How to Get Away with MoRTr: MIMO Beam Altering for Radio Window Privacy

Syed Ayaz Mahmud
School of Computing
University of Utah

Neal Patwari
McKelvey School of Engineering
Washington University in St. Louis

Sneha K. Kasera
School of Computing
University of Utah

***Abstract*—We consider the *radio window attack*, a privacy threat in which an attacker monitors the wireless link over a period of time, recording the channel state information (CSI) across multiple packets and uses a model to detect, estimate, or classify human movements. To prevent such privacy attacks on a wireless channel, we propose *modifying radio training (MoRTr)*, a novel system for Wi-Fi MIMO-OFDM devices that alters transmitted symbols over time, space and frequency via a pseudo-random process that mimics the changes due to human activity, particularly the training symbols that are used to measure the wireless channel by the receiver. We perform extensive experiments to demonstrate that an attacker is thwarted by the approach. At the same time, we demonstrate that any receiver is able to use its measured CSI to demodulate the data without any significant degradation in performance, despite the fact that the receiver is not measuring the true CSI.**

I. Introduction

Modern wireless networks provide for the privacy and integrity of transferred data over the wireless channel by means of cryptography and encryption algorithms in the upper layers of communication protocols. However in today's networks, the privacy and security can be breached at the physical layer by monitoring radio windows [1] in which an eavesdropper can observe activities from behind walls. The radio waves from any transmitter, for example, a Wi-Fi device in a home, are altered as they propagate near moving objects and bodies present in its vicinity. The characteristic of these changes can be measured from a distance as radio waves penetrate non-metal walls and objects. An adversary, without physical access to a private area, can use a receiver outside its perimeter or walls to measure the changes in the signal from the transmitter to learn about the activities or locations of people present in that environment. Such measurements have been shown to be capable of human presence detection [2], fall detection [3], motion detection [4], activity and gesture recognition [5]–[7], keystroke detection [8]. These attacks can go undetected as they do not require an attacker's device to transmit. Thus, *data* security is no barrier to a radio window attacker since radio signals, even from those networks secure against data eavesdropping, still penetrate walls and can be measured. Furthermore, RF sensing can be performed with widely available commercial off-the-shelf (COTS) devices [9]. Both received power and channel state information (CSI) measurements can be accessed via firmware or software. In this paper, we focus on CSI measurements because these are an integral part of multicarrier multiple-input multiple-output (MIMO) protocols, used to achieve high data rates in multipath channels.

Channel measurements reveal significant short and long term variations due to human activity, while normal variations without human activity are generally minimal. In a radio window attack, an attacker monitors the wireless link over a period of time, recording the CSI across multiple packets and recording temporal, spatial, frequency or wavelet-domain characteristics, and use a model to detect, estimate, or classify the human movements that may (or may not) have occurred.

To prevent such privacy attacks on a wireless channel, we propose *modifying radio training (MoRTr)*, a novel system for Wi-Fi MIMO devices that alters transmitted symbols, particularly the training symbols that are used to measure the wireless channel by the receiver. In MoRTr, the symbols are modified over time, space and frequency via a pseudo-random process that mimics the changes due to human activity by changing the amplitude and phase of each antenna's signal. Since the statistics of the artificial changes across multiple dimensions match those from actual human-caused changes, and the signal characteristics are distinct at different receiver antennas, a passive attacker is unable to distinguish the two even if they know the protocol, therefore degrading their ability to detect actual human activity.

One of the key challenges in designing MoRTr is being able to mimic real human gestures well enough to fool an eavesdropper. To do this, a transmitter must know how such gestures effect the received signal. Moreover, it must be able to mimic multiple gestures in an unpredictable order. The transmitter must know how to randomly modify the signals characteristics and match the spatio-temporal statistics of such gestures, i.e., it must have a prior knowledge of the statistical model of how a transmitted RF signal would change in amplitude or phase across multiple antennas while an action is performed by a human, and be able to generate signals that vary randomly with this model. MoRTr generates such a statistical model for a human activity from a set of real CSI measurements.

