channels.

The first two steps are completed within the coherence time for each run of the experiments. The last step is done offline, where we can adjust the received power and delay time of $y_{r,2}$ to emulate different relay settings. We use the transmitter-receiver channels measured from the $y_{r,1}$ packets as the ground truth for evaluations of the channel recovery module.

We generate the packets following the 802.11n standard and use band 11 at 2.4 GHz for experiments. The bandwidth is 20 MHz. Each channel measurement contains values of 52 subcarriers. The WARP nodes are calibrated to avoid random phase offsets and synchronized with CM-PLL modules.

We collect 30 sets of channels from a typical home environment and 50 sets of channels from a typical office environment over three weeks. Around one-third of data in each environment are collected in NLoS settings, where the obstacles include cubicle panels, chairs, books, and walls. We also ask volunteers to stand or sit down at multiple locations in the LoS



(a) Relay detection module

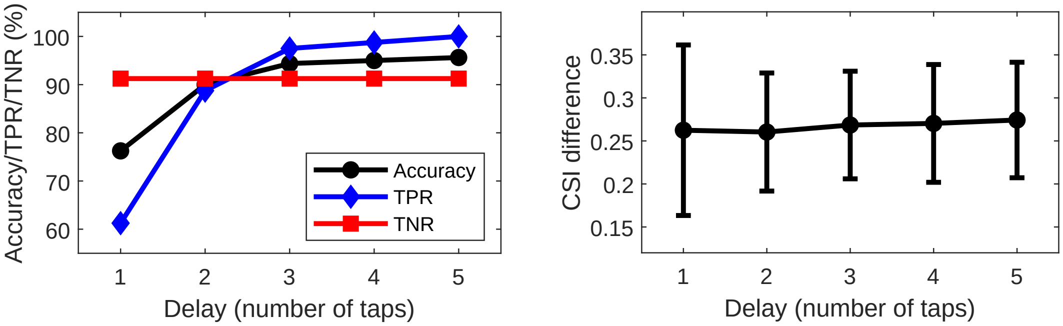
(b) CSI difference

Figure 3.4: Detection accuracies, TPR, TNR, and CSI difference results with different received power ratios from relays vs. transmitters and a 3-tap relay delay

between transmitting and receiving antennas. Each set of channels contains a transmitter-receiver channel and a transmitter-relay-receiver channel. In the home environment, they are collected from the living room and kitchen, which together form a 3.5 m × 7.5 m area. The office environment is a 12 m × 18 m room. It is an open-plan office with furniture for around 15 people. We change the locations of nodes before collecting every set of channels. The transmitter and receiver are placed 2-15 m apart. The relay is 1-10 m away from the transmitter and receiver. They are used in the following evaluations unless otherwise specified.

3.7.2 Relay Detection

Several prior works have employed CSI signatures for packet source identification [58, 45]. The CSI of a packet is compared against previously-collected CSI signatures, and the packet is considered legitimate if the CSI difference is within a certain threshold. To compare RelayShield with this general source identification approach, we calculate the average CSI difference per subcarrier between the mixed and transmitter-receiver channels. We evaluate the effectiveness of RelayShield against various relay configurations, using received power



(a) Relay detection module

(b) CSI difference

Figure 3.5: Detection accuracies, TPR, TNR, and CSI difference results with different delay times at relays and a 0 dB received power ratio from relays vs. transmitters

ratios of transmitter-relay-receiver and transmitter-receiver signals to quantify the amplification settings of relays. We refer to one sampling interval as one *tap* and consider delays of 1-5 taps at relays, within the delay range of state-of-the-art full-duplex implementations [21]. Figures 3.4 and 3.5 depict the evaluation results.

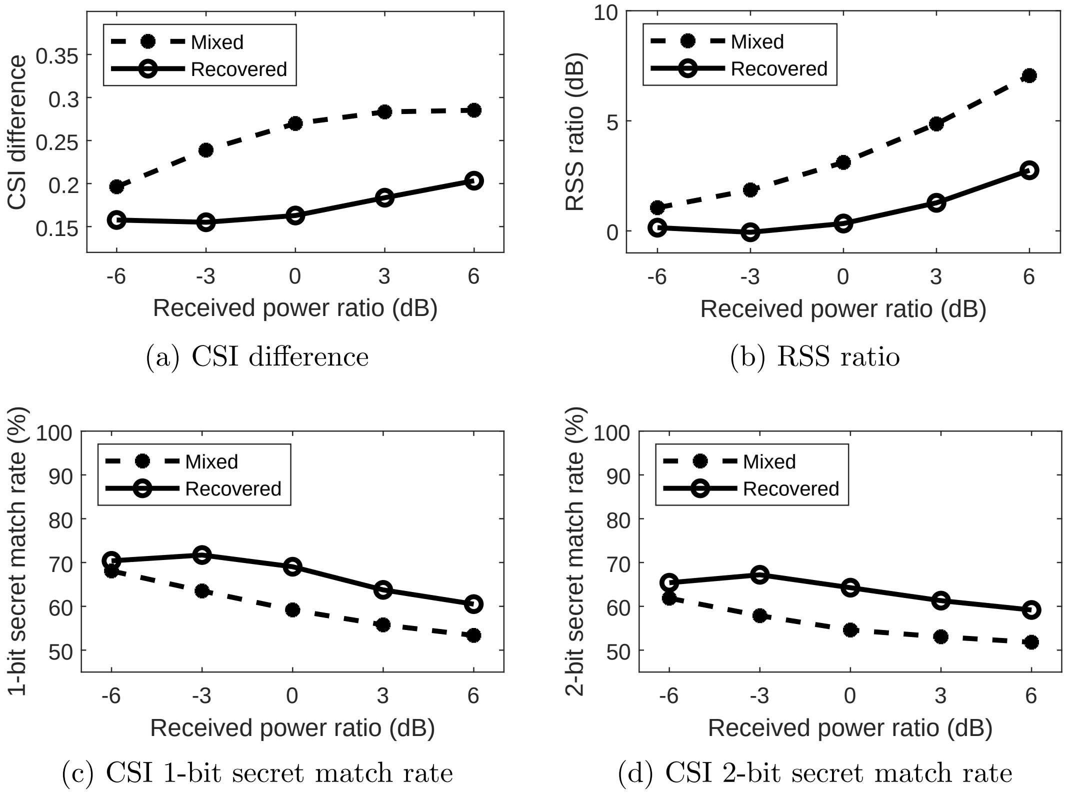
An increase in the amplification gain of a malicious relay leads to the signals through it taking a larger portion of the received signals, making the relay attacker more detectable to RelayShield, as shown in Figure 3.4(a). The detection accuracy of the system increases from 57.5% to 95.6% as the received power ratio increases from -12 dB to 6 dB. Furthermore, Figure 3.4(a) demonstrates that an increase in relay delay time also results in higher detection accuracy for the detection module. We observe that the channel-to-signal-path methods can resolve multipath components more accurately when the signal paths have distant parameters. Although the detection module does not resolve all multipath components, the neural network can still benefit from the distant parameters caused by the long delay time.

Increasing the amplification gain at the malicious relay results in greater differences between the channel measurements and the legitimate ones, as shown in Figure 3.4(b). Considering that most CSI difference values of two consecutive packets are below 0.13 in our test environments, it seems that with a proper threshold, using CSI difference to detect malicious relays outperforms RelayShield's relay detection module. However, this method assumes the existence of ground truth transmitter-receiver channels as signatures, which does not hold if the malicious relay begins with low amplification power to evade signature-based source authentication systems. In contrast, RelayShield's detection module can detect such attacks since it produces results independently of any signatures. In cases of gradual injection, although RelayShield's detection accuracy may initially be lower, we can eventually detect the attackers with high accuracy since they need to increase the amplification power to a certain level to succeed. In targeted attacks [73, 60], the RSS from the attackers must be equal to or greater than that from the legitimate transmitter for acceptable success rates.

We notice that the neural network's True Negative Rate (TNR) and True Positive Rate (TPR) are related to the received signal power ratio in the training dataset. When the training dataset includes mixed channels with low power from relays, the neural network may confuse them with channels from transmitters only, resulting in a decrease in TNR across all relay settings and a significant increase in TPR for cases with low received power ratios. Since frequent false alarms are more undesirable than occasionally missed detections in our targeted scenarios, we exclude mixed channels with low power from relays and train the model used in the above evaluations with received power ratios of 0-6 dB.

3.7.3 Channel Recovery

We evaluate the channel recovery module in two ways. First, we compare recovered channels and ground truth using metrics that measure errors in CSI and RSS. Second, we use the recovered channels as inputs for two typical applications and see if key features are preserved. The metrics and applications include:



(a) CSI difference

(b) RSS ratio

(c) CSI 1-bit secret match rate

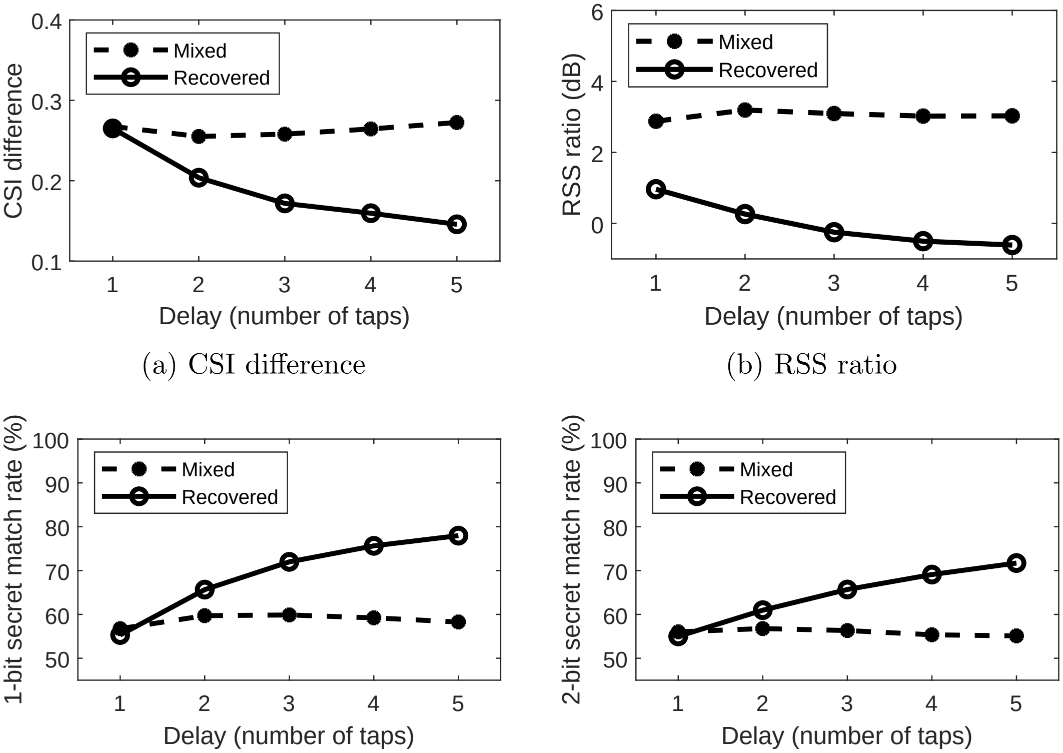
(d) CSI 2-bit secret match rate

Figure 3.6: Recovery results with different received power ratios from relays vs. transmitters and a 3-tap relay delay

- Average difference of normalized CSI per subcarrier: it describes the dissimilarity of CSIs in their shapes. We calculate it between the mixed/recovered channels and ground truth channels.

- RSS ratio: we calculate the ratios of mixed/recovered channel power and ground truth channel power. RSS ratios between recovered and ground truth channels closer to 0 dB indicate better recovery.

- Secret match rate of the CSI n-bit quantizer [47]: CSI n-bit quantizer is an example shared secret generation protocol. Smoothed CSI values are quantized into 2^n levels determined by the distribution and then converted to binary secrets. We consider $n = 1$ and 2.



(a) CSI difference

(b) RSS ratio

(c) CSI 1-bit secret match rate

(d) CSI 2-bit secret match rate

Figure 3.7: Recovery results with different delay times at relays and a 0 dB received power ratio from relays vs. transmitters

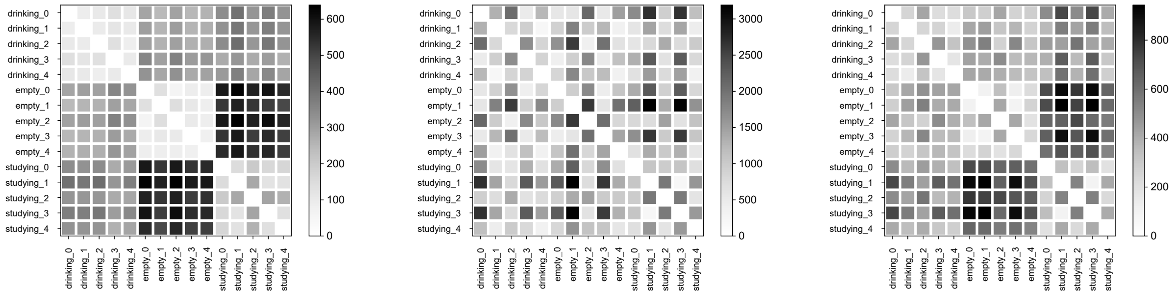
- E-eyes [80]: an example activity classification system. E-eyes first leverages the cumulative CSI moving variance to differentiate walking and in-place activities. It further classifies in-place activities by comparing an unknown trace's CSI distribution over time with profiles using the Earth Mover's Distance (EMD).

Channel Metrics and Recovered Channels for Shared Secret Generation

Recovery results with different relay settings are shown in Figures 3.6 and 3.7. The *mixed* channels are the injected channels before recovery, which are mixes of the legitimate channels and channels through the relay.

As shown in Figure 3.6(a), the CSI differences of recovered channels increase slightly with the received power ratio. That's because one signal path can affect the parameter estimation of other paths when we resolve them from channels. When paths through the relays have larger attenuation parameters, they bring more interference to the signal path parameter estimation of legitimate channels and cause more error in recovered results. As in Figure 3.7(a), the CSI differences of recovered channels decrease with delay time. This is because paths with distant parameters, especially traveling distances, are more likely to be resolved accurately. The RSS recovery results in Figures 3.6(b) and 3.7(b) downgrade with increased relay amplification and decreased delays for the same reasons. The recovery module can bring a decrease of CSI difference up to 0.127 and have recovered signal strength errors within 1 dB under most considered settings.

Figures 3.6(c)-3.6(d) and 3.7(c)-3.7(d) show the match rates of secrets generated from mixed/recovered channels and secrets generated from the ground truth channels. For both cases, the match rates of recovered channels decrease with the received power ratio and increase with the delay time. The match rates of CSI 2-bit secrets are lower than the corresponding 1-bit secret match rates because of their greater sensitivity to CSI fluctuations caused by the twofold quantization levels. Under all considered settings, the recovery module can bring an increase of up to 19.7% to the secret match rates.



(a) Ground truth

(b) Mixed traces

(c) Recovered traces

Figure 3.8: EMD results of indoor trace pairs

Recovered Channels for Activity Classification

E-eyes is a location-oriented activity classification system utilizing continuously collected channels from multiple devices around a home. After preprocessing the Wi-Fi channel measurements, E-eyes first runs a coarse activity determination to differentiate walking activities and in-place activities using moving variance. For in-place activities, it will further classify them by comparing their CSI distributions over time with the profiles of previously-collected activity profiles. In this evaluation, we assume a malicious full-duplex device attacks the E-eyes system by introducing additional fluctuations to the CSI measurements, and check if the channel recovery module in RelayShield can reduce the fluctuations and output results that are accurate enough to support sensing applications.

A full-duplex attacker can insert additional fluctuations to the CSI measurements by randomly changing its amplification factor. The extra fluctuations can affect E-eyes' sensing results in two ways: 1) In the coarse activity determination, in-place activities with less CSI variation can be taken as walking activities with greater variation; 2) In the in-place activity identification, the CSI distribution changes caused by the fluctuations can lead to misclassifications among the in-place activities.

For this evaluation, we collect 20 10-second traces of four activities: empty room (no human movements), walking, drinking water (sitting down with arm movements), and studying (sitting down, typing or writing) in the office environment at 15 packets/sec. The transmitter and receiver are placed 2 m away from each other. The volunteers repeat the activities 1-3 m away from the transmitter and receiver.

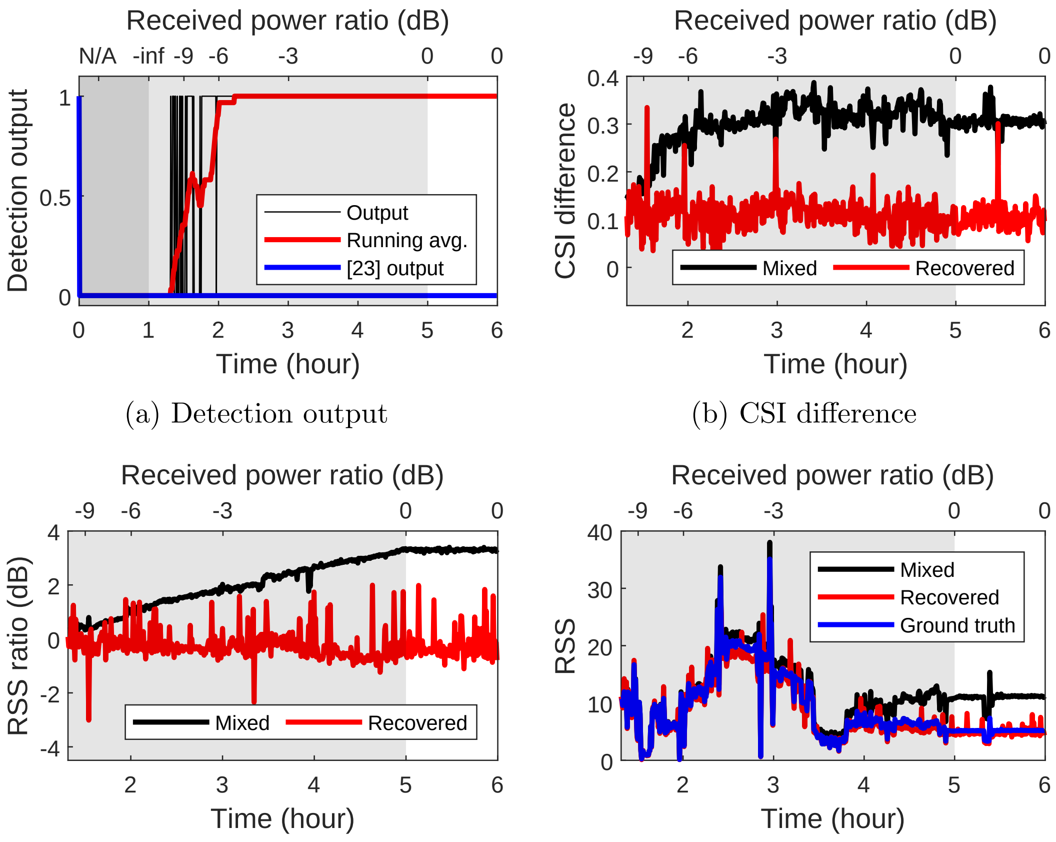
First, we focus on the coarse activity determination in E-eyes and choose traces collected in an empty room as in-place activity examples. The attacker will make E-eyes confuse these empty room traces with walking traces. To simulate fluctuations of walking, we change the received power ratio by a random value between -1 dB and 1 dB every packet. As in E-eyes, we normalize the maximum cumulative CSI moving variance by each trace's average power and present the results in Table 3.1. The mixed traces after the attack have a greater moving variance than the ground truth traces before the attack. They are very likely to be taken as walking traces considering that their variance value is much closer to the walking traces than the ground truth empty room traces. After processing by the channel recovery module of RelayShield, the moving variance of the traces drops from 0.0285 to 0.0127. The moving variance of the recovered channels is still greater than that of the ground truth, which indicates that not all the fluctuations introduced by the attacker are canceled. But it is brought back to a level close to the ground truth empty room traces and is more likely to be classified as an in-place activity.

