Malicious Relay Detection and Legitimate Channel Recovery

Xingya Zhao
The Ohio State University
Columbus, Ohio, USA
zhao.2053@osu.edu

Wei-Han Chen
The Ohio State University
Columbus, Ohio, USA
chen.7323@osu.edu

Kannan Srinivasan
The Ohio State University
Columbus, Ohio, USA
kannan@cse.ohio-state.edu

ABSTRACT

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 paper, we propose 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.

CCS CONCEPTS

• **Networks → Mobile and wireless security**.

KEYWORDS

physical layer security, wireless network

ACM Reference Format:
Xingya Zhao, Wei-Han Chen, and Kannan Srinivasan. 2023. Malicious Relay Detection and Legitimate Channel Recovery. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3558482.3590193

1 INTRODUCTION

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 [2, 19–22] and source authentication [9, 18, 23, 32, 34].

WiSec '23, May 29-June 1, 2023, Guildford, United Kingdom

ACM ISBN 978-1-4503-9859-6/23/05…$15.00
https://doi.org/10.1145/3558482.3590193

However, a new type of attack has been proposed that compromises these link-based security protocols with full-duplex relays [24, 29]. 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 [9, 18, 23, 32, 34] and hardware-based methods [3, 8, 10, 17] 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.

2 BACKGROUND AND RELATED WORKS

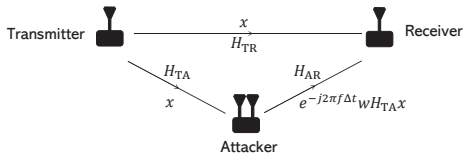


Figure 1: Attack model with a relay attacker. The receiver will measure the channel as $H_{TR}+e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$, where $e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$ is injected by the attacker.

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 1. Let us denote the genuine packet intercepted by the relay by x, the amplification factor by w, the carrier 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 $H = H_{TR} + e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$ with noise neglected. If w is sufficiently large, the injected component $e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$ 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 [29], 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 $e^{-j2\pi f\Delta t}wH_{TA}y$. The receiver receives $e^{-j2\pi f\Delta t}wH_{TA}H_{AR}y$ and calculates the channel as $e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$, which can be similar to the signature H given appropriate relay settings.

Similarly, in [24], 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 $e^{-j2\pi f\Delta t}wH_{TA}H_{AR}$.

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.

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:

$$h_f = ae^{-j2\pi\frac{df}{c}+j\phi} \tag{1}$$

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:

$$h_f = \sum_i^{N_p} a_i e^{-j2\pi\frac{d_i f}{c}+j\phi_i} \tag{2}$$

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.

$$\begin{aligned} \min \quad & \|H_{observed} - H\| \\ \text{s.t.} \quad & H = [h_{f_1}\ h_{f_2}\ \dots\ h_{f_{N_f}}] \\ & \forall k \in [1, N_f], h_{f_k} = \textstyle\sum_i^{N_p} a_i e^{-j2\pi\frac{d_i f_k}{c}+j\phi_i} \\ & \forall i \in [1, N_p], d_i > 0, a_i > 0, -\pi < \phi_i \leq \pi \end{aligned} \tag{3}$$

This concept has been utilized in several existing works for different purposes. For example, R2F2 [30] and OptML [4] focus on LTE cross-band channel prediction. They resolve multipath components from channel observations at one band to estimate channels at a different band. mD-Track [33] 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.

2.3 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 [9], the authors propose to defend against sybil attacks in sensor networks using RSS ratios from multiple receivers. [23] proposes using CSI signatures over time to detect an attacker impersonating transmitters in static environments. [34] proposed SecureArray, which utilizes multi-antenna APs 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 [10] propose to distinguish among unique devices through timing analysis of 802.11 probe request frames. In [8], the authors propose PARADIS, where differentiating artifacts of individual wireless frames are measured in the modulation domain to identify devices. [17] utilizes time-varying carrier frequency offset caused by oscillators for device authentication. The authors of [3] propose to use clock skew measurement as fingerprints of wireless devices. Recently, researchers have introduced machine learning techniques to help authentication [1, 27].

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.

2.4 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 [12, 13, 26] and Bluetooth systems [5, 16, 28]. 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 [7, 11, 14, 15, 25].