MoRTr protects the privacy of a person in an indoor environment from an eavesdropper sensing the wireless channel at the same time protecting the integrity of data to a legitimate receiver. We make the following contribution in this work:

1) We introduce a novel protocol, MoRTr, that modifies Wi-Fi transmit signals in such a way that the measured

CSI of the received signal mimics human activity in a way that thwarts a radio window attacker from obtaining reliable activity information from the signals.

2) We implement the Wi-Fi protocol IEEE 802.11n on a software defined radio (SDR) platform that can transmit, receive, and decode data packets and perform channel estimation. We implement MoRTr, in software, on this platform for experimental evaluation.
3) We evaluate MoRTr in a series of experiments that test 1) the performance of a receiver that demodulates data from a MoRTr transmission, and 2) the ability of a radio window attacker to perform gesture recognition, and in particular, distinguish between a real and a fake (MoRTr-generated) gesture. Our experimental results show that MoRTr is able to fool an eavesdropper with specific fake gestures, while genuine receivers are able to demodulate the signal and obtain the payload without any significant degradation in the bit error rate.

II. Background and Related Work

A. MIMO-OFDM and CSI

Commercial Wi-Fi devices, operating on 802.11n/ac standard, use multiple-input multiple-output (MIMO) and orthogonal frequency division multiplexing (OFDM) configuration to achieve a high data rate, increased reliability and a low bit error rate (BER). Generally, two diversity techniques are available for MIMO, *Spatial diversity*, and *Spatial multiplexing*. For obtaining spatial diversity, identical copies of data are sent over multiple transmit and receive antennas in parallel to improve reliability because it is unlikely that all paths will be degraded at the same time due to noise or interference. In contrast, spatial multiplexing divides the data into independent chunks and sends over multiple antennas in parallel to improve data rate. OFDM on the other hand divides the available 20MHz bandwidth into 64 sub-carriers thus, making the system bandwidth efficient. Each of these sub-carrier can be extracted at the receiver to estimate channel frequency response in the form of channel state information (CSI) allowing the channel sensing to be more accurate. The receiver uses the CSI measurement to equalize the channel effect and subsequently, for data demodulation. The CSI is a matrix with size $[M \times N \times K]$, where M is the number of transmit antennas, N is the number of receive antennas and K is the number of sub-carriers used in the OFDM modulation. Letting the transmitted signal vector on subcarrier k be $\mathbf{x}(k) = [x_0(k), x_1(k)]^T$, and the received signal vector on subcarrier k be $\mathbf{y}(k) = [y_0(k), y_1(k)]^T$,

$$\mathbf{y}(k) = H(k)\mathbf{x}(k) + \mathbf{z} \tag{1}$$

where $\mathbf{z}$ is additive noise and $H(k)$ is the channel response for frequency subcarrier k. While we consider a 2x2 MIMO channel, as shown in Figure 1, the channel matrix with coefficient between each antenna pair can be represented as

$$H(k) = \begin{bmatrix} h_{0,0}(k) & h_{0,1}(k) \\ h_{1,0}(k) & h_{1,1}(k) \end{bmatrix}.$$

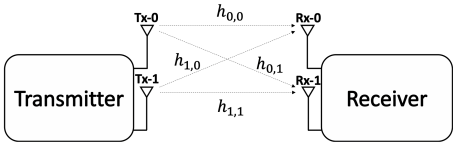


Fig. 1. Channel matrix coefficients for 2x2 MIMO

The changes in CSI over time captures information about the motion of objects and people, as each subcarrier's state is impacted by multipath. A complex-valued non-stationary channel can be represented by

$$h(k,t) = \sum_{i=0}^{N-1} a_i(t) e^{-j2\pi f_k \tau_i(t)}, \tag{2}$$

where $a_i(t)$ is the amplitude gain and $\tau_i(t)$ is the propagation delay at time t, and f_k is the center frequency of the kth subcarrier [10]. The amplitude $|h(k,t)|$ and phase $\angle h(k,t)$ changes with time t whenever the transmitter or receiver move, or when objects and people present in the environment move.