Table 3.1: Average normalized cumulative moving variance of different types of traces

Trace type	Normalized cumulative moving variance
Walking	0.0329
Empty room - mixed	0.0285
Empty room - ground truth	0.0104
Empty room - recovered	0.0127

We further consider the case where an attacker targets the in-place activity identification in E-eyes and affects the CSI distributions over time by inserting small fluctuations. In this case, the attacker changes the received power ratio by a random value between -0.5 and 0.5 dB every packet to simulate the small fluctuations caused by different in-place activities. Figure 3.8 shows the EMD results of the ground truth traces before the attack, the mixed traces after the attack, and the traces after the channel recovery. EMD calculates the minimal cost to transform one distribution into the other. Smaller EMDs indicate more similar distributions. It can be seen that the EMDs of ground truth channels have a clear pattern: traces of the same activity type have significantly smaller EMDs with each other than with traces of other activity types. After the attack, this pattern does not hold for mixed traces anymore. The EMDs of mixed traces are not related to activity types because of the randomness introduced by the fluctuations. After channel recovery, the EMDs of recovered traces are still different from the ground truth traces, but they show a similar pattern to the ground truth, where trace pairs of the same activity have much smaller distances than others.

We pick one trace of each activity as the profile and use them to classify the remaining traces. The classification accuracy of the ground truth traces is 100%. It drops to 33.3% for mixed traces and is back to 100% for the recovered traces. Our in-place activity classification accuracy for recovered traces is higher than reported in the original E-eyes paper for two reasons. First, we consider a simplified scenario of the in-place activity classification than in evaluations of E-eyes, in which is all activity traces are collected at the same location. E-eyes is a location-oriented system. It infers location information by checking traces from multiple devices in the environment, and then narrowing down the range of possible activities with the location information. Second, the number of traces we collected for each type of activity is smaller than in the evaluations of E-eyes. The experiment results we have are more to show that the recovered channels can still be used as inputs for sensing systems, rather than evaluate the performance of E-eyes itself.



(a) Detection output

(b) CSI difference

(c) RSS ratio

(d) RSS value

Figure 3.9: Detection and recovery results of the system test

Table 3.2: Relay detection and channel recovery results of channel-to-signal-path methods

Input data		Detection accuracy	Avg. CSI difference	Recovered RSS	CSI 1-bit secret match rate	CSI 2-bit secret match rate
Mixed channels		-	0.2813	3.0188 dB	57.69%	53.50%
Channels recovered with	R2F2 [74]	36.25%	0.2971	-2.3567 dB	49.81%	49.92%
	OptML [18]	54.38%	0.2902	4.1404 dB	52.76%	51.63%
	mD-Track [86]	26.25%	0.2833	-6.2592 dB	49.80%	51.13%
	RelayShield	95.63%	0.1457	-0.6090 dB	77.97%	71.72%

3.7.4 System Test

To evaluate our system's performance during real-world channel injections, we collect channels in the home environment continuously for 6 hours on a weekend day, when the volunteers living in that household are highly active. There is no obstacle between the nodes when we deployed the devices, but the volunteers occasionally blocked the LoS paths between each pair of nodes due to their daily activities. We emulate the full injection process in three phases: no injection (the first hour), gradual injection (the second to the fifth hour), and stable injection (the last hour). During the gradual injection, the attacker slowly increases its amplification power from zero until the received powers from the transmitter and through the relay are equivalent. During the stable injection, the attacker sticks to this amplification setting. We assume a 3-tap delay for the attacker to process and send the signal.

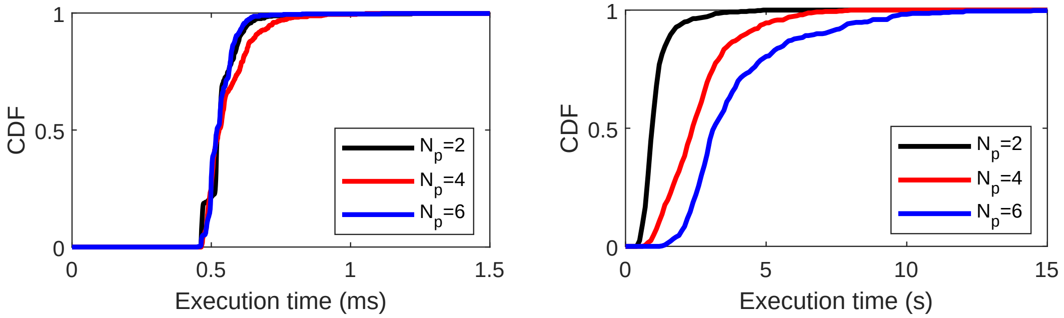
The detection results of RelayShield and [58] are shown in Figure 3.9(a). The dark grey, light grey, and white sections represent the phases of no injection, gradual injection, and stable injection, respectively. We tune parameters for [58] so that it achieves over 95% accuracy with ground truth traces without a relay attacker. Our detection module first reports an active relay when the received power from the relay is 10.87 dB lower than the transmitter. The results switch between detected and not detected for a while because of the low amplification at the relay, but a larger percentage of them turn to detected over time, as can be inferred from the running average within a 15-minute window. The results stay at detected since when the received power from the relay is 6.06 dB lower than the transmitter. While [58] does not report any detection since the injection begins. The recovery results are shown in Figures 3.9(b)-3.9(d). They are plotted with traces after the detection module first reports a relay. The recovery module brings the CSI difference compared with ground truth channels to around 0.1, and the recovered RSS values are also very close to the ground truth except for a few outliers.

3.7.5 Comparison with Existing Channel-to-Signal-Path Methods

Since the channel-to-signal-path idea has been adopted in multiple existing works, one might wonder if it is possible to apply one of those methods directly to the channel and check the resolved multipath components as mentioned in section 3.4. To answer this question, we resolve multipath components with the channel-to-signal-path methods in R2F2 [74], OptML [18], and mD-Track [86], and see if we can use them for relay detection and channel recovery. Since channel-to-signal-path techniques in these works are not designed to defend against relay attackers, we define the following mechanisms to detect relays and recover channels from multipath components:

- Relay detection: if all multipath components satisfy the constraint in Equation (3.5), we say that this channel is not affected by a malicious relay. Otherwise, we say that a relay attacker is found.
- Channel recovery: we exclude components violating the constraint in Equation (3.5) and use the remaining ones to reconstruct the legitimate channel.

It is possible that some components through the relay also satisfy the constraints when compared with the LoS path, especially when the relay has a short delay time and the signal paths through it are resolved as components with small traveling distances. To reduce the chance of this case and make the comparison as fair as possible, we use mixed channels with 5-tap delays as inputs. The comparison results are shown in Table 3.2. OptML achieves higher accuracy in relay detection, but is still close to random guessing results. The recovery metrics of all comparing systems are close to the values of the mixed channels, which means that little recovery is achieved. We have noticed that the considered systems do not always produce multipath components that make sense in our experiment environment, which explains the



(a) Relay detection runtime (b) Channel recovery runtime

Figure 3.10: Runtime distributions of the relay detection and channel recovery modules of 500 input channels

performances. Although all tested methods have been proven effective in their targeted scenarios, we can not expect the results to be accurate in all environments for all purposes.

3.7.6 Runtime

We run RelayShield on a laptop with the Intel Core i7-8550U processor and 16 GB memory with simulated channels that have N_p signal paths in the transmitter-receiver channels and transmitter-relay-receiver channels. We use simulated channels to have better control of the number of signal paths. Figure 3.10 shows that the execution time of the detection module does not change much with the number of signal paths increasing. This is because the runtime of neural networks is mostly decided by their structures, not the input values. In all tests, the relay detection module can produce results within 1 ms, which makes it practical for real-time processing. The execution time of the recovery module increases with the number of signal paths because of more variables in the optimization problem. The average runtimes of $N_p = 2, 4, 6$ are 1.08 s, 2.58 s, and 3.88 s, respectively. Although the recovery module takes too long for wireless nodes to process channels locally in real time, it can be implemented with the link-based applications as a prior step in devices with more computing power and get activated only when necessary.

3.8 Discussions

3.8.1 Simulated Channels as Training Datasets

RelayShield takes simulated channels to train the neural network models used in relay detection and channel recovery modules, and is tested with real channels collected in two indoor environments as discussed in Section 3.7. Since neural networks assume that training and testing data are independent and identically distributed, using real-world channels to train the models is supposed to improve the system performance. However, using real-world data also has some potential drawbacks. First, collecting a large enough dataset of real-world channels is expensive and requires considerable effort. Considering that the two neural network models take the relay attacker existence and multipath parameters as labels, collecting real-world training data requires the user to build an attack prototype and use devices with a GHz-level sampling rate to separate multipaths. Second, multipath in channels is affected by device locations and dynamic channel conditions. If the real-world training dataset does not contain data under all possible conditions in an environment, the trained model could be sensitive to any changes in the runtime. Compared with collecting real-world channels as training datasets, generating simulated channels takes significantly less effort. The random parameters used in simulated channel generation also make the datasets cover all possible cases in environments with similar features. Therefore, despite the potential performance improvement of using real-world channels, we believe that using simulated channels as training datasets is a viable approach.

3.8.2 Optimizing System Parameters for Different Environments

To ensure accurate channel recovery results, it is crucial to select appropriate values for system parameters such as signal path numbers and traveling distances based on the environment. A quick way to determine suitable values for these parameters is to collect several channels from different locations in an environment, resolve their parameters with the problem defined in Equation (3.7), and observe the resulting ranges. We found that if the ranges of traveling distances and path counts are set to be smaller than needed, it is difficult for any channel-to-signal-path method to find good fits to input channels. Conversely, if the ranges are larger than needed, it will negatively impact the results, but not as much as with smaller ranges. Therefore, we recommend starting with larger ranges and gradually decreasing them until the appropriate values are determined.

3.8.3 Performance of Channel-to-Signal-Path Methods

When evaluating our channel recovery module and comparing RelayShield to other systems that use channel-to-signal-path methods discussed in Section 3.7.5, we test these methods using simulated indoor channels and compare the results to ground truth channel parameters. We find that all tested methods resolve signal path parameters less accurately when the signal paths have close-by parameters, especially close-by traveling distances. The performances of all systems can also decrease with a large number of signal paths and high noise levels. Additionally, we observe that it is possible for two different sets of signal paths to form channels with a negligible difference. We believe that while channel-to-signal-path methods are useful tools for many applications, they cannot resolve multipath components perfectly in all cases. However, it may be possible to improve their performance by limiting the search range based on the specific environment or application scenario.

3.8.4 RelayShield Limitations

Based on existing attack works [73, 60], we assume that the malicious full-duplex relay uses the same complex amplification factor w for all subcarriers. It is based on this assumption that we believe the relayed signals can be interpreted as some abnormal multipath components, as explained in Section 3.4. However, if the attacker set various amplification factors for different subcarriers, RelayShield might fail to detect attackers and recover the channels.

Chapter 4: Eavesdropping MU-MIMO Systems with Malicious Full-Duplex Relays

As the demand for high-speed and reliable wireless networks increases, MU-MIMO technology has become a popular choice for wireless communication systems. However, this technology also brings new security challenges, one of which is the vulnerability during the channel sounding process. In this paper, we propose an active eavesdropping attack targeting MU-MIMO systems. The attack consists of two phases. First, the attacker sends a forged pilot packet to the victims. After that, the AP transmits streams intended for victims to the attacker, who operates in full-duplex mode and relays the streams to the victims. Compared to existing eavesdropping attacks targeting MU-MIMO systems, our proposed attack requires less prior knowledge and coordination from attackers and maximizes eavesdropping opportunities. We evaluate the proposed attack in various settings and prove its effectiveness with multiple victims and partial channel knowledge. Additionally, we explore the use of physical-layer features to detect our proposed attack.

4.1 Motivation

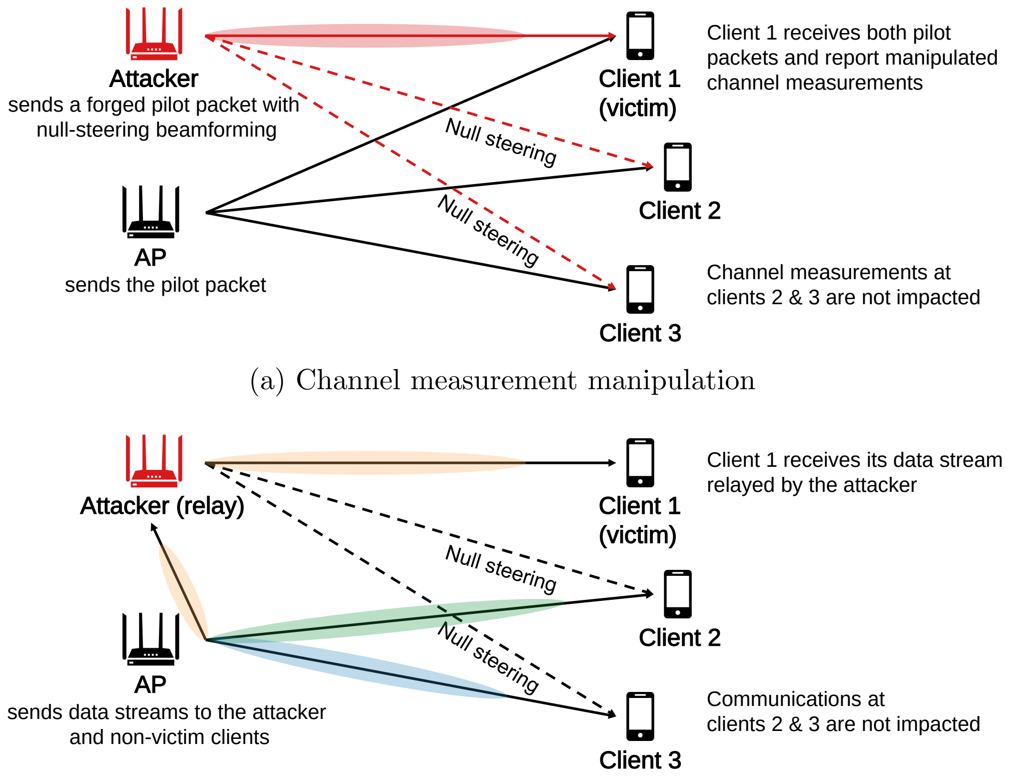
Wireless communication has become an essential part of modern society with a growing demand for high-speed and reliable wireless networks. In response to this demand, MIMO technology has been widely adopted in wireless communication systems due to its ability to improve spectral efficiency and enhance the quality of service [14, 36, 2, 16]. MU-MIMO further extends MIMO technology. It allows multiple users to communicate with a multi-antenna AP simultaneously at the same frequency by spatial multiplexing. MIMO technology has been incorporated into the latest wireless communication standards, such as IEEE 802.11ac [15] and 5G [46, 62]. The proliferation of wireless devices and the exponential growth of data traffic have also made MU-MIMO increasingly popular in both academic research and industrial applications in recent years [49, 88, 70, 57, 26].

While MU-MIMO technology offers significant benefits to wireless communication systems, it also introduces new security challenges. One of them arises from the channel sounding process [72]. To perform MU-MIMO, the AP needs to measure accurate CSI between the clients and itself in the channel sounding process. This is completed through the exchange of control packets between the clients and the AP. To ensure that clients at different locations can all participate in the channel sounding and later MU-MIMO communications, the AP broadcasts the control packets omnidirectionally. Additionally, to reduce the overhead of the channel sounding, the control packets are all transmitted in plaintext. The broadcasted plaintext packets make it possible for a potential attacker to passively eavesdrop on CSIs of clients or even launch active attacks.

Several studies in the literature have investigated vulnerabilities in the channel sounding process, leading to various attacks on downlink MU-MIMO transmissions. Tung et al. [72] and Mao et al. [53] propose active eavesdropping attacks for MU-MIMO systems with explicit or implicit channel feedback. To eavesdrop on the victim client, the malicious party executes the attacks by joining the network as a malicious client and sending forged CSI feedback or pilots to the AP to corrupt its channel measurements. The polluted channel measurements allow the attacker to receive signals containing the information intended for the victim client and the attacker itself. When signals intended for the attacker are known, the attacker can cancel them from the received signals and decode the messages meant for the victim from the remaining signals. Wang et al. [79] extend this attack model to attack multiple victims with more attacker devices as malicious clients.