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 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

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

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:

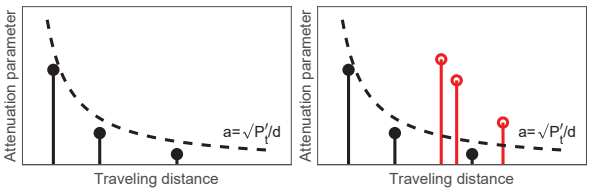
$$\forall i \in [1, N_p], a_i \leq \frac{\sqrt{P'_t}}{d_i} \tag{5}$$

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) 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:

$$\begin{aligned} h'_f &= e^{-j2\pi f \Delta t} w h_{TA} h_{AR} \\ &= e^{-j2\pi f \Delta t} w a_{TA} e^{-j2\pi \frac{d_{TA} f}{c} + j\phi_{TA}} a_{AR} e^{-j2\pi \frac{d_{AR} f}{c} + j\phi_{AR}} \\ &= w a_{TA} a_{AR} e^{-j2\pi \frac{(d_{TA} + d_{AR} + c\Delta t) f}{c} + j(\phi_{TA} + \phi_{AR})} \end{aligned} \tag{6}$$

The result obtained above can be compared with equation (1). This comparison reveals that the signal path through the relay attacker will be resolved as a component with parameters $a' = w a_{TA} a_{AR}$, $d' = d_{TA} + d_{AR} + c\Delta t$, and $\phi' = \phi_{TA} + \phi_{AR}$. Considering that a relay attacker tends to use a large amplification factor w to



(a) Without a relay (b) With a relay

Figure 2: Examples of multipath components

ensure the success of attacks and the delay Δt can be a few sampling intervals in state-of-the-art full-duplex implementations [6], 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 (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 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 (5), which is visualized as dashed lines in figure 2. If all components comply with the constraint, as seen in figure 2a, 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 2b, 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 4 and present their details.

4 SYSTEM DESIGN

The RelayShield system consists of two components: a relay detection module and a channel recovery module. As depicted in figure 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.

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.

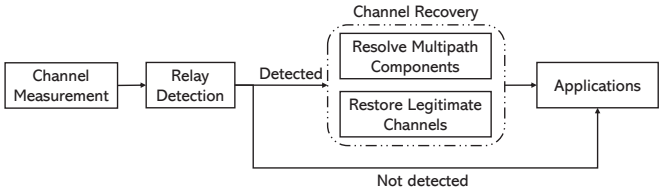


Figure 3: An overview of RelayShield

4.1 Relay Detection

In section 3, we have discussed the expected signal path parameters for channels from a single source. We observe that while optimizing equation (3) with the additional constraint in equation (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.

4.2 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 4.1. 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.

$$
\begin{aligned}
\min \quad & \|H_{observed} - H_{TR} - H_A\| - \alpha \textstyle\sum_m^M a_{A,m} \\
\text{s.t.} \quad & H_{TR} = [h_{TR,f_1}\ h_{TR,f_2}\ \dots\ h_{TR,f_{N_f}}] \\
& H_A = [h_{A,f_1}\ h_{A,f_2}\ \dots\ h_{A,f_{N_f}}] \\
& \forall k \in [1, N_f], h_{TR,f_k} = \textstyle\sum_n^N a_{TR,n} e^{-j2\pi \frac{d_{TR,n} f_k}{c} + j\phi_{TR,n}} \\
& \forall k \in [1, N_f], h_{A,f_k} = \textstyle\sum_m^M a_{A,m} e^{-j2\pi \frac{d_{A,m} f_k}{c} + j\phi_{A,m}} \\
& \forall m \in [1, M], a_{A,m} > 0, -\pi < \phi_{A,m} < \pi \\
& \forall m \in [1, M], \|d_{A,m} - d_{A,m,init}\| < r_d \\
& \forall n \in [1, N], a_{TR,n} > 0, -\pi < \phi_{TR,n} < \pi \\
& \forall n \in [1, N], \|d_{TR,n} - d_{TR,n,init}\| < r_d \\
& \forall n \in [1, N], a_{TR,n} \leq \frac{\sqrt{P'_t}}{d_{TR,n}}
\end{aligned}
\tag{7}
$$

We formulate the optimization problem as shown in equation (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.

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:

$$
\begin{aligned}
H_{TR} &= [h_{TR,f_1}\ h_{TR,f_2}\ \dots\ h_{TR,f_{N_f}}] \\
h_{TR,f_k} &= \textstyle\sum_n^N a_{TR,n} e^{-j2\pi \frac{d_{TR,n} f_k}{c} + j\phi_{TR,n}}
\end{aligned}
\tag{8}
$$

4.3 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.

To simulate channels from transmitters, we use algorithm 1. 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 (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.

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.