Open-source software has been developed to obtain CSI from commercial routers and laptops with certain types of network interface cards (NICs) [9], [11]. Several existing works extract CSI from these NICs and use these for wireless sensing applications including device free human activity recognition [12], for keyboard stroke behaviour recognition (e.g., WiKey [13]), for occupant activity recognition [14], and for monitoring, logging and taking necessary actions during sleeping (e.g., sleep guardian [15]). CSI is susceptible to eavesdropping by third parties in the coverage area of a transmitter and can be used by them to detect the motion, gestures or location of a human present in between the transmitter and receiver nodes while the victim is oblivious to this information leakage. Figure. 2 depicts such an attack.

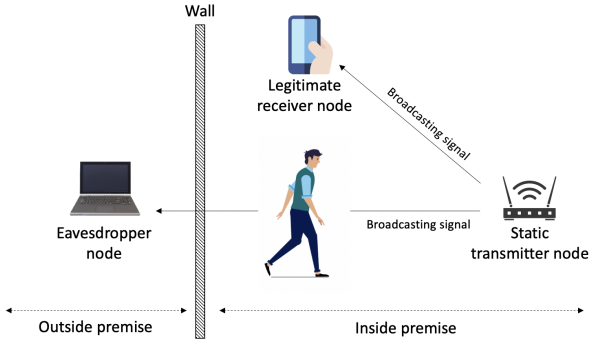


Fig. 2. Attack on privacy using channel state information (CSI).

B. Related work

Most of the indoor wireless nodes are equipped with omni-directional antennas which radiate radio signal in all directions and are prone to eavesdropping. One approach for defending against this eavesdropping is to use a directional antenna such that the signals can be directed to the intended receiver and

thereby reducing the possibility of the eavesdropper obtaining the signal [16]. While Wi-Fi standards like 802.11n/ac have adopted MIMO and are capable of beamforming, these require a handshake phase between the transmit and receive nodes. Therefore, if any party is not compatible with the beamforming protocol, the eavesdropper will be able to receive the signal. Furthermore, even when transmit beamforming is in place, an attacker may re-position its node to find the line of sight path to receive a higher signal to noise ratio.

Another approach to defend against an eavesdropper is to add artificial noise [17] where an authorized transmitter generates a formulated interfering signal in such a way that the eavesdropping channel is degraded while the legitimate channel is intact. The main difference between such an idea and our work is that we intend to fool the eavesdropper with patterns of real human gesture and not with random noise. Researchers have developed counter measure techniques [18] where the signal to noise ratio at the attacker is significantly degraded. Similarly, WiGuard defeats the attacker by causing interference with the attacker's node resulting in a distorted CSI thereby reducing the rate of successful attacks [19]. In another related work, an interference-negligible RF sensing shield [20] preserves authorized RF sensing and incapacitates eavesdroppers by using external hardware to distort the transmitter's signal. In the same vain, [21] uses an additional supporting node called the *forwarded* that receives the original transmitted signal, modifies it properties, and forwards the modified signal back to the wireless channel. An attacker node receives both the original and the modified signal resulting in a distorted CSI measurement.

To the best of our knowledge, there is no prior work on modification of training symbols of a wireless transmitter for defending against the radio window attack. In this paper, we perform extensive experiments to demonstrate that such modification on the transmitter deceive the attacker but do not significantly influence the performance of data extraction and demodulation at the genuine receiver.

III. Threat model

We consider the scenario where an attacker tries to capture a gesture performed by a targeted person from CSI time series data in a MIMO wireless channel.

- The attacker can position its wireless device in the vicinity of a stationary legitimate transmitter node in an indoor environment or behind a wall where it cannot be discovered easily.
- The attacker's node is equipped with tools for CSI extraction and has the capability to post process the data for noise reduction and obtain meaningful CSI measurement using machine learning.
- The attacker acts as a passive device, and does not have any access to the targeted node physically or remotely to tamper with the transmitted signals.
- There is only one eavesdropping node in the vicinity of transmitter, and number of antennas at both transmitter and eavesdropper are equal. The challenge posed by multiple attack nodes is beyond the scope of this paper.