While active eavesdropping attacks have been proven effective in compromising MU-MIMO systems, they place specific requirements on the attacker devices: (1) **Participation in targeted transmissions as client(s)**: If there are more clients than the maximum data streams an AP can support in one transmission, only a subset of clients is selected for each transmission based on channel conditions, user fairness, and system capacity [89, 70]. This client selection process can reduce the opportunities for successful eavesdropping attacks [77], especially in multi-victim scenarios where the number of attackers participating in the transmissions must be equal to or greater than the number of victims [79]. (2) **Prior knowledge of packets for malicious client(s)**: Attacker devices need to know contents of the packets intended for them as input for signal cancellation, which typically requires cooperating servers to transmit predefined data. (3) **Shared eavesdropped signals among attacker devices**: In multi-victim scenarios, the multiple attacker devices must collaborate and share eavesdropped signals to decode messages intended for the victims.

In this chapter, we present a novel eavesdropping attack in MU-MIMO systems, where the attacker only needs a multi-antenna full-duplex device. Our proposed attack consists of two phases, as illustrated in Figure 4.1. During the first phase, the attacker sends a forged pilot packet with null-steering beamforming to the victims while the AP sends the legitimate pilot packet to the clients. The pilot in the forged packet is manipulated so that the channels measured from this packet will cancel the AP-victim channel and inject the AP-attacker channel. In the second phase, the AP transmits the streams intended for the victims to the selected antennas at the attacker, who operates in MIMO full-duplex mode and relays the received streams to the victims.


Attacker
sends a forged pilot packet with null-steering beamforming
AP
sends the pilot packet
Null steering
Null steering
Client 1 (victim)
Client 2
Client 3
Client 1 receives both pilot packets and report manipulated channel measurements
Channel measurements at clients 2 & 3 are not impacted
Attacker (relay)
AP
sends data streams to the attacker and non-victim clients
Null steering
Null steering
Client 1 (victim)
Client 2
Client 3
Client 1 receives its data stream relayed by the attacker
Communications at clients 2 & 3 are not impacted

(a) Channel measurement manipulation

(b) Data stream relaying

Figure 4.1: Attack model of the active eavesdropping in MU-MIMO systems

Compared to existing eavesdropping methods targeting MU-MIMO systems, the proposed attack offers several significant advantages. First, *it demands less prior knowledge and coordination from the attacker*. To execute this attack, the attacker only requires control over a single multi-antenna full-duplex device. This malicious device does not need to join the network as a client together with the victims, and our attack does not rely on external servers to transmit any known data packets. What's more, *it maximizes eavesdropping opportunities* by operating independently of user selection results. The attack can be performed whenever the targeted victims are selected in the MU-MIMO transmissions. We conclude our contributions as follows:

- We propose a scalable active eavesdropping attack on MU-MIMO systems that requires less prior knowledge and coordination from attackers than existing attacks. Our proposed attack does not require attackers to join the communications between the AP and clients.

- We prove the effectiveness of our proposed attack in various settings, including cases with multiple victims, and with partial channel knowledge. The secrecy capacity[1] at the victims can be downgraded to zero.

- We evaluate the effectiveness of using physical-layer features, such as the AoA and Carrier Frequency Offset (CFO), to detect the proposed attack.

4.2 Background

4.2.1 Downlink MU-MIMO Transmissions

MU-MIMO is a space division multiplexing technology used by wireless communication systems. By creating multiple independent spatial streams, it allows a multi-antenna AP to communicate with multiple users simultaneously in one frequency band and thus significantly improves the overall network efficiency. Downlink MU-MIMO has been introduced as a mandatory feature to Wi-Fi protocols since 802.11ac [15] and has been supported by numerous commercial devices [26].

To generate independent spatial streams, the AP needs to measure the channels between the clients and itself during the channel sounding process. In MU-MIMO systems with explicit channel feedback, the AP first broadcasts a pilot packet. Upon receiving the pilot packet, each client measures the channels from the AP's antennas to itself based on the known pilot and sends the channel measurements back to the AP in the form of a feedback

[1] Secrecy capacity measures the maximum rate of the confidential information sent from the transmitter to the receiver under the threat of eavesdroppers. It can be calculated as $C_S = \max\{0, C - C_E\}$ where C denotes the legitimate channel capacity and C_E denotes the capacity of the eavesdropper.

packet. Based on the received feedback packets, the AP calculates the appropriate weights to apply to each data stream to transmit at its antennas to reduce interference among clients. The matrix formed by these weights is called the *precoding matrix*.

Precoding methods can be classified as linear and non-linear. Although the achievable capacity of linear precoding methods is slightly lower than some more complicated non-linear methods such as dirty paper coding, the linear precoding methods are widely preferred for their lower computation overheads [30, 81]. A representative example of linear precoding methods is zero-forcing beamforming [71, 89, 11]. Consider a case of an M-antenna AP and N single-antenna clients ($N < M$) and let $\mathbf{H}$ denote the N-by-M channel matrix between the AP and clients, where the entry in the i-th column and j-th row represents the channel value from the AP's i-th antenna observed at the j-th client. With zero-forcing beamforming, the precoding matrix $\mathbf{C}$ is calculated as:

$$\mathbf{C} = \mathbf{H}^{+} = \mathbf{H}^{*}(\mathbf{H}\mathbf{H}^{*})^{-1} \tag{4.1}$$

where $\mathbf{H}^{*}$ represents the conjugate transpose of $\mathbf{H}$, $(\cdot)^{+}$ represents Moore-Penrose inverse, and $(\cdot)^{-1}$ represents inverse.

Let $\mathbf{x}$ denote the N-by-1 data vector to be transmitted to the N clients, and $\mathbf{P}$ denote the diagonal N-by-N power allocation matrix $\mathrm{diag}(p_1, \cdots, p_N)$, where p_j represents the power allocated to the j-th client during transmission. The precoded vector to be sent at M antennas is $\mathbf{C}\sqrt{\mathbf{P}}\mathbf{x}$ and the received signal at receivers will be:

$$\mathbf{y} = \mathbf{H}\mathbf{C}\sqrt{\mathbf{P}}\mathbf{x} + \mathbf{n} = \mathbf{H}\mathbf{H}^{*}(\mathbf{H}\mathbf{H}^{*})^{\mathbf{-1}}\sqrt{\mathbf{P}}\mathbf{x} + \mathbf{n} = \sqrt{\mathbf{P}}\mathbf{x} + \mathbf{n} \tag{4.2}$$

where $\mathbf{n}$ denotes the noise vector observed at receivers. With precoding, the received signal at each receiver will have negligible interference from other clients, and each client can decode the signal independently without any knowledge about the other clients.

The power allocation matrix $\mathbf{P}$ needs to satisfy the constraint $\left\|\mathbf{C}\sqrt{\mathbf{P}}\mathbf{x}\right\|^2 \leq P$, where P is the total transmit power. The values of each entry in $\mathbf{P}$ can be decided by the specific power allocation strategy. The two most representative strategies are equal power allocation and maximal throughput power allocation. The equal power allocation maximizes fairness among concurrent receivers with $p_1 = \cdots = p_N$, and the maximal throughput power allocation maximizes the aggregated capacity of concurrent receivers with $\text{argmax}_{p_j} \sum_{j=1}^{N} \log_2(1+p_j/|n_j|^2)$, where n_j represents the noise observed at the j-th client and $|n_j|^2$ is the noise power.

4.2.2 Channel Sounding Process

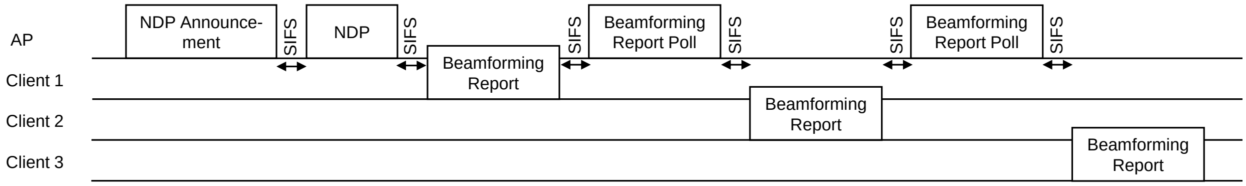


Figure 4.2: An example channel sounding process in 802.11ac for three clients

MU-MIMO systems rely on accurate channel measurements for effective beamforming and data stream separation. In MU-MIMO systems using explicit channel feedback, the AP initiates channel measurement by broadcasting a pilot packet to all clients. Upon receipt of the pilot packet, each client estimates the channel between itself and the AP and reports the result. Figure 4.3 demonstrates an example channel sounding process in 802.11ac, where the pilot packet is referred to as the Null Data Packet (NDP). Initially, the AP sends an announcement packet notifying clients about the start of this process. After the NDP packet is sent, the AP notifies each client to send the beamforming report poll packets (the first client to respond is specified in the NDP announcement packet). The Short InterFrame Space (SIFS) is the minimum separation time between high-priority frames, such as these control frames used for channel sounding.

4.3 Related Work

4.3.1 Eavesdropping in Wireless Networks

Wireless networks are highly susceptible to eavesdropping attacks due to the broadcast nature of wireless transmissions. Eavesdropping attacks can be categorized into passive or active eavesdropping attacks. Passive eavesdropping attacks involve an attacker intercepting wireless transmissions proactively. They are typically carried out using a wireless receiver that can capture the transmissions between legitimate transmitters and receivers in systems such as RFID systems [37], Wi-Fi systems [8], and millimeter wave communications [19]. On the other hand, active eavesdropping attacks involve an attacker participating in wireless transmissions. Examples include transmitting jamming signals [90] and initiating man-in-the-middle attacks [73].

One line of work closely related to our proposed attack is active eavesdropping attacks targeting MU-MIMO networks. It is worth noting that passive eavesdropping is unfeasible in MU-MIMO systems. Specifically, in MU-MIMO systems, only clients with predetermined AP-client channels can receive the corresponding data streams after signal precoding. Passive eavesdropping is ineffective unless the eavesdropper is placed within half a wavelength of the victim, which is usually a short distance and can increase the risk of detection. Among active eavesdropping attacks, reference [72] first proposes to let the attacker join the MU-MIMO communications as a malicious client and send forged CSI feedback during the channel sounding process. With carefully designed forged CSI, the attacker device can later receive signals containing mixed information for the victim and itself. By canceling the known signals intended for the attacker sent by a cooperating server, the attacker can eavesdrop on messages received by the other client in the network. In [53, 52], the authors propose a similar eavesdropping attack targeting networks adopting implicit channel feedback such as time-division duplex systems. Instead of sending forged CSI feedback, the malicious client sends forged pilots to the base station to pollute its channel measurements. In [79], the authors generalize this attack to multiple victim client scenarios and let multiple attackers forge CSI feedback as a polynomial function of the CSI of the victims and attackers. Considering that to perform this attack the number of attackers must be no less than the victims in one MU-MIMO communication, in [77], the authors study how to optimize the opportunity of having the attackers, i.e. malicious clients, being selected with the victim clients in the same transmissions.

4.3.2 Attacks in MIMO Systems

In addition to eavesdropping attacks, MIMO systems face various security threats. For example, in [72], a power attack is also proposed that allows a malicious client to boost its capacity at the cost of the victim's by forging CSI feedback. [68] introduces an attack using jamming signals during channel sounding to degrade the channel gain matrix estimate. [6] minimizes downlink transmission rates in multi-user massive MIMO networks through pilot contamination. And authors of [67] propose a known-plaintext attack that trains an adaptive filter to bypass the orthogonal blinding schemes that disturb an eavesdropper's signal reception.

4.4 Attack Model and Methodology

We make the following assumptions about the attacker:

(i) The attacker controls a multi-antenna full-duplex device whose antenna count is greater than or equal to the number of targeted victim clients. This attacker device always has sufficient transmit power.

(ii) The attacker device is within the communication ranges of the AP, victim clients, and optionally non-victim clients.

(iii) The attacker has some basic knowledge of the communication system, such as packet format and pilot for channel sounding.

(iv) The attacker device can anonymously query the channels from the victim clients, and optionally the non-victim clients to itself.

Assumption (iii) is based on the fact that pilots used for channel measurements are usually defined in corresponding standards and are thus commonly known by devices [53]. Combined with assumption (ii), the attacker is able to measure channels from the AP, victim clients, and optionally non-victim clients to itself from regular transmissions in the system, such as beacons, channel sounding packets for MU-MIMO user selection updates, and previous data transmissions. If some parties have not participated regular transmissions for a long time, the attacker can leverage assumption (iv) to query the channels of interest. Assumption (iv) has been proved feasible in real-world Wi-Fi networks, where an AP will always respond Clear-To-Send (CTS) frames to fake request-to-send frames [78], or acknowledgment frames to fake data frames [4] even if the client is unauthorized. To query the channels from clients, the attacker can send fake beacons and get the clients' responses [5].

With these assumptions, we propose an active eavesdropping attack on MU-MIMO systems with explicit channel feedback, outlined in two phases as shown in Figure 4.1. In the first phase, during the AP's channel measurement, the attacker simultaneously sends a forged pilot packet with null-steering beamforming to victims. The pilot in the forged packet manipulates channels measured from this packet to cancel the AP-victim channel and inject the AP-attacker channel. In the second phase, the AP precodes data streams with measured channels and transmits the stream intended for victims to selected attacker antennas. To ensure the communications for the victims are not interrupted, the attacker operates in the MIMO full-duplex mode and relays the received streams to the victims.

In the remainder of this section, we will first introduce the details of the attacker's actions with an example case where there is only one victim client and the attacker has prior knowledge of channels from all clients to itself. Then we will extend this attack to multi-victim cases and discuss the attack strategy when some non-victims' channels are not accessible.

4.4.1 Channel Measurement Manipulation

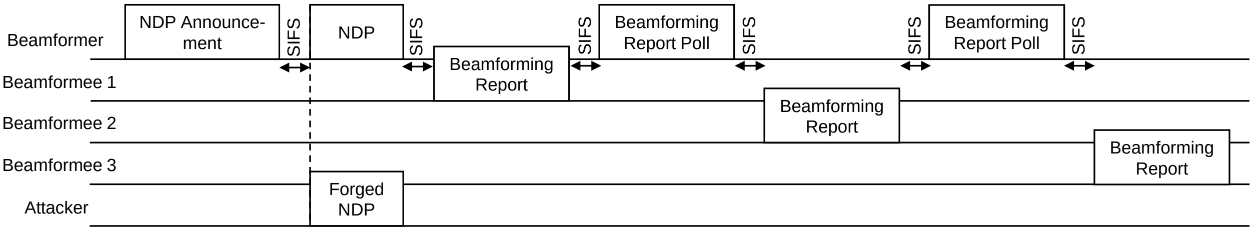


Figure 4.3: Channel measurement manipulation. The attacker sends a forged NDP simultaneously with the AP to alter the victim's channel measurement.

MU-MIMO systems rely on accurate channel measurements for effective beamforming. In MU-MIMO systems that utilize explicit channel feedback, the beamformer sends pilot packets to the beamformees. Then the beamformees measure their channels with the beamformer and reports the channel measurements to the beamformer. To manipulate the channel measurements at the victim, the attacker node transmits a forged pilot packet at the same time as the AP, as illustrated in Figure 4.3. The simultaneous transmission can be achieved by letting the attacker prepare the forged pilot packet in advance and send it one SIFS after the NDP announcement transmission.

The forged pilot is designed to contain the information of a channel that can cancel the AP-victim channel and inject the AP-attacker channel. When the victim client receives both the original pilot packet and this forged packet, its measurement result will be the channel between the AP and the attacker, rather than the channel between the AP and itself. To formulate this process, we consider the case of an M-antenna AP (the transmitter), N

clients (receivers), and a K-antenna attacker. Let $h_{t_i r_j}$ denote the channel from the AP's i-th antenna to the j-th client, $h_{t_i a_k}$ denote the channel from the AP's i-th antenna to the attacker's k-th antenna, and $x_{p,i}$ denote the original pilot value sent from the AP's i-th antenna. Assume that the first client is chosen as the victim, and the attacker wants to inject the channel of its first antenna $h_{t_i a_1}$ with a scaling factor α. Then the attacker needs to modify the forged pilot so that the victim can receive it as $(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i}$, where $h_{t_i r_1}$ can be heard from the victim's broadcasted beamforming report in the last round of MU-MIMO channel measurement (based on assumption (ii)), and $h_{t_i a_1}$ can be queried directly from the AP (based on assumption (iv)). Together with the original pilot packet $h_{t_i r_1} x_{p,i}$ received from the AP, the victim client will consider

$$y = (\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i} + h_{t_i r_1} x_{p,i} + n = \alpha h_{t_i a_1} x_{p,i} + n \tag{4.3}$$

as the received pilot value, where n is the noise value. And it will report a channel value close to $\alpha h_{t_i a_1}$ if the noise power is significantly smaller than the signal power.

While manipulating the channel measurements at the victim, the impact of forged pilots on non-victim clients should be minimized to avoid interference with their communications with the AP. To address this issue, the attacker utilizes zero-forcing beamforming on both the victim and non-victim clients when transmitting the forged pilot packet. In this transmission, we let the data intended for the victim to be $(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i}$ as derived above, and the data intended for the non-victims to be null, i.e. the data vector would be

$$\mathbf{x_{A,i}} = [(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i} \quad 0 \ \cdots \ 0]^T \tag{4.4}$$

where $(\cdot)^T$ denotes matrix transpose. Let $\mathbf{P_A}$ represent the power allocation matrix used by the attacker, where the attacker sets $p_{A,1} = 1$ with its sufficient transmit power. According to Equation 4.2, if the channel stays stable and the noise has significantly lower power than signals, the signal vector $\mathbf{y_{A,i}}$ received from the attacker is supposed to be very close to

$\sqrt{\mathbf{P_A}}\mathbf{x}$ in zero-forcing beamforming transmissions. Thus the victim will receive the forged pilot from the attacker while all other clients will receive zero, i.e.,

$$\mathbf{y_{A,i}} \approx [(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i} \quad 0 \quad \cdots \quad 0]^T \tag{4.5}$$

Meanwhile, the AP is also broadcasting the original pilot packet to all users. For the pilot value $x_{p,i}$, it will arrive at clients as its original value multiplied by corresponding channels between the i-th antenna of the AP to the clients. When noise is significantly weaker than signals, the signal vector $\mathbf{y_{T,i}}$ received from the AP will be

$$\mathbf{y_{T,i}} \approx [h_{t_i r_1} x_{p,i} \quad h_{t_i r_2} x_{p,i} \quad \cdots \quad h_{t_i r_N} x_{p,i}]^T \tag{4.6}$$

and the sum signal vector will be

$$\mathbf{y_i} = \mathbf{y_{A,i}} + \mathbf{y_{T,i}} \approx [\alpha h_{t_i a_1} x_{p,i} \quad h_{t_i r_2} x_{p,i} \quad \cdots \quad h_{t_i r_N} x_{p,i}]^T \tag{4.7}$$

This approach ensures that the victim client measures its channel as $\alpha h_{t_i a_1}$ while non-victim clients are less impacted. To control the power of the injected channels, we introduce the scaling factor α. The power of $h_{t_i a_1}$ can differ significantly from $h_{t_i r_1}$ due to variations in transmit power and locations between the AP and the attacker, which could affect the power allocation or even user selection results in MU-MIMO networks. The impact of this scaling factor on the attack efficiency will be evaluated in Section 4.5.2.