5 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 4.3. 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

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 (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 π
$\vec{a}$ = 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**
      $h_i = h_i + a_j e^{-j2\pi \frac{d_j f_i}{c} + j\phi_j}$
    end
  end
end
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
Δt_{min}, Δt_{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
Δt = a random number between Δt_{min} and Δt_{max}
$\vec{h_a} = e^{-j2\pi\Delta t}\vec{h_a}$
r = a random number between r_{min} and r_{max}
$\vec{h} = \vec{h_{tr}} + \frac{\|\vec{h_{tr}}\|}{\|\vec{h_a}\|} r\vec{h_a}$

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.

6 EVALUATION

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

6.1 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 [6]. 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.

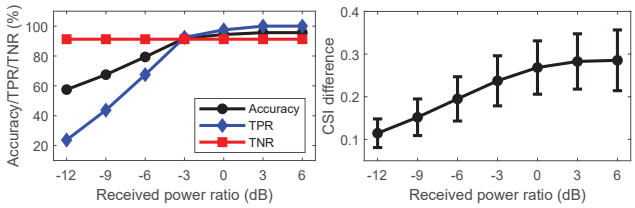
The first two steps are completed within the coherence time for each run of the experiments. The last step is done offline, where we can adjust the received power and delay time of $y_{r,2}$ to emulate different relay settings. We use the transmitter-receiver channels measured from the $y_{r,1}$ packets 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 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 $\times$ 7.5 m area. The office environment is a 12 m $\times$ 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.

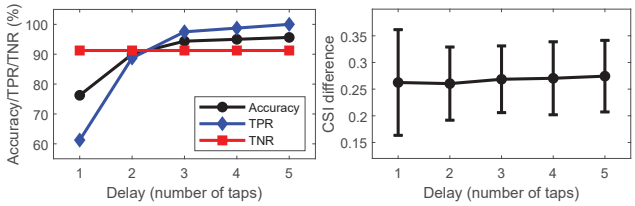
6.2 Relay Detection

Several prior works have employed CSI signatures for packet source identification [18, 23]. 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 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 [6]. Figures 4 and 5 depict the evaluation results.



(a) Relay detection module (b) CSI difference

Figure 4: Detection accuracies, true positive rate (TPR), true negative rate (TNR), and CSI difference results with different received power ratios from relays vs. transmitters and a 3-tap relay delay



(a) Relay detection module (b) CSI difference

Figure 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

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 4a. 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 4a 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 4b. 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 [24, 29], 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 TNR and 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.

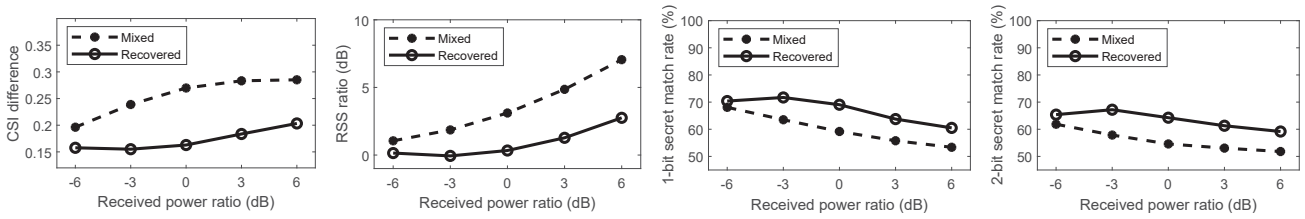
6.3 Channel Recovery

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:

- 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 [19]: 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.
- E-eyes [31]: 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).

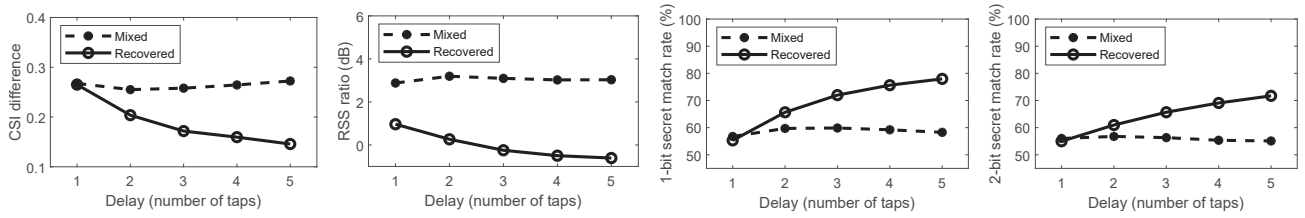
Channel Metrics and Recovered Channels for Shared Secret Generation. Recovery results with different relay settings are shown in figures 6 and 7. The *mixed* channels are the injected channels before recovery, which are mixes of the legitimate channels and channels through the relay.