4.4.2 Data Stream Relaying

After a successful injection of the pilot signals, the AP will consider the AP-attacker channel as the channel to the victim, and transmit the victim's data stream to the attacker. To avoid interrupting the communication between the AP and the victim client, we let the attacker device work as a multi-antenna full-duplex relay during data transmissions. Similar to the pilot injection phase, the attacker performs null-steering zero-forcing beamforming while transmitting the relayed signal. We first consider the case with only one frequency band. Let $x_{d,j}$ denote the data intended for the j-th client at this frequency band. Assume that the first client is selected as the victim, then according to equation 4.2, the attacker will receive

$$y_{d,1} \approx \sqrt{p_1} x_{d,1} \tag{4.8}$$

during the data transmission when the noise power is neglectable. To relay the signal with null-steering zero-forcing beamforming, the attacker prepares the data vector to relay as

$$\mathbf{r_d} = [\beta y_{d,1} \quad 0 \quad \cdots \quad 0]^T \tag{4.9}$$

where the data stream for the victim is a scaled version of what the attacker receives about the victim's data, and the data streams for non-victims are null. We use β to denote the scaling factor used in data stream relaying. In this way, the victim client can get the information intended for it from the attacker, while other non-victim clients are less impacted by the relayed signals.

Many commonly used communication protocols use Frequency Division Multiplexing (FDM) methods that involve multiple subcarriers, such as OFDM used in LTE [2] and Wi-Fi standards since 802.11a [13]. A common practice to perform zero-forcing beamforming for packets with multiple subcarriers is to first multiply the modulated symbols with the precoding matrix at each subcarrier in the frequency domain, as mentioned in Section 4.2. After that, the transmitter uses inverse Fourier transform to convert the precoded symbols of all subcarriers to the time domain, and adds the Cyclic Prefix (CP) to form a complete OFDM symbol.

In the pilot injection phase, we use a similar method to obtain the precoded pilots. However, performing beamforming in the frequency domain is not feasible in the data transmission phase. This is because during the pilot injection phase, the attacker has sufficient time to prepare a forged pilot packet before the channel measurement process begins, with the pilot and channels to forge as prior knowledge. However, during the data transmission phase, the data intended for the victim user is unknown, and the attacker needs to perform zero-forcing beamforming and relay signals in real time. The delay time would be intolerable if the attacker chooses to decode the packet first, and later precode the data in the frequency domain and retransmit.

To facilitate real-time beamforming, we transform the precoding matrices in the frequency domain into precoding filters in the time domain. In [21], the authors implement a MIMO full-duplex relay with a construct-and-forward filter to make relayed signals constructively combine with the direct signals from the source. Our precoding filters can be implemented in the same way without introducing additional delay time.

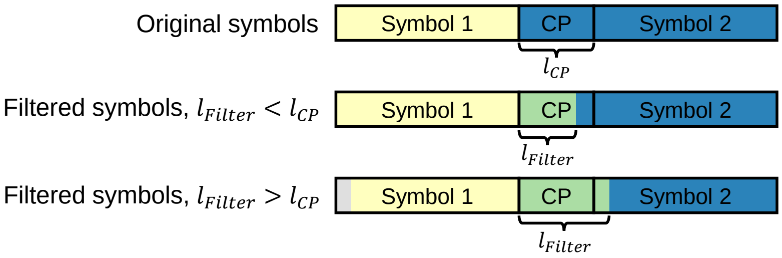


Figure 4.4: Impact of filter lengths on symbols. The green sections represent samples that contain information from both symbol 1 and symbol 2.

After being converted to the time domain, the precoding filters will have the same initial length as the number of subcarriers. However, when a large number of subcarriers are used, the length of precoding filters may exceed the maximum possible length permitted by the relay implementation. Moreover, if the precoding filter length is greater than the cyclic prefix length, applying the precoding filters will increase the inter-symbol interference, as shown in Figure 4.4. This increased interference can adversely affect data transmission, especially when the filter length is greater than the CP length.

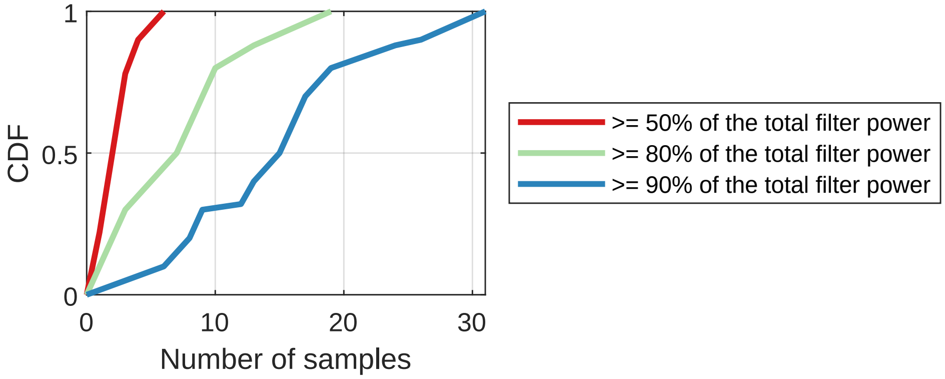
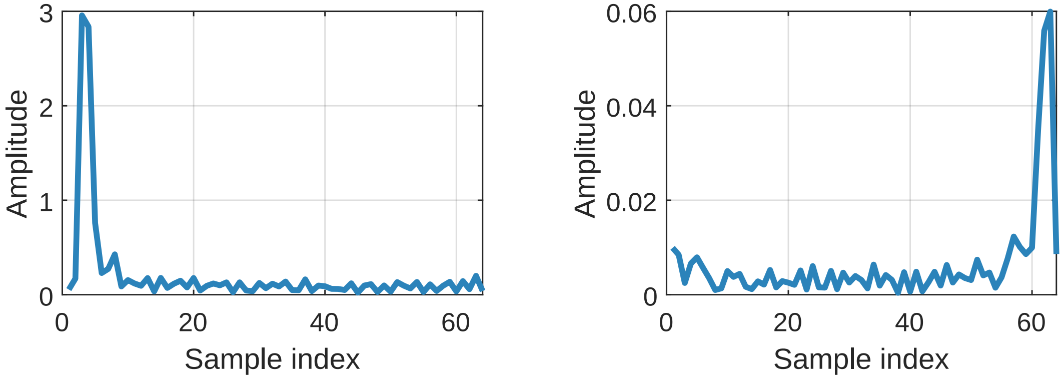


Figure 4.5: Distribution of the sample numbers taken to reach certain percentages of the total power of precoding filters

To constrain the length of precoding filters, we leverage the empirical observation that *the power of precoding filters usually concentrates on a few continuous samples.* Figure 4.5 presents the Cumulative Distribution Function (CDF) of the minimum number of continuous samples required to reach specific power levels, expressed as percentages of the total filter power. The figure shows results from 50 traces of an MU-MIMO network that serves three clients, with each channel having 64 subcarriers. Across all 50 traces, we find that selecting as few as 6 consecutive samples from the filters accounts for over 50% of the total filter power. Moreover, selecting up to 19 consecutive samples still accounts for over 80% of the total filter power.



(a) Example channel

(b) Precoding filter

Figure 4.6: An example channel in the time domain and its corresponding precoding filter at full length

We believe that this observed conclusion will hold in most cases. For each subcarrier, while calculating the precoding matrix $\mathbf{C}$ according to equation 4.1, we have observed that for every row of the matrix $\mathbf{HH}^*$, the magnitude of the diagonal entry is usually larger than or equal to the sum of the magnitudes of non-diagonal entries in this row, i.e., $\mathbf{HH}^*$ is usually a diagonally dominant matrix. This is because in $\mathbf{HH}^*$ the diagonal entries represent the channel powers of clients, and the non-diagonal entries represent the interference of channels between client pairs. In the inverse of a strict diagonally dominant matrix, such as $(\mathbf{HH}^*)^{\mathbf{-1}}$ in many cases, the largest entry in each column is on the diagonal [55]. Thus in $\mathbf{C} = \mathbf{H}^*(\mathbf{HH}^*)^{-1}$, the values from a scaled version of $\mathbf{H}^*$ can take a large part. When considering the precoding values of multiple subcarriers, the dominating conjugate of channel values in the frequency domain (values from $\mathbf{C}$'s of these subcarriers) will lead to a conjugate reverse of channel values in the time domain. Since in time-domain channel values, the power will usually concentrate on the first few samples, we can conclude that in the precoding filters, the filter power will usually concentrate on the last few samples, as shown in the example in Figure 4.6.

By selecting these continuous samples with dominant power, we can obtain shorter filters without significantly compromising beamforming performance. We will further discuss the impact of filter lengths on the data relaying performance in Section 4.5.2.

4.4.3 Scaling to Multiple Victims

The proposed attack can be expanded to multi-victim scenarios by utilizing the multiple antennas available to the attacker-controlled device. In the one-victim scenario discussed earlier, the attacker performs null-steering zero-forcing beamforming and has only one single non-null data stream to transmit the forged pilot or relayed data packets to the victim. However, in the multi-victim scenario, the attacker can create multiple non-null data streams, one for each victim, using a similar method. Theoretically, a K-antenna attacker node has the capability of creating up to K data streams, enabling it to attack up to K clients in the system. If the attacker has equal or more antennas than the AP ($K \geq M$), it can attack all clients served in one transmission. In this section, we assume that the attacker aims to attack V out of N clients in the system. For ease of explanation, we assume that the first V clients are chosen as the victims, although the attack can be applied to any subset of V clients.

To initiate the attack, the attacker must select V antennas to receive data streams intended for the V victims. As the channels from the AP to these selected antennas will later be injected into the AP's channel measurement as channels for victims, the attacker should choose the V antennas with the least correlated channels from the AP to ensure that these selected antennas can receive signals from the AP with minimal interference. For ease of explanation, we assume that the first V antennas are selected, and the v-th victim corresponds to the v-th antenna.

As previously defined in Section 4.4.1, we consider the case of an M-antenna AP, N clients, and a K-antenna attacker. We will continue to use the following notation: $h_{t_i r_j}$ for the channel from the AP's i-th antenna to the j-th client, $h_{t_i a_k}$ for the channel from the AP's i-th antenna to the attacker's k-th antenna, $x_{p,i}$ for the original pilot value sent from the AP's i-th antenna, and α for a scaling factor chosen by the attacker. In the pilot injection phase, similar to Equation 4.4, the attacker prepares a data vector

$$\mathbf{x_{A,i}} = [(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i} \ \cdots \ (\alpha h_{t_i a_V} - h_{t_i r_V})x_{p,i} \quad 0 \ \cdots \ 0]^T \tag{4.10}$$

where the first non-zero V values are the forged pilot value for the victims, and the following $N-V$ zeros are the null data streams for non-victim clients. With zero-forcing beamforming, the forged pilot will be received by clients as

$$\mathbf{y_{A,i}} \approx [(\alpha h_{t_i a_1} - h_{t_i r_1})x_{p,i} \;\; \cdots \;\; (\alpha h_{t_i a_V} - h_{t_i r_V})x_{p,i} \quad 0 \;\; \cdots \;\; 0]^T \tag{4.11}$$

together with the the signal $\mathbf{y_{T,i}}$ received from the AP

$$\mathbf{y_{T,i}} \approx [h_{t_i r_1}x_{p,i} \;\; \cdots \;\; h_{t_i r_V}x_{p,i} \quad h_{t_i r_{V+1}}x_{p,i} \;\; \cdots \;\; h_{t_i r_N}x_{p,i}]^T \tag{4.12}$$

the sum signal vector $\mathbf{y_i} = \mathbf{y_{A,i}} + \mathbf{y_{T,i}}$ will be

$$\mathbf{y_i} \approx [\alpha h_{t_i a_1}x_{p,i} \;\; \cdots \;\; \alpha h_{t_i a_V}x_{p,i} \quad h_{t_i r_{V+1}}x_{p,i} \;\; \cdots \;\; h_{t_i r_N}x_{p,i}]^T \tag{4.13}$$

In this way, the v-th victim client will measure its channel as $\alpha h_{t_i a_v}$, while measurements at non-victim clients are not impacted.

In the data relaying phase, the attacker behaves similarly to the method presented in Section 4.4.2, except that there will be V antennas receiving signals for the eavesdropping purpose, and now there are V data streams to relay to the clients with zero-forcing beamforming. Let $x_{d,j}$ denote the data intended for the j-th client, and $y_{d,j} \approx \sqrt{p_j}x_{d,j}$ denote the signal received by the attacker about the j-th victim's data. To relay the signal with null-steering zero-forcing beamforming to multiple victims, the attacker prepares the data vector to relay as

$$\mathbf{r_d} = [\beta y_{d,1} \;\; \cdots \;\; \beta y_{d,V} \quad 0 \;\; \cdots \;\; 0]^T \tag{4.14}$$

The precoding filters can be shortened in the same way as mentioned in Section 4.4.2.

4.4.4 Strategy with Partial Channel Knowledge

Assumptions (ii) and (iv) described at the beginning of Section 4.4 take into account scenarios where the attacker may not be aware of the channel information of all non-victim clients. This can occur due to two reasons. Firstly, if the attacker is located far away from some non-victim clients, their signals might not be detectable. Secondly, if the attacker has fewer antennas than the number of clients in an MU-MIMO transmission ($K < N$), it can generate only K data streams for V victim clients and relayed data packets, as well as $K - V$ null data streams for non-victim clients. In this scenario, the attacker might possess complete knowledge of the channels between itself and all clients, yet it can generate only a subset of data streams so it has to consider only partial channel feedback.

We suggest that the attacker can safely disregard the non-victim clients with the weakest RSS. In cases where some non-victim clients are not heard due to weak signal power, but the attacker's antenna count equals or exceeds the total number of victims and known non-victims, the attacker can proceed with the attack as usual. In cases where the attacker's antenna count is insufficient, it can ignore the channels with the lowest RSS values. For the non-victim clients who are neither known nor ignored by the attacker, their channel measurements and data transmissions with the AP can be impacted by the attacker's signals, since the attacker does not generate null data streams for them. Nonetheless, given that they receive weaker signals owing to their low RSS values, the interference will have a lesser impact on their communication as compared to other non-victim clients with higher RSS values. Thus, neglecting them will yield optimal global performance when the number of antennas at the attacker is limited.

4.5 Evaluation

4.5.1 Data Collection

Certain key information for the attack evaluation, such as the raw signal and Signal-to-Interference-plus-Noise Ratio (SINR), is not accessible in commonly used commercial devices. To overcome this limitation, we use WARP v3 software-defined radios to collect channels in a typical indoor office environment. The full-duplex device parameters are set as in [21]. We emulate the full-duplex relay scheme by first letting the AP transmit and the relay receive. After that, the AP remains silent, and the relay retransmits its received signal. Both received signals are later combined to form a single received signal during the attack.

We generate packets following the 802.11ac physical layer standard and use band 11 at 2.4 GHz for the experiments. The bandwidth is 20 MHz. Each channel measurement contains values of 64 subcarriers, where 52 of them are data subcarriers. The AP and the attack are both equipped with four VERT2450 antennas, and the AP serves three single-antenna clients unless otherwise specified.

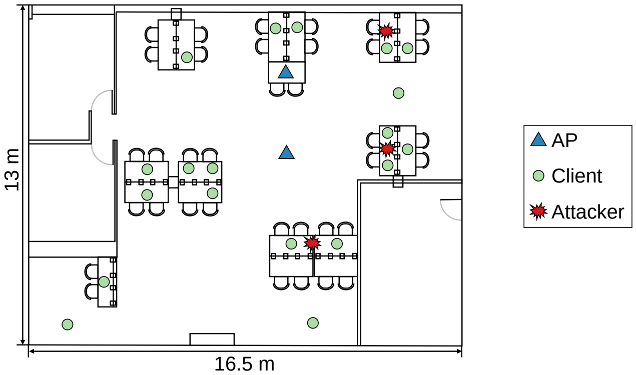
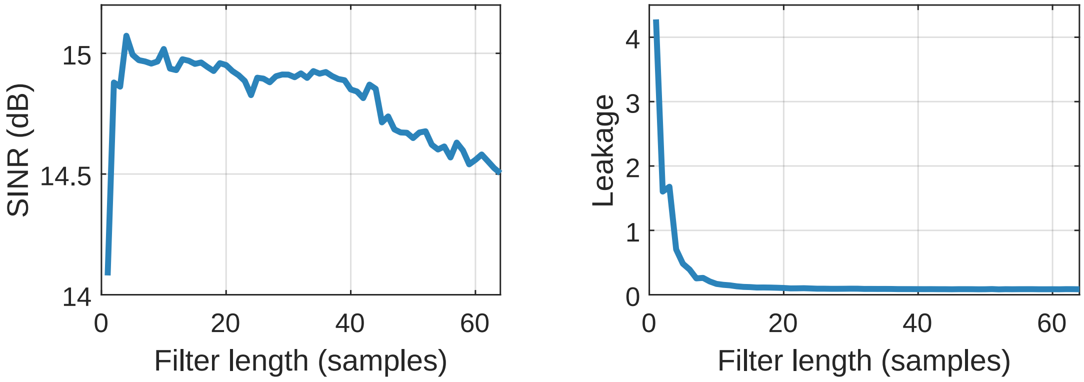


Figure 4.7: Layout of the office space and device locations

We consider a total of 30 settings in the typical office environment, and collect 5 channels with each setting. Each setting has a unique combination of AP/clients/attacker locations. The data collection is conducted over a period of two months and includes both LoS and NLoS settings. In the NLoS settings, we introduce everyday office objects as obstacles such as cubicle panels, chairs, and books. The office environment has an open-plan room layout, with dimensions of 13 m × 16.5 m. Figure 4.7 illustrates the office layout and some example locations of the AP, clients, and attacker.

4.5.2 Impact of Key Parameters on Eavesdropping Efficiency

Precoding filter lengths: To evaluate the impact of the precoding filter lengths on the attacker's data relaying performance, we select 50 traces collected at 5 locations with 3 clients and 1 victim, and emulate the data relaying performance with varying precoding filter length. We evaluate the filter lengths with two metrics: SINR at the victim of the received relayed data, and the leakage at the non-victims caused by the transmissions. We define the leakage to be the sum of received signal power at the non-victim clients from the attacker. Lower leakage means that the attacker will cause less interference to the non-victims' communications with the AP.



(a) SINR at the victim

(b) Leakage at non-victims

Figure 4.8: SINR at the victim client and leakage at non-victim clients with varying precoding filter length

Figure 4.8 displays the results. We observe that the SINR at victims initially increases as the filter length increases. This is because very short filters cannot fully perform the beamforming. However, as the filter length exceeds 20, we observe a decrease in SINR with filter length due to increased inter-symbol interference, as previously explained in Section 4.4.2. The average leakage at non-victims decreases with filter length, since the null data streams for non-victims will not suffer from inter-symbol interference and will only benefit from the better zero-forcing beamforming performance provided by longer filters. Therefore, we choose to use precoding filters of length 16 for balanced performance in the following experiments.

Scaling factors for channel manipulation and data stream relaying: The scaling factors in the pilot injection and data relaying phases, α and β, play a significant role in the eavesdropping efficiency and MU-MIMO communication performance. To eliminate the impact of varying RSS values of AP-attacker channels and keep the eavesdropping efficiency constant among different settings, we redefine the scaling factors with the RSS at non-victims as references, i.e.,

$$\alpha' = \frac{\sum_{k \in \mathbf{V}} \alpha \|\mathbf{H_{ta,k}}\| / |\mathbf{V}|}{\sum_{j \notin \mathbf{V}} \|\mathbf{H_{tr,j}}\| / (N - |\mathbf{V}|)}, \quad \beta' = \frac{\sum_{k \in \mathbf{V}} \beta \|\mathbf{H_{ta,k}}\| / |\mathbf{V}|}{\sum_{j \notin \mathbf{V}} \|\mathbf{H_{tr,j}}\| / (N - |\mathbf{V}|)} \tag{4.15}$$

where $\mathbf{V}$ denotes the victim set, $\mathbf{H_{ta,k}}$ denotes the channel matrix from the AP to the attacker's k-th antenna, and $\mathbf{H_{tr,j}}$ denotes the channel matrix from the AP to the j-th client, and N is the number of clients. Let C denote the number of subcarriers and A_t denote the number of antennas at the AP, then both $\mathbf{H_{ta,k}}$ and $\mathbf{H_{tr,j}}$ will be of size A_t-by-C.

To select appropriate scaling factors α' and β', we aim to fulfill the following requirements:

- The estimated signal-to-interference-plus-noise ratios (SINR) of all clients should closely match their actual SINRs during data transmissions.
- The SINR of the attacker should be close to or higher than that of the victims.
- The victims should achieve as high SINRs as possible.

The first requirement is to accommodate the rate adaptation, where the AP will choose a Modulation and Coding Scheme (MCS) to achieve the most appropriate transmission rate based on the current wireless channel conditions. If the transmission rate is set too high, there may be a high rate of packet loss and retransmissions due to errors in the wireless channel. On the other hand, if the transmission rate is set too low, the network throughput may be lower than it could be, leading to slower data transfer speeds. The second requirement maximizes eavesdropping efficiency, while the third requirement minimizes the impact of the attacker on the victims' communications.

To evaluate the effectiveness of different scaling factors in meeting three critical requirements, we consider an experimental scenario where an AP serves three clients, of which one is selected as the victim. We consider 10 settings in the office environment, and collect 5 traces in each setting. We vary the scaling factors α' from 0.25 to 1.5 and β' from 0 to 4 and use both equal power and maximal throughput power allocation strategies. We calculate three metrics corresponding to the three requirements. The first metric is the absolute difference between the SINR estimated from channel measurements and the actual SINR during data transmissions with an active attacker. The second metric is the SINR difference between the attacker and the victim. The third metric is the SINR of the victim during data transmissions while the attacker is on.

The results of these three metrics with varying scaling factors are shown in Figure 4.9. Regarding the SINR difference between estimated and actual SINRs, we have noticed that for regular MU-MIMO transmissions without an attacker, the difference is mostly between 2.5 and 3.5 dB. Figures 4.9(a) and 4.9(d) indicate that selecting $\alpha' = 0.75$ and $\beta' = 1.5$ or 2 can make the absolute difference stay in this range for both power allocation strategies. From Figures 4.9(b) and 4.9(e) we can see that in most cases the SINR at the attacker is higher than that at the victim and the secrecy capacity is decreased to 0. This outcome occurs because the AP considers the attacker as the victim and sends the data stream directly to it using zero-forcing beamforming. As a result, the victim receives both the attacker's relayed signal and interference from the AP's beamforming, explaining the SINR downgrade compared with the attacker. Although this communication quality downgrade is brought by the nature of our attack model, with proper scaling factor selection considering the first metric, the AP will take the victims as clients with inherently weaker channels and adjust the intended data transmission rates to accommodate them. Finally, Figures 4.9(c) and 4.9(f) show that with a fixed α' value, the victim's SINR increases with β', which indicates greater amplification is applied by the attack while relaying the signals during data transmissions.

It is worth noting that although the received power significantly influences the attack efficiency in different settings and we already considered its effect by normalizing α' and β' by non-victim RSS values, the received power is not the only factor that decides the attack efficiency and this normalization does not guarantee constant eavesdropping efficiency under all possible settings. While analyzing the results, we notice that channel correlations and noise levels also affect the optimal scaling factor choice. We leave it as a future work to propose an algorithm for determining optimal scaling factors.

In the following evaluations, we will use $\alpha' = 0.75$ and $\beta' = 2$. The corresponding α and β values are calculated with equation 4.15.

4.5.3 Overall Eavesdropping Efficiency

To investigate the overall eavesdropping efficiency, we collect traces with 30 settings with varying AP/clients/attacker locations in the office environment, and run the experiment 5 times with each setting. We consider a case of one AP serving three clients, and the AP adopts the equal power or maximal throughput power allocation strategies. To establish a baseline, while collecting each trace, we disable the attack once and monitor the signals received by the attacker. This baseline represents a receiver colocated with the attacker when the proposed attack is not executed. We refer to the baseline as a *passive eavesdropper*. The passive eavesdropper targets the same victim as the attacker in each transmission.

Figure 4.10 shows the SINR distributions at the victim, attacker, and passive eavesdropper. From the results, we observe that in almost all cases, the attacker gets higher SINRs than the victim, which means the victim's secrecy capacity can be downgraded to zero. Compared with the passive eavesdropper at the same location, an attacker performing our proposed active eavesdropping attack has an SINR gain of around 18 dB when the AP performs the equal power allocation strategy, and around 14 dB when the AP performs the maximal throughput power allocation strategy.

4.5.4 Eavesdropping Efficiency with Multiple Victims

To evaluate how the eavesdropping efficiency varies with the number of victims, we collect traces with 10 settings in the office environment with varying AP/clients/attacker locations, and run the experiment for 5 times with each setting. We consider a case of one AP serving three clients, and the AP adopts the equal power or maximal throughput power allocation.

Figure 4.12 shows the distribution of average SINRs at the victims and the corresponding selected antennas at the attacker with 1-3 victims. We can see that with both power allocation strategies, the SINRs of both victims and the attacker decrease as the victim count increases. This is because channels across attacker antennas are more correlated than channels across clients. In our test settings, the average correlation among channels from the AP to different antennas at the attacker is 0.623, while the average correlation among channels from the AP to different clients is 0.496. As more clients are selected as victims, the AP will take more channels from attacker antennas as the channels from the clients after the pilot injection, which makes the observed channels more correlated and causes more interference at beamformees, i.e. the attacker's antennas and non-victim clients. The signals with more interference are relayed to victims, which explains the SINR drops at them.

4.5.5 Eavesdropping Efficiency with Partial Channel Knowledge

To evaluate the eavesdropping efficiency with partial channel knowledge of non-victim clients, we collect traces with 10 settings with varying AP/clients/attacker locations in the office environment, and run the experiment for 5 times with each setting. We assume one of the three clients is selected as the victim, and the attacker is aware of channels of 0-2 non-victim clients. For the cases of one known non-victim, we assume the non-victim with higher RSS at the attacker is known and the other one is unknown.

Figure 4.11 depicts the SINRs of the attacker, the victim, and non-victim clients with different numbers of non-victim channels known at the attacker. From Figures 4.11(a) and 4.11(b), we observe that the number of known non-victim clients has a negligible effect on the attacker and the victim for both power allocation strategies. In contrast, the SINRs of unknown non-victim clients decrease significantly compared with known non-victim clients, as shown in Figures 4.11(c) and 4.11(d). We attribute this to the fact that the attacker does not generate null streams to unknown non-victim clients, which causes them to suffer from the interference of the relayed signals from the attacker, resulting in lower SINRs. Another observation is that the SINRs of known non-victim clients decrease as more non-victims are known at the attacker. This is because as more clients are involved in calculating the precoding matrices, the precoding values for the same client across subcarriers become less correlated and exhibit an uneven pattern. Compared with the scenario where precoding matrices have close amplitudes for the same client across subcarriers, this scenario yields received signals with lower powers. Since we assume that the attacker can adjust its transmit power to maintain the RSS at the victim at a constant level, the attacker needs to allocate higher transmit power per client to maintain the constant RSS levels when there are more known non-victim clients. Consequently, the higher transmit power causes more leakage at non-victim clients, resulting in lower SINRs.

4.5.6 Comparative Analysis with the Malicious Client Eavesdropping Attack

We compare the eavesdropping efficiency of our proposed attack with a representative eavesdropping attack for downlink transmissions in MU-MIMO systems [72]. The attack in [72] involves the attacker joining the MU-MIMO communications as a malicious client and reporting carefully-designed forged channels to receive signals containing information about both the intended message for the victim and itself. By utilizing the message intended for itself as prior knowledge, the attacker cancels the signals of this part and decodes the message intended for the victim with the remaining signals. We refer to this attack as the *malicious client* method and our proposed attack as the *malicious relay* method in the remainder of this section.

To compare the two attack methods, we collect traces with 10 different settings with varying AP/clients/attacker locations in the office environment. We run the experiment 5 times with each setting. Following the system setting in [72], we consider a case of one AP serving two clients, and the AP adopts either equal power or maximal throughput power allocation strategy. While implementing the eavesdropping attack in [72], we set the adjustable coefficient $w = 1$.

Figure 4.13 illustrates the SINRs at the attacker and the victim with two attack methods. In our proposed method, the attacker's SINR is around 8-10 dB higher than the victim's in most cases. In contrast, with the malicious client method, the attacker's SINR was mostly around 10-15 dB lower than the victim's. This difference can be attributed to the nature of the two attack methods. In our proposed attack, the attacker's SINR is higher than the victim's because the signals received by the attacker are only the signals intended for the victim, while at the victim they are a mix of the signals relayed by the attacker and the interference signal from the AP. In the attack with a malicious client, the attacker needs to estimate how the received signals of its own data streams are supposed to be and cancel the estimated signals from its received signal, leading to unavoidable cancellation errors during this processing that can decrease the victim's SINR. The victim's communications with the AP are not affected by the attacker, as has been proven by the authors, resulting in a higher SINR for the victim than the attacker.

Our evaluation results of reference [72] vary in values from the evaluations in the original paper, and we believe this is mostly due to the different testing environments. Our testing environment is a larger office room with more obstacles, and we have observed that the channels collected there can be dynamic over a short period. The more multipath-rich and dynamic environment increases the difficulty of running any systems that rely on accurate channel feedback. We started our implementation of [72] with simulated channels representing the ideal case and have observed a much smaller SINR difference between the attacker and the victim.

4.6 Countermeasures

In prior research on eavesdropping attacks in MU-MIMO systems, the authors have proposed various countermeasures using CSI. For example, in [72], the authors propose to use secret pilot values for channel sounding. In [53], a two-phase pilot commitment process is proposed to prevent unauthorized access to CSI.

We believe that these secure methods can effectively defend against our attacks by safeguarding pilot information. However, they require altering the existing communication protocols and can introduce extra control signal exchange overhead in MU-MIMO, which already has noticeable delays. Therefore, we evaluate the effectiveness of two representative features, AoA and CFO, used in physical-layer source authentication in detecting our attacks. These source authentication methods take metrics calculated during decoding as input. They are fully compatible with existing protocols and thus introduce minimal overhead.

4.6.1 Detection with Angle of Arrival

AoA is a fundamental concept in wireless communication that describes signal arrival direction at the receiver. The MUltiple SIgnal Classification (MUSIC) algorithm [66] estimates AoA with multi-antenna devices by spatial and temporal processing. Recent research has applied AoA profiles in detecting malicious activity in wireless networks, like traffic injection [87] and eavesdropping in MU-MIMO systems [77].

To detect our eavesdropping attack, the AP can employ the MUSIC algorithm with CSIs from the feedback packets as input and monitor changes in the AoA profiles for each client. Sudden deviations in the AoA profiles of a client can indicate the attack's initiation. This is because even though the feedback packets are sent by victim clients, the channel measurements within these packets post-pilot injection represent the AP-Attacker antenna channels, whose AoA profiles may differ from the AP-victim channels before the attack.

Table 4.1: Confusion matrix of AoA Detection

		Predicted	
		Positive	Negative
Actual	Positive	0.8	0.2
	Negative	0	1

We evaluate the effectiveness of using AoA to detect our proposed attack with the traces collected in Section 4.5.6. For the 5 traces collected at each setting, we use the first 3 traces to extract the AoA signature of the victim client, and use the victim's channels before and after the attack in the remaining traces as the input. We employ a simplified version of the method proposed in [87], i.e., getting AoA spectra of channel observations with the MUSIC algorithm, and extracting local maximum angles as features.

Figure 4.14 displays two example spectra. Figures 4.14(a) and 4.14(b) show that the AoA spectra closely match the victim's signature in the absence of an attack. During an attack, there are noticeable differences in the spectra for some cases such as in Figure 4.14(a). However, Figure 4.14(b) presents a more challenging scenario that is actually missed by the test detection method based on local maxima. The similar spectra may occur due to the closer locations of the attacker and the victim. Table 4.1 presents the confusion matrix of our tested AoA-based detection method. The AoA-based detection method achieves an accuracy of 90%, with an 80% TPR and a 100% TNR.

4.6.2 Detection with Carrier Frequency Offset

CFO represents the carrier signal frequency difference between the transmitter and receiver. It is a ubiquitous phenomenon in wireless communication systems and is usually caused by oscillator drifts or Doppler shifts. In [41], CFO is employed as a radiometric signature for device authentication based on transmitter-receiver oscillator biases. Given a carrier frequency offset of Δf, the received signal in the time domain will experience a phase rotation. If the original received signal without CFO is denoted as $y(t)$, the signal after accounting for the carrier frequency offset can be expressed as

$$y'(t) = y(t)e^{j2\pi\Delta ft} \tag{4.16}$$

Receivers need to estimate and correct the CFO to successfully demodulate and decode the received signals. The CFO can be estimated with the phase shift of repeated pilot symbols. Let $y_{pilot,1}$ represent a received sample at the first pilot symbol, $y_{pilot,2}$ represent the corresponding sample at the second pilot symbol, and T_s represent the symbol duration. The CFO can be estimated as

$$\Delta f \approx \frac{\angle y_{pilot,2} - \angle y_{pilot,1}}{2\pi T_s} \tag{4.17}$$

In practice, we can average the CFO estimations of all samples in the pilots for better accuracy.

To detect the proposed attack with CFO signatures, clients need to monitor changes in the CFOs between the AP and themselves over time. Since CFO values are already estimated with pilots for successful decoding, reusing CFO as authentication signatures will introduce minimal overhead. Sudden deviations in CFO values observed by a client can indicate the attack's initiation and this client is a victim. This is because during the attack victim clients receive mixed signals from the attacker and AP, leading to combined CFO values due to oscillator drifts.

We extract CFO distributions from 50 traces for the same AP-client pair as the signature, comparing them to CFO distributions before and during attacks from the experiment in Section 4.5.6. Figure 4.15 displays Probability Density Function (PDF) and Gaussian approximations, where clear distinctions can be observed between clean signatures and observations during attacks. When we approximate the data points with Gaussian distributions, the signature approximates to $\mu = -701.671$ Hz and $\sigma = 744.385$, while observations during attacks result in $\mu = -999.907$ Hz and $\sigma = 1229.99$. Meanwhile, the distributions of the signatures and observations before attacks are similar, as seen in Figure 4.15(b). The observations before attacks approximate to $\mu = -784.253$ Hz and $\sigma = 771.849$.

Table 4.2: Confusion matrix of CFO Detection

		Predicted	
		Positive	Negative
Actual	Positive	0.4	0.6
	Negative	0.1	0.9

To evaluate the detection accuracy with CFO, we use 4 of the 5 traces per setting to create distribution profiles with and without the attack. The remaining traces are employed for testing. We determine the CFO observation result by comparing the likelihood of PDF functions at that CFO value. The confusion matrix in Table 4.2 reports a 65% accuracy, with a 40% TPR and a 90% TNR.