As shown in figure 6a, 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 7a, 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 6b and 7b 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.



(a) CSI difference (b) RSS ratio (c) CSI 1-bit secret match rate (d) CSI 2-bit secret match rate

Figure 6: Recovery results with different received power ratios from relays vs. transmitters and a 3-tap relay delay



(a) CSI difference (b) RSS ratio (c) CSI 1-bit secret match rate (d) CSI 2-bit secret match rate

Figure 7: Recovery results with different delay times at relays and a 0 dB received power ratio from relays vs. transmitters

Figures 6c-6d and 7c-7d 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.

Recovered Channels for Activity Classification. E-eyes is a location-oriented activity classification system utilizing continuously collected channels from multiple devices around a home. It infers location information by checking traces from each device to narrow down the range of possible activities. For our evaluation, we only consider one location and make the number of activity types per location comparable to E-eyes' evaluations. 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 nodes are placed 2 m away from each other and volunteers repeat the activities 1-3 m away from the nodes.

We assume that the malicious relay disturbs E-eye's sensing by randomly changing its amplification factor. The extra fluctuations can affect sensing results in two ways: first, in-place activities with less CSI variation can be taken as walking with greater variation; second, changes in CSI distributions can lead to misclassifications among in-place activities.

First, we choose empty room traces as in-place activity examples and make E-eyes confuse them 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 1. The

Table 1: Average normalized cumulative moving variance of different types of traces

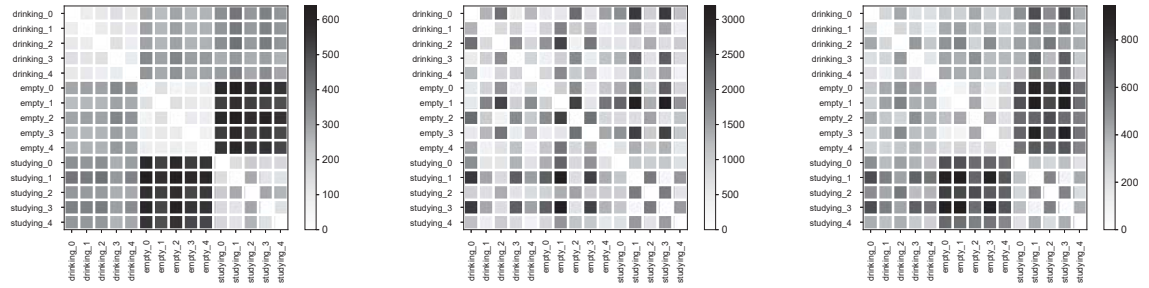
Trace type	Normalized cumulative moving variance
Walking	0.0329
Empty room - mixed	0.0285
Empty room - ground truth	0.0104
Empty room - recovered	0.0127

mixed traces are very likely to be taken as walking traces because of their greater variance, but the recovery module is capable to bring the metric back to a level close to the ground truth.

We further change the received power ratio by a random value between -0.5 and 0.5 dB every packet to simulate the small fluctuations caused by in-place activities. It can be seen from figure 8 that the EMDs of mixed traces show a random pattern. But after channel recovery, the EMD pattern is similar 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 accuracy improves from 33.3% to 100%.

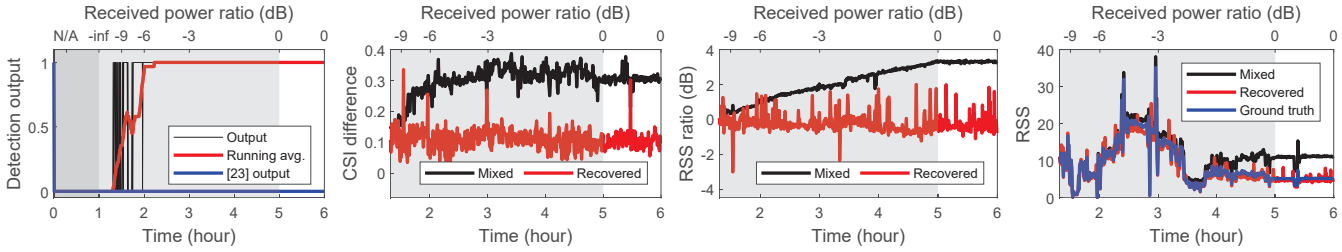
6.4 System Test

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



(a) Ground truth (b) Mixed traces (c) Recovered traces

Figure 8: EMDs of indoor trace pairs. EMD calculates the minimal cost to transform one distribution into the other. Smaller EMDs indicate more similar distributions.