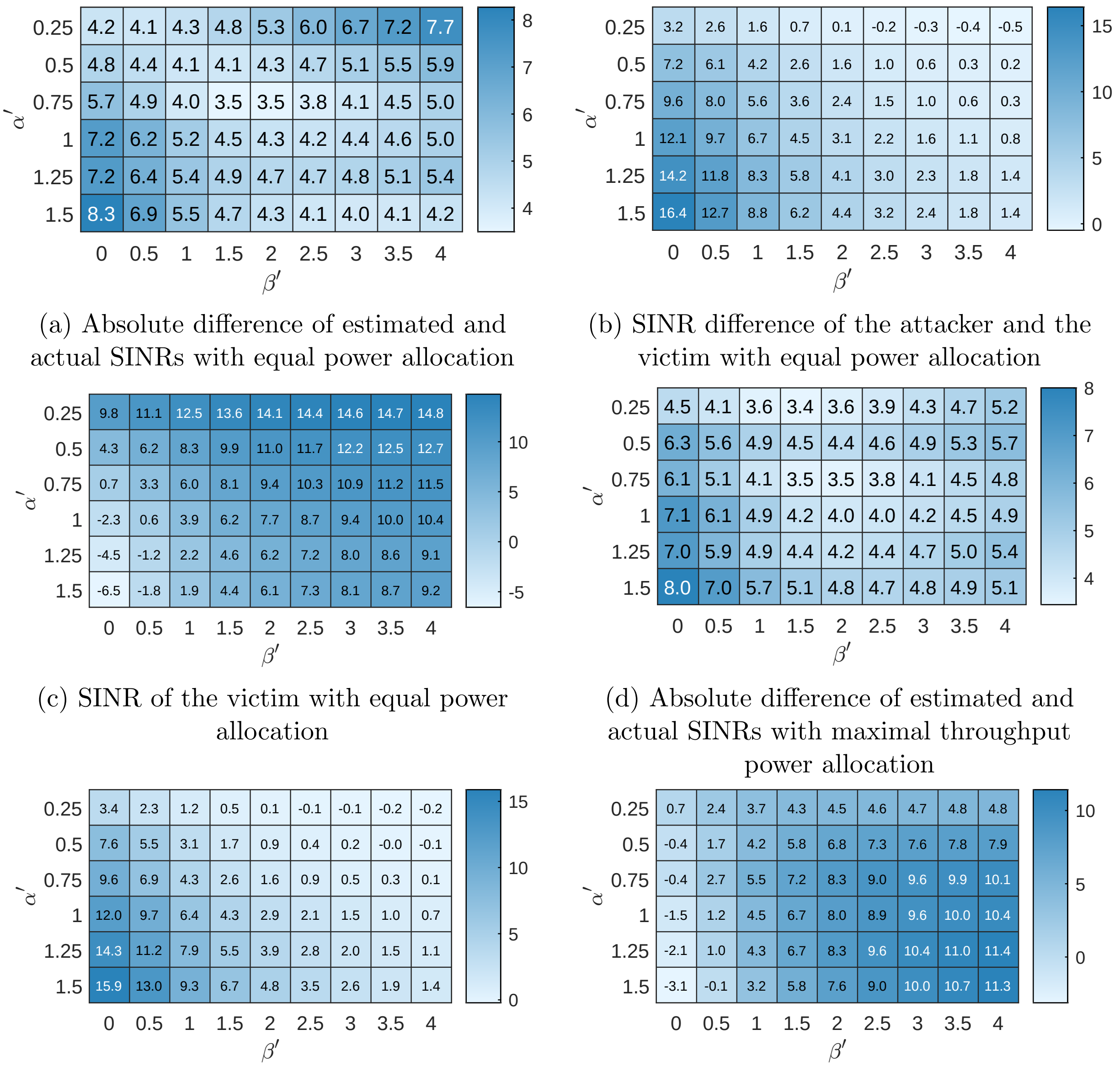
The low accuracy of the CFO-based method is due to significant overlap in CFO distributions. Observations within this overlap region have a high probability of misclassification. This overlap range may arise from close values of the inherent oscillator biases among the node pairs.

4.7 Discussion

In this work, we introduce an attack that lets an attacker access signals transmitted to one or more victims in downlink MU-MIMO transmissions. Given the wide adoption of end-to-end encryption in higher network layers, having access to the raw signals in the physical layer does not necessarily enable the attacker to decode data in the application layer. Given this context, the impact of our proposed attack, as well as other eavesdropping attacks focusing on lower network layers, can be understood as follows.

First, some applications of today, such as 15% websites [1] and some Android applications [59] are still not protected by end-to-end encryption. This gives the attackers a chance to decode the data transmitted if they can access signals in lower network layers.

Second, even with end-to-end encryption, physical layer eavesdropping attacks like ours can provide input for malicious traffic analysis applications. These applications can allow attackers to access user information, such as phrases during voice over internet protocol calls [82], motion and scene changes in videos [75], and visited websites [56].



(a) Absolute difference of estimated and actual SINRs with equal power allocation

(b) SINR difference of the attacker and the victim with equal power allocation

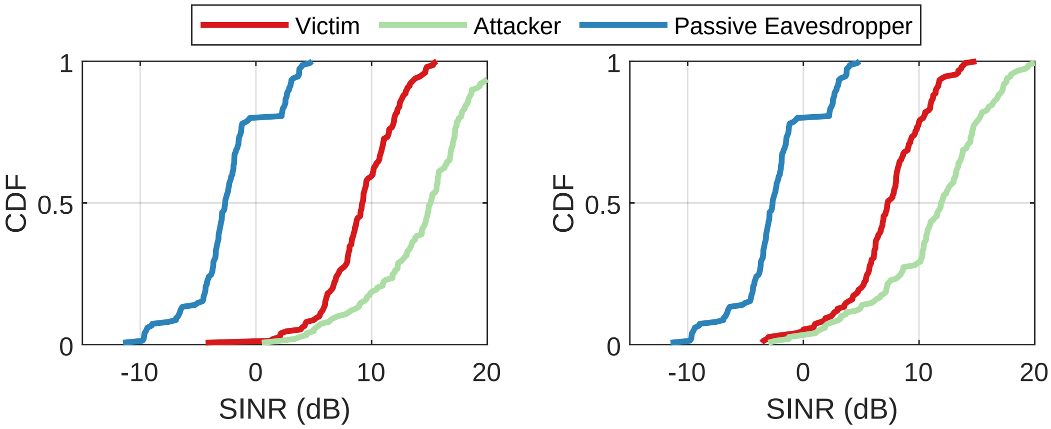
(c) SINR of the victim with equal power allocation

(d) Absolute difference of estimated and actual SINRs with maximal throughput power allocation

(e) SINR difference of the attacker and the victim with maximal throughput power allocation

(f) SINR of the victim with maximal throughput power allocation

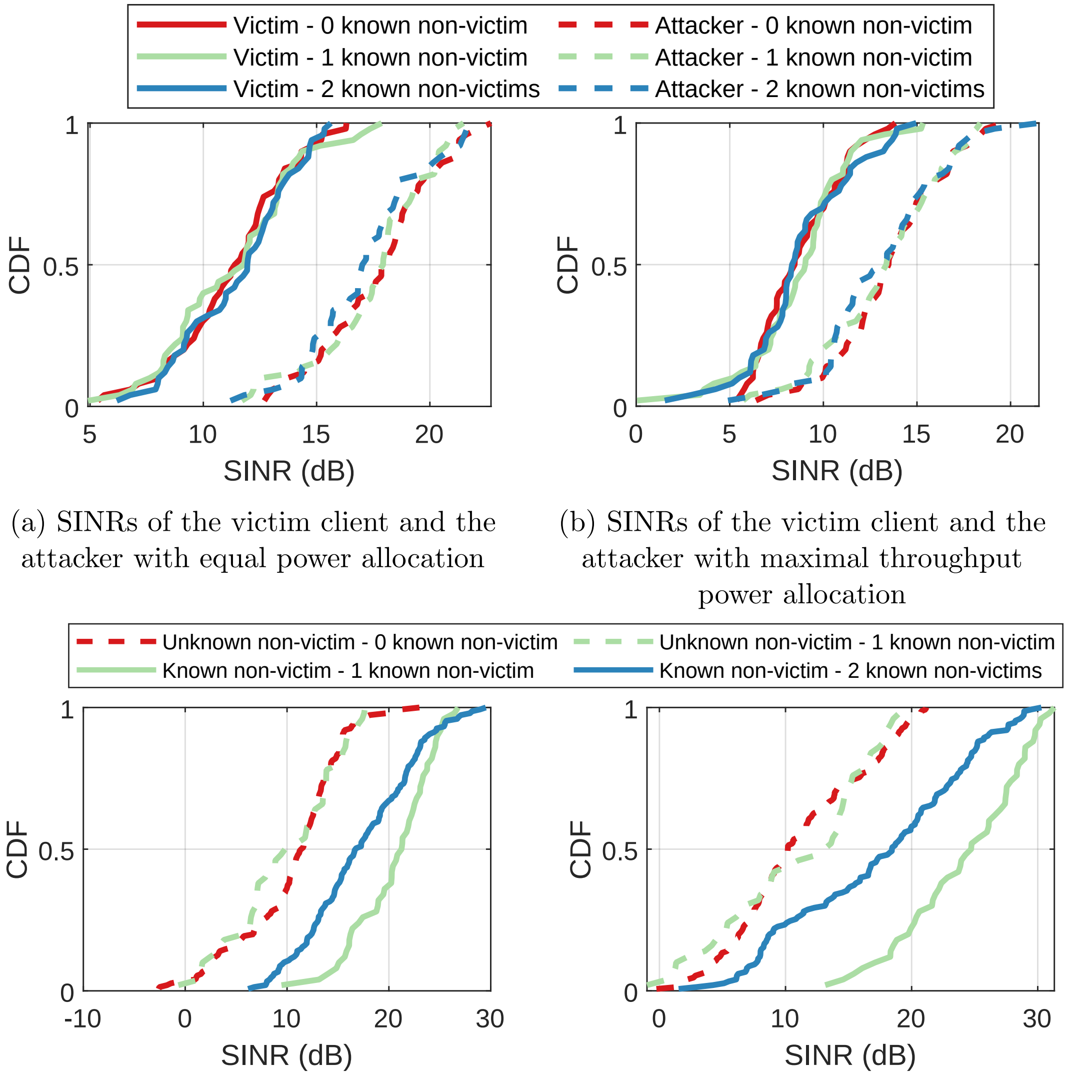
Figure 4.9: Metrics with varying scaling factors α' and β'



(a) Equal power

(b) Maximal throughput

Figure 4.10: Distributions of average SINRs at the victims, the attacker, and a passive eavesdropper located alongside the attacker when the proposed attack is not executed



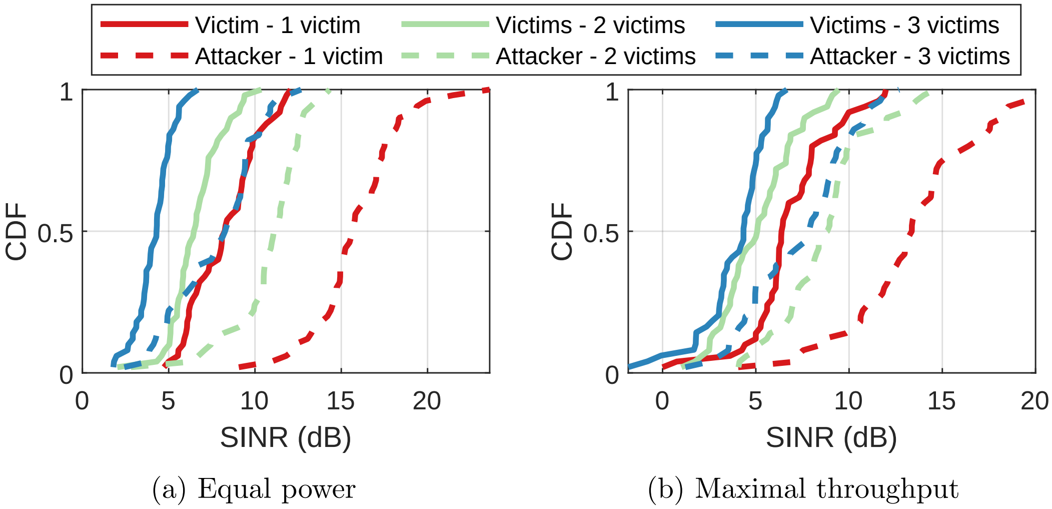
(a) SINRs of the victim client and the attacker with equal power allocation

(b) SINRs of the victim client and the attacker with maximal throughput power allocation

(c) SINRs of the non-victim clients with equal power allocation

(d) SINRs of the non-victim clients with maximal throughput power allocation

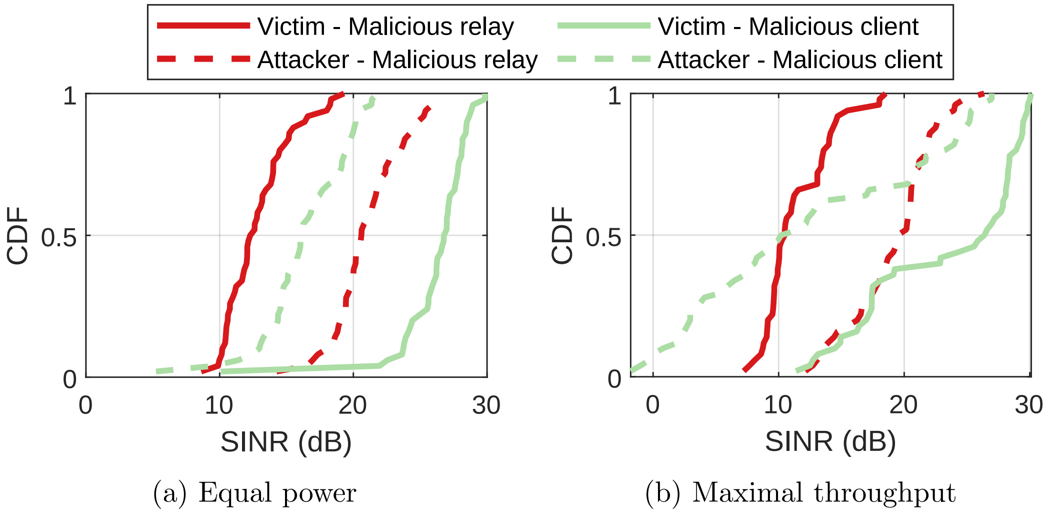
Figure 4.11: Distributions of SINRs with partial channel feedback



(a) Equal power

(b) Maximal throughput

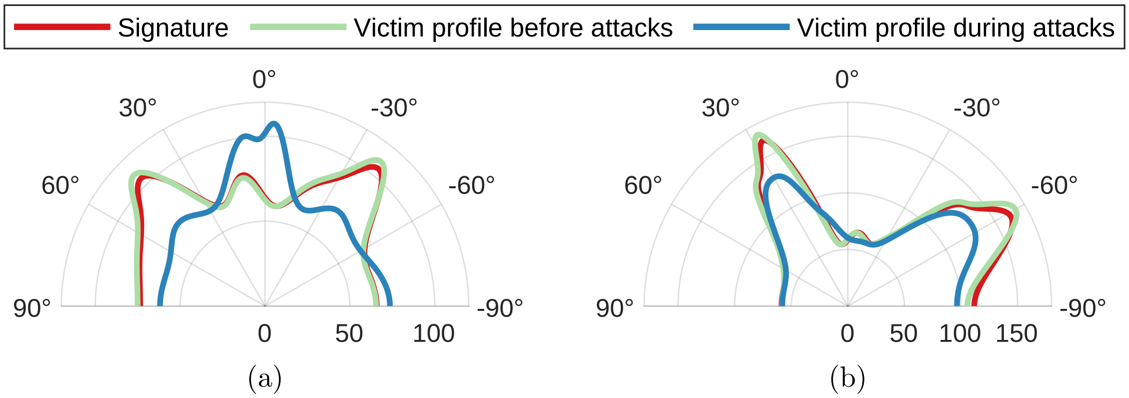
Figure 4.12: Distributions of average SINRs at the victims and the attacker with varying victim counts



(a) Equal power

(b) Maximal throughput

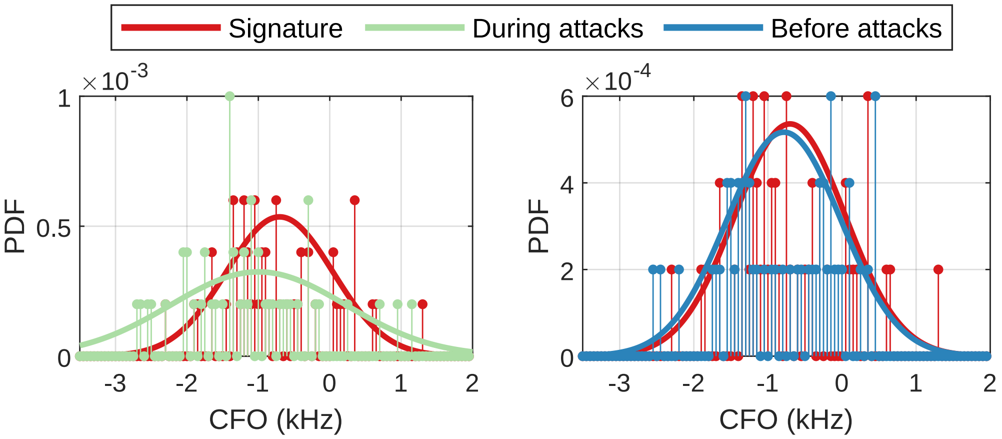
Figure 4.13: Distributions of average SINRs at the victims and the attacker with different attack methods



(a)

(b)

Figure 4.14: Example AoA spectra for the eavesdropping detection



(a) (b)

Figure 4.15: Example CFO distributions for the eavesdropping detection

Chapter 5: Future Work

This chapter introduces some future directions based on the attack described in Chapter 4. Section 5.1 focuses on some potential attacks targeting uplink MU-MIMO transmissions. In the proposed attacks, the multi-antenna attacker devices need to manipulate channel measurements of the clients in a similar way as in Chapter 4 as the initial steps. In Section 5.2, we discuss how the attackers in Chapter 4 or Section 5.1 can potentially bypass the AoA-based and CFO-based countermeasures described in Section 4.6.

5.1 Attacks for Uplink MU-MIMO Transmissions

The vulnerabilities of uplink MU-MIMO transmissions are less discussed in existing works than its downlink counterpart, which might be because of its more recent introduction to communication standards such as 802.11ax. To the best of my knowledge, the only existing attack model targeting uplink MU-MIMO transmissions is simply having one malicious client report forged channel feedback to reduce the throughput of all clients in the transmission [76, 83]. In this section, we will first introduce uplink MU-MIMO transmissions, then discuss about our plan to study two types of attacks other than the existing denial of service attacks: the eavesdropping attack and the spoofing attack.

5.1.1 Uplink MU-MIMO Systems

In uplink MU-MIMO transmissions, multiple client devices send multiple data streams simultaneously to a multi-antenna AP. Coordinating clients to precode data streams in uplink transmissions could introduce significant overhead. So the clients simply send multiple data streams to the AP, and it becomes the AP's responsibility to separate the data streams based on the received signals at its antennas. An uplink MU-MIMO transmission starts with a triggering frame sent by the AP. After receiving the trigger frame, the selected clients transmit their data packets simultaneously. The process of an example MU-MIMO transmission is shown in Figure 5.1.

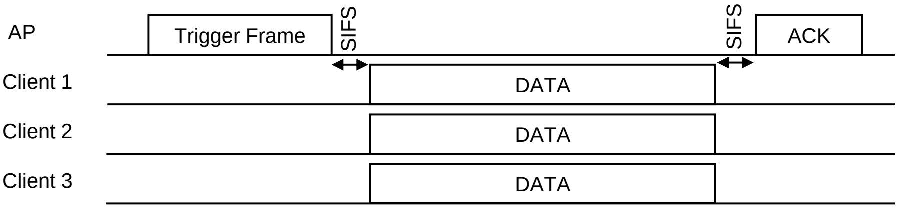


Figure 5.1: An example uplink MU-MIMO transmission for three clients

To separate the data streams received from the clients, the AP needs to know the channels from the clients to itself. With the channel information, the AP can use a *detector* designed with a detection algorithm to separate the data streams based on its received signals from all of its antennas and decode each of them [44]. Similar to precoders for downlink MU-MIMO, the detectors can also be classified as linear and non-linear detectors, and the linear ones are often preferred because of their lower computational complexity.

Consider the case of an M-antenna AP serving N single-antenna clients. A linear detector can be represented as an N-by-M detector matrix $\mathbf{A}$. With a detector matrix, the AP can estimate the signals sent from all clients $\hat{\mathbf{x}}$ as:

$$\hat{\mathbf{x}} \approx \mathbf{A}\mathbf{y} \tag{5.1}$$

where $\mathbf{y}$ represents the M-by-1 signal vector received by the M antennas of the AP and $\hat{\mathbf{x}}$ represents the N-by-1 estimated signal vector sent by the N clients.

The zero-forcing detector and Minimum Mean Squared Error (MMSE) detector are two popular options for linear detectors. Let $\mathbf{H}$ denote the M-by-N uplink channel matrix from clients to the AP. The zero-forcing detector aims to reduce the interference among data streams. Its detector matrix is defined as:

$$\mathbf{A}_{ZF} = \mathbf{H}^{+} = (\mathbf{H}^{*}\mathbf{H})^{-1}\mathbf{H}^{*} \tag{5.2}$$

Zero forcing can cause noise amplification if the channel matrix is ill-conditioned. In this case, the linear MMSE detector can be applied to reduce the sensitivity of linear receivers to the conditioning of the channel. The linear MMSE detector matrix is defined as:

$$\mathbf{A}_{MMSE} = (\mathbf{H}^{*}\mathbf{H} + \frac{\sigma_n^2}{E_t}\mathbf{I})^{\mathbf{-1}}\mathbf{H}^{*} \tag{5.3}$$

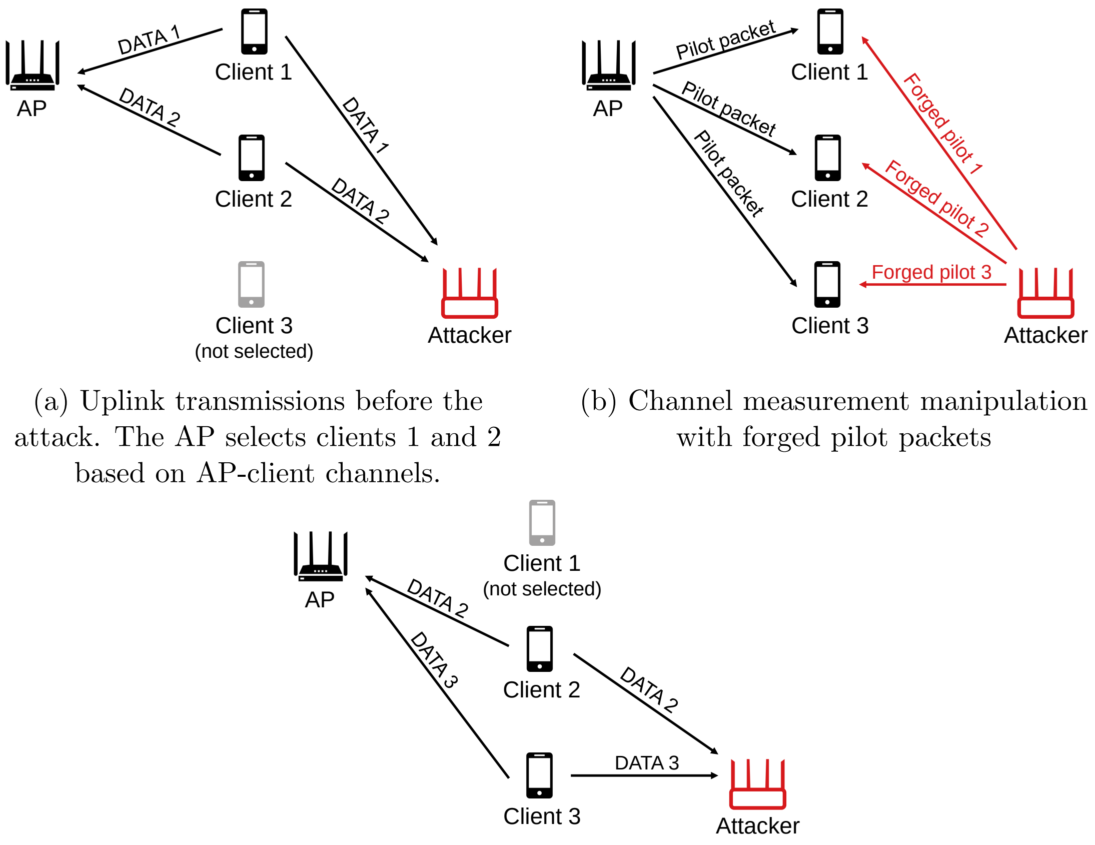
where $\mathbf{I}$ is the identity matrix, σ_n^2 is the noise power, and E_t is the average transmit power of clients.

The total throughput of uplink MU-MIMO transmissions is usually better when the received signal powers from all clients are comparable. To ensure this, in 802.11ax, the trigger frame contains information about the expected received signal power for the clients to adjust their transmit powers. Each client also leverages the trigger frame to correct the CFO between the AP and itself to a certain range for synchronization.

5.1.2 An Eavesdropping Attack for Uplink MU-MIMO Transmissions

In uplink MU-MIMO transmissions, if there is a passive eavesdropper within the clients' communication ranges, its received signals will be decided by the clients-attacker channel and the transmitted data streams. Compared to a potential passive eavesdropper targeting downlink MU-MIMO transmissions, a passive eavesdropper targeting uplink MU-MIMO transmissions requires less prior knowledge because the absence of precoding reduces the number of unknowns.

To perform passive eavesdropping on uplink MU-MIMO transmissions, the passive eavesdropper needs to equip at least as many antennas as the uplink data streams and have prior



(a) Uplink transmissions before the attack. The AP selects clients 1 and 2 based on AP-client channels.

(b) Channel measurement manipulation with forged pilot packets

(c) Uplink transmissions after channel manipulation. The AP selects clients 2 and 3 based on manipulated channels, which are easier for the attacker to eavesdrop on.

Figure 5.2: Attack model for the eavesdropping attack in uplink MU-MIMO transmissions

knowledge about the clients-attacker channel. The attacker can estimate the clients-attacker channel by listening to previous uplink transmissions and analyzing the signals with public knowledge about preambles. If some clients have not participated in uplink transmissions for a long time and the attacker cannot get the corresponding channel, it can also query the channels of interest anonymously by sending fake data frames [4]. With the channel information, the attacker can learn which clients will participate in an upcoming uplink MU-MIMO transmission by listening to and analyzing the transmission trigger frame, and develop its own detector as introduced in Section 5.1.1. Once the transmission begins, the eavesdropper receives signals from all of its antennas, applies the detector to separate the data streams, and decodes the transmitted packets.

Challenge: One significant challenge for the attacker to achieve good eavesdropping efficiency is introduced by the uncontrollable channel conditions between the attacker and participating clients. In MU-MIMO transmissions, the AP selects which clients to serve in one transmission based on various factors, such as the AP-clients channel conditions and fairness [70, 85]. In general, the less correlated the AP-clients channels are, the better the system throughput will be. The AP can also inform the clients to adjust their transmit powers with the user information fields in the trigger frame for more balanced channels and better performances [17].

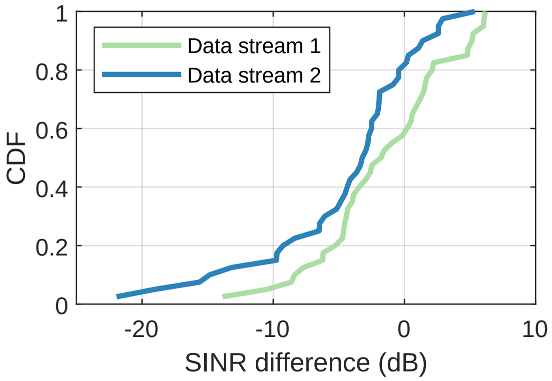


Figure 5.3: SINR difference distribution of the AP and a passive eavesdropper

For the passive eavesdropper, to achieve the best eavesdropping efficiency, it expects similar features in the clients-attacker channel as the AP, i.e., the channels from different clients are uncorrelated and the received signal powers from different clients are comparable. However, the clients are selected and configured based on the AP-clients channels. It is unlikely for a passive eavesdropper to see the same features in the clients-attacker channels when it is not located close to the AP. Figure 5.3 shows the SINR difference distribution between a passive eavesdropper and an AP which are both equipped with four antennas in 40 uplink MU-MIMO transmissions. The AP selects two out of three clients for the uplink transmissions based on their channel correlations. From the figure, we can see that SINRs at the eavesdropper are significantly lower than the AP in most transmissions. The SINR of the eavesdropper can be up to 21.9 dB lower than the AP.

Proposed work: For better eavesdropping efficiency, we hope the user selection and power pre-correction settings can be manipulated so that the channels between selected clients and the attacker can also be uncorrelated and of comparable attenuation, which further allows the attacker to separate the data streams and decode them. Since the AP's user selection and power pre-correction algorithms take AP-client channels as input, we propose to let the attacker manipulate channel measurements between the AP and clients to achieve this goal. The attack model is shown in Figure 5.2. Similar to the attack in Chapter 4, the first step is to let the attacker send forged pilot packets simultaneously with the AP to manipulate the channel measurements at the clients. The forged pilots are designed to let the AP make user selection and power pre-correction decisions favorable to the attacker. After the channel measurement manipulation, the attacker performs passive eavesdropping while the uplink MU-MIMO transmissions are happening.

5.1.3 A Spoofing Attack for Uplink MU-MIMO Transmissions

In the spoofing attack, the attacker aims to impersonate one or more clients and let the AP accept forged packets from the attacker instead of the legitimate packets from the victim(s). A critical condition to achieve this is to make the AP believe that the AP-attacker's channel is from the victim(s) instead of the attacker. Consider an attacker that simply transmits a forged packet simultaneously with legitimate clients in uplink MU-MIMO transmissions. The AP will receive the legitimate packets and this forged packet as

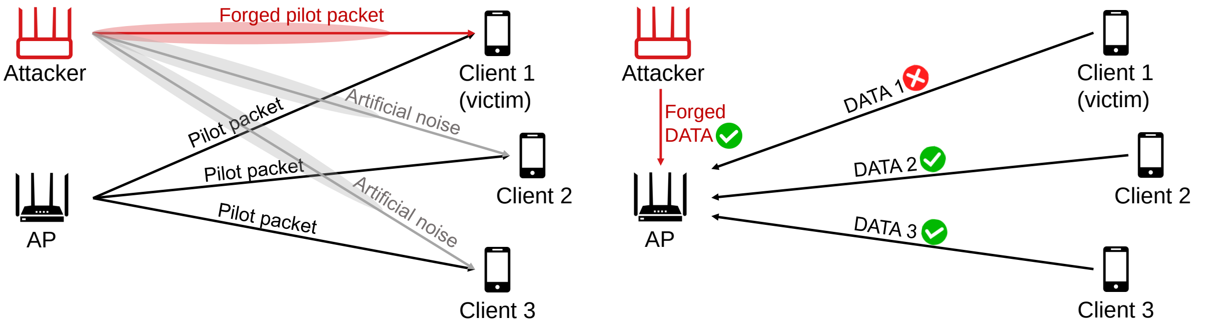
$$\mathbf{y} = \mathbf{H}_{clients-AP}\mathbf{x} + \mathbf{H}_{attacker-AP}\mathbf{x}_{forged} + \mathbf{n} \tag{5.4}$$

where $\mathbf{H}_{clients-AP}$ is the M-by-N channel matrix from the clients to the AP, $\mathbf{x}$ is the N-by-1 signal vector sent by the clients, $\mathbf{H}_{attacker-AP}$ is an M-by-K matrix representing the channel from the K antennas at the attacker to the M antennas at the AP, and $\mathbf{x}_{forged}$ denotes the K-by-1 signal vector for the forged packet. Upon receiving $\mathbf{y}$, the AP will estimate the data streams with the detector. Assuming the AP has derived a zero-forcing detector based on earlier measurements of the clients-AP channel, it will estimate the data streams as

$$\hat{\mathbf{x}} = \mathbf{A}_{ZF}\mathbf{y} = \mathbf{x} + \mathbf{A}_{ZF}\mathbf{H}_{attacker-AP}\mathbf{x}_{forged} + \tilde{\mathbf{n}} \tag{5.5}$$

where $\tilde{\mathbf{n}} = \mathbf{A}_{ZF}\mathbf{n}$. Considering that $\mathbf{A}_{ZF}$ is derived based on the clients-AP channel which is uncorrelated with the attacker-AP channel, $\mathbf{A}_{ZF}\mathbf{H}_{attacker-AP}\mathbf{x}_{forged}$ is very likely to affect all data streams (i.e., all rows in $\hat{\mathbf{x}}$) instead of one or some specific data streams.

However, if the attacker can make the AP believe that the attacker-AP channel is from the victim(s) before the uplink transmission starts, the AP will derive a detector matrix based on the channels from the attacker and non-victim clients. This detector matrix will separate data streams with specific channels, and treat other signals as interference or noise, similar to the $\mathbf{A}_{ZF}\mathbf{H}_{attacker-AP}\mathbf{x}_{forged}$ in Equation 5.5. Based on this insight, we propose a spoofing attack targeting uplink MU-MIMO transmissions.



(a) Channel measurement manipulation with forged pilot packet(s)

(b) Uplink transmissions during the attack. The AP takes the forged data stream as from the victim client and treat the legitimate data stream as interference or noise.

Figure 5.4: Attack model for the spoofing attack in uplink MU-MIMO transmissions

Proposed Work: The spoofing attacker controls a multi-antenna device located within the communication ranges of the uplink MU-MIMO system. The attack consists of two steps, as illustrated in Figure 5.4 with a one-victim example. The first step takes place during the channel measurement process. While the AP is sending the pilot packet, the attacker simultaneously sends a forged pilot packet to the victim, which contains a pilot that will cancel the legitimate pilot from the AP and inject the channel from one antenna of the attacker. In this way, when the victim later reports this channel measurement, the AP will believe this AP-attacker channel is from the victim. The second step happens during data transmissions, where the attacker sends forged data packets simultaneously with the legitimate clients. Since the detector matrix is derived with manipulated channel measurements, the AP will separate the forged data stream as the data stream from the victim.

While the forged data stream can replace the legitimate data stream from the victim, the victim is unaware of this attack happening and still transmits the legitimate data stream. The legitimate data stream will be treated by the AP as additional interference or noise and can potentially affect all data streams in this transmission. If the AP knows accurate SINR values at the clients, it will assign the best MCS for each client based on the observed SINR to achieve the optimal data rate. However, our proposed spoofing attack introduces a mismatch between the observed SINRs during channel measurements and the actual SINRs during data transmissions, which can negatively affect the transmissions of non-victim clients. To reduce this impact, we let the attacker send artificial noise to non-victims with beamforming during the first step of this attack. The noise levels are selected to make the observed and actual SINRs at non-victims align as much as possible.

This attack can also be extended to spoof multiple victims. During the channel manipulation, the attacker sends one forged pilot packet to each victim. The forged pilot packets are designed to cancel the corresponding legitimate pilots and inject channels from selected antennas at the attacker to the AP. For non-victim clients, the attacker still sends artificial noise to them to reduce the impact on AP's rate adaption. During the data transmissions, the attacker uses the selected antennas with injected channels to send corresponding forged data streams. Theoretically, an attacker with K antennas can spoof up to K clients. If the attacker is equipped with as many antennas as the AP, it can spoof any number of clients in a transmission and introduce little interference to the non-victims with artificial noise.

5.2 Attack Strategies to Countermeasure Awareness

If the attackers in Chapter 4 and Section 5.1 are aware of the countermeasures adopted by the legitimate parties, they can adjust their attack strategies to bypass some of the countermeasures. In this section, we will introduce attack modifications that can potentially help attackers bypass the countermeasures introduced in Section 4.6.

5.2.1 Adaptations to AoA-based countermeasures

In AoA-based detection methods, the AP monitors the AoA profiles for each client and reports potential attacks if sudden deviations are noticed. In attacks proposed in Chapter 4 and Section 5.1.3, the attackers will replace the victims to communicate with the AP. Thus the attackers will need to make their AoA spectra similar to the victims' to bypass the AoA-based countermeasures.

The most straightforward way for an attacker to create an AoA spectrum similar to the victim's is to locate the malicious device close to the victim device so that their channels to the AP can be correlated. When there is no obstacle between a victim device and the AP, the victim's AoA spectrum will be dominated by the angle of the LoS path. In this case, the attacker can choose a location in the LoS between the victim and the AP to get a similar AoA spectrum.