(a) Detection output (b) CSI difference (c) RSS ratio (d) RSS value

Figure 9: Detection and recovery results of the system test. The dark grey, light grey, and white sections represent the phases of no injection, gradual injection, and stable injection, respectively. The recovery results (b)-(d) are plotted with traces after the detection module first reports a relay.

Table 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 [30]	36.25%	0.2971	-2.3567 dB	49.81%	49.92%
	OptML [4]	54.38%	0.2902	4.1404 dB	52.76%	51.63%
	mD-Track [33]	26.25%	0.2833	-6.2592 dB	49.80%	51.13%
	RelayShield	95.63%	0.1457	-0.6090 dB	77.97%	71.72%

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 [23] are shown in figure 9a. We tune parameters for [23] 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 [23] does not report any detection since the injection begins. The recovery results are shown in figures 9b-9d. 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.

6.5 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. To answer this question, we resolve multipath components with the channel-to-signal-path methods in R2F2 [30], OptML [4], and mD-Track [33], 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 (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 (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 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 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.

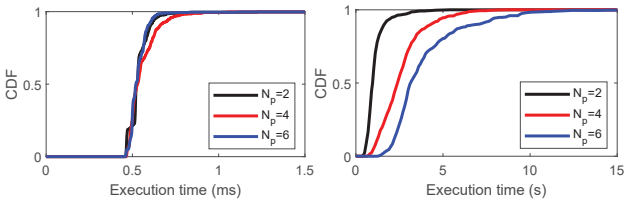
6.6 Runtime

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 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.

7 DISCUSSIONS

7.1 Simulated Channels as Training Datasets

RelayShield uses simulated channels to train neural network models for relay detection and channel recovery modules, as real-world channels are expensive to collect and may not cover all possible scenarios. Although using real-world channels could potentially improve system performance, it requires the user to build an attack prototype and use devices with a GHz-level sampling rate



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

Figure 10: Runtime distributions of the relay detection and channel recovery modules of 500 input channels

to separate signal paths. Furthermore, 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 at runtime. In contrast, generating simulated channels takes significantly less effort and covers all possible cases in environments with similar features. Thus, using simulated channels as training datasets is a viable approach.

7.2 Optimizing System Parameters for Different Environments

To ensure accurate channel recovery results, appropriate values for system parameters, such as signal path numbers and traveling distances, should be selected based on the environment. A quick way to determine suitable values for these parameters is to collect several channels from different locations, resolve their parameters with the problem defined in equation (7), and observe the resulting ranges. We recommend starting with larger ranges and gradually decreasing them until the appropriate values are determined.

7.3 RelayShield Limitations

Based on existing attack works [24, 29], 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. However, if the attacker set various amplification factors for different subcarriers, RelayShield might fail to detect attackers and recover the channels.

8 CONCLUSION

We propose 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, RelayShieldis 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.

ACKNOWLEDGMENTS

We would like to thank our anonymous shepherd and reviewers for their insightful comments and constructive suggestions. This work is supported by NSF ECCS-2128567 and CNS-2112471 awards.

REFERENCES

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

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

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

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

[5] Lars Baumgärtner, Alexandra Dmitrienko, Bernd Freisleben, Alexander Gruler, Jonas Höchst, Joshua Kühlberg, Mira Mezini, Richard Mitev, Markus Miettinen, Anel Muhamedagic, et al. 2020. 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)*. IEEE, 458–467.

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

[7] Stefan Brands and David Chaum. 1994. 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*. Springer, 344–359.

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

[9] Murat Demirbas and Youngwhan Song. 2006. 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*. 565–570.

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

[11] Saar Drimer, Steven J Murdoch, et al. 2007. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks.. In *USENIX security symposium*, Vol. 312.

[12] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. 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.

[13] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. 2010. 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*. Springer, 35–49.

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

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

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

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

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

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

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

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

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

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

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

[25] Kasper Bonne Rasmussen and Srdjan Capkun. 2010. Realization of RF Distance Bounding.. In *USENIX security symposium*. 389–402.

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

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

[28] Paul Staat, Kai Jansen, Christian Zenger, Harald Elders-Boll, and Christof Paar. 2022. Analog Physical-Layer Relay Attacks with Application to Bluetooth and Phase-Based Ranging. *arXiv preprint arXiv:2202.06554* (2022).

[29] Yu-Chih Tung, Kang G Shin, and Kyu-Han Kim. 2016. 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*. 331–340.

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

[31] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. 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*. 617–628.

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

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

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