In some cases, the attacker might not be free to choose the location of the malicious device. However, if the attacker has some prior knowledge about the victim's AoA spectrum and controls a multi-antenna device, it can precode the data stream for this victim to alter the AoA spectrum observed by the AP. For example, assume that the victim's AoA spectrum is dominated by N_A angles $\{\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_{N_A}\}$ and $y_i(t)$ is the time-domain signal received from the i-th angle by the reference antenna at the AP. If the AP is equipped with a linear antenna array of M antennas separated by half-wavelength, its steering vector for the i-th angle will be

$$\mathbf{v}(\theta_i) = [1, e^{-j\pi sin(\theta_i)}, e^{-j2\pi sin(\theta_i)}, \ldots, e^{-j(M-1)\pi sin(\theta_i)}]^T \tag{5.6}$$

and the received signal vector of the M antennas at the AP will be

$$\mathbf{y}(t) = \overset{N_A}{\underset{i=1}{}} \mathbf{v}(\theta_i) y_i(t) + \mathbf{n}(t) \tag{5.7}$$

where $\mathbf{n}(t)$ is the noise vector. When the attacker wants to let the AP derive a similar AoA spectrum, it can first derive the expected received signal vector $\mathbf{y}(t)$ as in the above equations. It can then treat each antenna's received signal in $\mathbf{y}(t)$ as a separate data stream for this antenna, and precode these data streams as introduced in Section 4.2 to ensure that each antenna receives its corresponding data stream with minimal interference.

5.2.2 Adaptations to CFO-based countermeasures

In CFO-based countermeasures, the AP or clients can keep track of the CFOs between themselves and the other legitimate parties and report potential attacks if sudden deviations are noticed. To bypass CFO-based countermeasures, the attacker can modify its signals to transmit to control the CFO measured at the receiver. For example, assume that the attacker is aware of the CFO between the AP and itself $\Delta f_{AP-attacker}$ and the CFO between the AP and the victim $\Delta f_{AP-victim}$. Assume the original signal the attacker wants to send is $x(t)$. To make the victim measure the CFO as $\Delta f_{AP-victim}$ instead of $\Delta f_{AP-attacker}$, the attacker can rotate its signals to transmit as

$$x'(t) = x(t)e^{j2\pi(\Delta f_{AP-victim} - \Delta f_{AP-attacker})t} \tag{5.8}$$

It is usually easier for a device to access the CFO between itself and another device than that between two other devices. When the attacker cannot access $\Delta f_{AP-victim}$, it can measure $\Delta f_{victim-attacker}$ and estimate $\Delta f_{AP-victim}$ as

$$\Delta f_{AP-victim} = \Delta f_{AP-attacker} - \Delta f_{victim-attacker} \tag{5.9}$$

Chapter 6: Conclusion

In this thesis, we first introduce RelayShield, a system that detects relay attackers and recovers channels that have been manipulated by the relays. By resolving signal path information from observed channels, RelayShield is able to accurately detect relays and recover the original channels independent from any previously-collected signatures. Our extensive evaluations prove that both the relay detection and channel recovery modules are effective. Next, we introduce an active eavesdropping attack on MU-MIMO systems that exploits a multi-antenna full-duplex device. Our proposed attack involves two phases, where the attacker first transmits a forged pilot packet to the victim with zero-forcing beamforming to cancel out the AP-victim channel and inject the AP-attacker channel. It then relays the received data stream to the victim in full-duplex mode. We perform extensive experiments to evaluate the effectiveness of the proposed attack under various settings and demonstrate its capability to eavesdrop on AP-victim communications and bring the victims' secrecy capacity down to zero. We also investigate the feasibility of using physical-layer features to detect the proposed attack.

Bibliography

[1] Usage statistics of default protocol HTTPS for websites. `http://w3techs.com/technologies/details/ce-httpsdefault`, accessed: 2024-05.

[2] 3rd Generation Partnership Project (3GPP). TS 36.201: LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) physical layer; general description. 2010.

[3] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Enabling high-quality untethered virtual reality. In *NSDI*, pages 531–544, 2017.

[4] Ali Abedi and Omid Abari. Wi-fi says "hi!" back to strangers! In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, pages 132–138, 2020.

[5] Ali Abedi and Deepak Vasisht. Non-cooperative wi-fi localization & its privacy implications. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 570–582, 2022.

[6] Berk Akgun, Marwan Krunz, and O Ozan Koyluoglu. Vulnerabilities of massive mimo systems to pilot contamination attacks. *IEEE Transactions on Information Forensics and Security*, 14(5):1251–1263, 2018.

[7] Amani Al-Shawabka, Francesco Restuccia, Salvatore D'Oro, Tong Jian, Bruno Costa Rendon, Nasim Soltani, Jennifer Dy, Stratis Ioannidis, Kaushik Chowdhury, and Tommaso Melodia. Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 646–655. IEEE, 2020.

[8] Daniele Antonioli, Sandra Siby, and Nils Ole Tippenhauer. Practical evaluation of passive COTS eavesdropping in 802.11 b/n/ac WLAN. In *International Conference on Cryptology and Network Security*, pages 415–435. Springer, 2017.

[9] Tomoyuki Aono, Keisuke Higuchi, Takashi Ohira, Bokuji Komiyama, and Hideichi Sasaoka. Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels. *IEEE Transactions on Antennas and Propagation*, 53(11):3776–3784, 2005.

[10] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David Kotz. On the reliability of wireless fingerprinting using clock skews. In *Proceedings of the third ACM conference on Wireless network security*, pages 169–174, 2010.

[11] Ehsan Aryafar, Narendra Anand, Theodoros Salonidis, and Edward W Knightly. Design and experimental evaluation of multi-user beamforming in wireless lans. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 197–208, 2010.

[12] Ehsan Aryafar, Mohammad Amir Khojastepour, Karthikeyan Sundaresan, Sampath Rangarajan, and Mung Chiang. Midu: Enabling mimo full duplex. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 257–268, 2012.

[13] IEEE Standards Association. 802.11 a-1999—IEEE Standard for Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band, 1999.

[14] IEEE Standards Association. IEEE 802.11n-2009 IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput. IEEE, 2009.

[15] IEEE Standards Association. IEEE Standards 802.11ac-2013: Enhancements for very high throughput for operation in bands below 6 GHz, 2013.

[16] IEEE Standards Association. IEEE 802.16-2017 IEEE standard for air interface for broadband wireless access systems. IEEE, 2017.

[17] IEEE Standards Association. IEEE 802.11ax-2021 IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications amendment 1: Enhancements for high-efficiency WLAN. IEEE, 2021.

[18] Arjun Bakshi, Yifan Mao, Kannan Srinivasan, and Srinivasan Parthasarathy. Fast and efficient cross band channel prediction using machine learning. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[19] Sarankumar Balakrishnan, Pu Wang, Arupjyoti Bhuyan, and Zhi Sun. Modeling and analysis of eavesdropping attack in 802.11 ad mmWave wireless networks. *IEEE Access*, 7:70355–70370, 2019.

[20] Lars Baumgärtner, Alexandra Dmitrienko, Bernd Freisleben, Alexander Gruler, Jonas Höchst, Joshua Kühlberg, Mira Mezini, Richard Mitev, Markus Miettinen, Anel Muhamedagic, et al. Mind the gap: Security & privacy risks of contact tracing apps. In *2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom)*, pages 458–467. IEEE, 2020.

[21] Dinesh Bharadia and Sachin Katti. Fastforward: Fast and constructive full duplex relays. *ACM SIGCOMM Computer Communication Review*, 44(4):199–210, 2014.

[22] Dinesh Bharadia and Sachin Katti. Full duplex mimo radios. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 359–372, 2014.

[23] Dinesh Bharadia, Emily McMilin, and Sachin Katti. Full duplex radios. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 375–386, 2013.

[24] Stefan Brands and David Chaum. Distance-bounding protocols. In *Advances in Cryptology—EUROCRYPT'93: Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings 12*, pages 344–359. Springer, 1994.

[25] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127, 2008.

[26] Wireless CAT. List of MU-MIMO supported devices. 2022. `https://wikidevi.wi-cat.ru/List_of_MU-MIMO_supported_devices`.

[27] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. Airexpress: Enabling seamless in-band wireless multi-hop transmission. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 566–577, 2015.

[28] Lu Chen, Fei Wu, Jiaqi Xu, Kannan Srinivasan, and Ness Shroff. Bipass: Enabling end-to-end full duplex. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 114–126, 2017.

[29] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. Achieving single channel, full duplex wireless communication. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 1–12, 2010.

[30] Max Costa. Writing on dirty paper (corresp.). *IEEE transactions on information theory*, 29(3):439–441, 1983.

[31] Murat Demirbas and Youngwhan Song. An RSSI-based scheme for sybil attack detection in wireless sensor networks. In *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 565–570, 2006.

[32] Loh Chin Choong Desmond, Cho Chia Yuan, Tan Chung Pheng, and Ri Seng Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on Wireless network security*, pages 46–55, 2008.

[33] Saar Drimer, Steven J Murdoch, et al. Keep your enemies close: Distance bounding against smartcard relay attacks. In *USENIX security symposium*, volume 312, 2007.

[34] Aurélien Francillon, Boris Danev, and Srdjan Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

[35] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. Practical nfc peer-to-peer relay attack using mobile phones. In *Radio Frequency Identification: Security and Privacy Issues: 6th International Workshop, RFIDSec 2010, Istanbul, Turkey, June 8-9, 2010, Revised Selected Papers 6*, pages 35–49. Springer, 2010.

[36] GSMA. MIMO in HSPA: the Real-World Impact. 2012. `https://www.gsma.com/spectrum/wp-content/uploads/2012/03/umtsmimofinal.pdf`.

[37] Gerhard P Hancke. Practical eavesdropping and skimming attacks on high-frequency RFID tokens. *Journal of Computer Security*, 19(2):259–288, 2011.

[38] Gerhard P Hancke and Markus G Kuhn. An rfid distance bounding protocol. In *First international conference on security and privacy for emerging areas in communications networks (SECURECOMM'05)*, pages 67–73. IEEE, 2005.

[39] Jens Hermans, Roel Peeters, and Cristina Onete. Efficient, secure, private distance bounding without key updates. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 207–218, 2013.

[40] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. Smart locks: Lessons for securing commodity internet of things devices. In *Proceedings of the 11th ACM on Asia conference on computer and communications security*, pages 461–472, 2016.

[41] Weikun Hou, Xianbin Wang, Jean-Yves Chouinard, and Ahmed Refaey. Physical layer authentication for mobile systems with time-varying carrier frequency offsets. *IEEE Transactions on Communications*, 62(5):1658–1667, 2014.

[42] Kai-Cheng Hsu, Kate Ching-Ju Lin, and Hung-Yu Wei. Full-duplex delay-and-forward relaying. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 221–230, 2016.

[43] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. Practical, real-time, full duplex wireless. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 301–312, 2011.

[44] Yi Jiang, Mahesh K Varanasi, and Jian Li. Performance analysis of zf and mmse equalizers for mimo systems: An in-depth study of the high snr regime. *IEEE Transactions on Information Theory*, 57(4):2008–2026, 2011.

[45] Zhiping Jiang, Jizhong Zhao, Xiang-Yang Li, Jinsong Han, and Wei Xi. Rejecting the attack: Source authentication for Wi-Fi management frames using CSI information. In *2013 Proceedings IEEE INFOCOM*, pages 2544–2552. IEEE, 2013.

[46] Xingqin Lin, Jingya Li, Robert Baldemair, Jung-Fu Thomas Cheng, Stefan Parkvall, Daniel Chen Larsson, Havish Koorapaty, Mattias Frenne, Sorour Falahati, Asbjorn Grovlen, et al. 5G new radio: Unveiling the essentials of the next generation wireless access technology. *IEEE Communications Standards Magazine*, 3(3):30–37, 2019.

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

[48] Hongbo Liu, Jie Yang, Yan Wang, and Yingying Chen. Collaborative secret key extraction leveraging received signal strength in mobile wireless networks. In *2012 Proceedings IEEE INFOCOM*, pages 927–935. IEEE, 2012.

[49] Lingjia Liu, Runhua Chen, Stefan Geirhofer, Krishna Sayana, Zhihua Shi, and Yongxing Zhou. Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO. *IEEE Communications Magazine*, 50(2):140–147, 2012.

[50] Yanpei Liu, Stark C Draper, and Akbar M Sayeed. Exploiting channel diversity in secret key generation from multipath fading randomness. *IEEE Transactions on information forensics and security*, 7(5):1484–1497, 2012.

[51] Yunfei Ma, Nicholas Selby, and Fadel Adib. Drone relays for battery-free networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 335–347, 2017.

[52] Yunlong Mao, Ying He, Yuan Zhang, Jingyu Hua, and Sheng Zhong. Secure tdd mimo networks against training sequence based eavesdropping attack. *IEEE Transactions on Mobile Computing*, 19(12):2916–2932, 2019.

[53] Yunlong Mao, Yuan Zhang, and Sheng Zhong. Stemming downlink leakage from training sequences in multi-user mimo networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1580–1590, 2016.

[54] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 128–139, 2008.

[55] Alexander M Ostrowski. Note on bounds for determinants with dominant principal diagonal. *Proceedings of the American Mathematical Society*, 3(1):26–30, 1952.

[56] Eva Papadogiannaki and Sotiris Ioannidis. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[57] Hannaneh Barahouei Pasandi and Tamer Nadeem. LATTE: online MU-MIMO grouping for video streaming over commodity WiFi. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 491–492, 2021.

[58] Neal Patwari and Sneha K Kasera. Robust location distinction using temporal link signatures. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 111–122, 2007.

[59] Sajjad Pourali, Nayanamana Samarasinghe, and Mohammad Mannan. Hidden in plain sight: exploring encrypted channels in android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2445–2458, 2022.

[60] Yue Qiao, Kannan Srinivasan, and Anish Arora. Channel spoofer: Defeating channel variability and unpredictability. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 402–413, 2017.

[61] Yue Qiao, Ouyang Zhang, Wenjie Zhou, Kannan Srinivasan, and Anish Arora. Phycloak: Obfuscating sensing from communication signals. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 685–699, 2016.

[62] Qualcomm. Exploring 5G new radio: Use cases, capabilities timeline, 2016.

[63] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of rf distance bounding. In *USENIX security symposium*, pages 389–402, 2010.

[64] Michael Roland, Josef Langer, and Josef Scharinger. Applying relay attacks to google wallet. In *2013 5th International Workshop on Near Field Communication (NFC)*, pages 1–6. IEEE, 2013.

[65] Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Shamnaz Riyaz, Stratis Ioannidis, and Kaushik Chowdhury. Oracle: Optimized radio classification through convolutional neural networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 370–378. IEEE, 2019.

[66] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[67] Matthias Schulz, Adrian Loch, and Matthias Hollick. Practical known-plaintext attacks against physical layer security in wireless mimo systems. In *Network and Distributed System Security (NDSS) Symposium*, 2014.

[68] Shabnam Sodagari and T Charles Clancy. Efficient jamming attacks on MIMO channels. In *IEEE International Conference on Communications (ICC)*, pages 852–856. IEEE, 2012.

[69] Paul Staat, Kai Jansen, Christian Zenger, Harald Elders-Boll, and Christof Paar. Analog physical-layer relay attacks with application to bluetooth and phase-based ranging. *arXiv preprint arXiv:2202.06554*, 2022.

[70] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. Practical MU-MIMO user selection on 802.11 ac commodity networks. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 122–134, 2016.

[71] David Tse and Pramod Viswanath. *Fundamentals of wireless communication.* Cambridge university press, 2005.

[72] Yu-Chih Tung, Sihui Han, Dongyao Chen, and Kang G Shin. Vulnerability and protection of channel state information in multiuser MIMO networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 775–786, 2014.

[73] Yu-Chih Tung, Kang G Shin, and Kyu-Han Kim. Analog man-in-the-middle attack against link-based packet source identification. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 331–340, 2016.

[74] Deepak Vasisht, Swarun Kumar, Hariharan Rahul, and Dina Katabi. Eliminating channel feedback in next-generation cellular networks. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 398–411, 2016.

[75] Christopher Wampler, Selcuk Uluagac, and Raheem Beyah. Information leakage in encrypted IP video traffic. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2015.

[76] Fei Wang, Wei Xi, Jinsong Han, Kun Zhao, and Yuan Gao. Security in uplink MU-MIMO networks. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 351–352, 2017.

[77] Sulei Wang, Zhe Chen, Yuedong Xu, Qiben Yan, Chongbin Xu, and Xin Wang. On user selective eavesdropping attacks in MU-MIMO: CSI forgery and countermeasure. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1963–1971. IEEE, 2019.

[78] Wei Wang, Raj Joshi, Aditya Kulkarni, Wai Kay Leong, and Ben Leong. Feasibility study of mobile phone Wi-Fi detection in aerial search and rescue operations. In *Proceedings of the 4th Asia-Pacific workshop on systems*, pages 1–6, 2013.

[79] Xiaoshan Wang, Yao Liu, Xiang Lu, Shichao Lv, Zhiqiang Shi, and Limin Sun. On eavesdropping attacks and countermeasures for MU-MIMO systems. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pages 40–45. IEEE, 2017.

[80] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: device-free location-oriented activity identification using fine-grained WiFi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 617–628, 2014.

[81] Hanan Weingarten, Yossef Steinberg, and Shlomo Shitz Shamai. The capacity region of the gaussian multiple-input multiple-output broadcast channel. *IEEE transactions on information theory*, 52(9):3936–3964, 2006.

[82] Charles V Wright, Lucas Ballard, Scott E Coull, Fabian Monrose, and Gerald M Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 35–49. IEEE, 2008.

[83] Wei Xi, Rong Ma, Yuanhang Cai, and Kun Zhao. Prevent CSI spoofing in uplink MU-MIMO transmission. In *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition*, pages 13–18, 2015.

[84] Liang Xiao, Larry J Greenstein, Narayan B Mandayam, and Wade Trappe. Using the physical layer for wireless authentication in time-variant channels. *IEEE Transactions on Wireless Communications*, 7(7):2571–2579, 2008.

[85] Xiufeng Xie and Xinyu Zhang. Scalable user selection for MU-MIMO networks. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 808–816. IEEE, 2014.

[86] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[87] Jie Xiong and Kyle Jamieson. Securearray: Improving WiFi security with fine-grained physical-layer information. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 441–452, 2013.

[88] Qing Yang, Xiaoxiao Li, Hongyi Yao, Ji Fang, Kun Tan, Wenjun Hu, Jiansong Zhang, and Yongguang Zhang. BigStation: Enabling scalable real-time signal processingin large MU-MIMO systems. *ACM SIGCOMM Computer Communication Review*, 43(4):399–410, 2013.

[89] Taesang Yoo and Andrea Goldsmith. On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. *IEEE Journal on selected areas in communications*, 24(3):528–541, 2006.

[90] Yong Zeng and Rui Zhang. Active eavesdropping via spoofing relay attack. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2159–2163. IEEE, 2016.