IV. RF Sensing in MIMO-OFDM

In this section, we describe how any receiver (including an eavesdropper) receives and processes the broadcast signal from a MIMO-OFDM transmitter for channel estimation and payload demodulation. A receiver must use the protocol's known training symbols or preamble in order to perform several tasks required for reliable communication and RF sensing. In any MIMO-OFDM protocol, the training symbols are predefined and available to both transmitter and receiver. These unmodulated symbols are used by the receiver to detect the start of packet, synchronize to the symbol timing, correct frequency offset, and measure the change in the transmitted signal due to the multipath channel. This last task, known as *channel estimation*, is required to counteract the channel's effects and demodulate the payload portion of the packet. In our threat model, an eavesdropper only cares about the preamble section of the packet as its goal is to measure the channel for human gesture recognition, while a legitimate receiver is primarily concerned with the demodulation of the payload. In this section, we first describe an example frame structure. Next, we explain how the channel is estimated from the received preamble in order to mathematically show how both the eavesdropper and the legitimate receiver perform channel estimation. Finally, we describe the impact of the channel estimate on payload demodulation.

A. Frame structure

We use the IEEE 802.11n standard as an example of the OFDM-MIMO frame structure. This WiFi standard has three different preamble formats designed to operate across the a/b/g/n standards [22]. Of these three preamble formats, we use the *greenfield* mode in our experimental work. We note that the methods developed in this paper can also be applied to other modes with minor modifications. The structure of the greenfield frame, and receiver tasks performed with each subfield of this frame are described as follows:

- *High-throughput short training field (HTSTF)*: Symbols in this field are defined in the 802.11n standard. In time domain, after appending a cyclic prefix (CP), the HTSTF is a 10x repetition of the identical sixteen samples, which then is used for frame detection and synchronization.
- *High-throughput long training field (HTLTF)*: These symbols are used for frequency offset estimation, correction, and channel estimation. For each transmit antenna, there is one HTLTF, which then allows CSI estimation across a TX / RX pair. In our experiments, we use two transmit antennas, therefore, we have two HTLTFs.
- *High-throughput signaling fields 1 and 2 (HTSIG1 and HTSIG2)*: These fields contain information about the modulation and coding scheme, packet length, and the type of forward error coding used. In our tests, we exclude the signaling fields, as our goal is not to replicate the exact protocol but to be able to estimate CSI in a

similar manner and then apply modification. We assume that the receiver has all the necessary information to decode the packet.

- *Payload*: These fields carry the data, and in addition, use 4 subcarriers for pilot signals. Other subcarriers are modulated with QPSK, QAM-16 or QAM-64. The pilot symbols are used for phase correction after the channel is equalized.

Note that while the HTSTF and HTSIG fields are identically sent from each TX antenna, the HTLTF field signals are different at each TX antenna and sent in an orthogonal pattern. Let $L(k)$ be the HTLTF symbol value as a function of subcarrier k, and let $\mathbf{x}_s(k)$ be the $M \times 1$ vector transmitted during HTLTF-s. Consider the $M \times M$ matrix $X(k) = [\mathbf{x}_0(k), \ldots, \mathbf{x}_{M-1}(k)]$ which describes the transmitted signal across antennas (rows) and across HTLTF fields (columns). The 802.11n protocol defines an orthogonal matrix P such that

$$X(k) = L(k)P, \tag{3}$$

where $L(k)$ is a scalar multiplying each element of P. For example, when there are $M = 2$ transmit antennas,

$$P = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{4}$$

In other words, $L(k)$ is sent equally on each transmit antenna during HTLTF-1, but sent with orthogonal linear combinations on the M antennas in later HTLTFs. In general for OFDM-MIMO channel estimation, the transceivers must know the structure of these fields in frequency and the spatial domains in order to estimate the CSI as described next.

B. CSI Estimation

Channel state information must compute the channel gain for each antenna pair, which we refer to as the spatial dimension, and across subcarriers, which we refer to as the frequency dimension k. The MIMO channel estimates are computed with the received HTLTFs $\mathbf{y}(k)$ in each HTLTF field s from $s \in \{0, \ldots, N-1\}$. In a MIMO system, each receive antenna measures a linear combination of the transmit signal as described in (1). Using (3), we have

$$Y(k) = L(k)H(k)P + \mathbf{z}, \tag{5}$$

where $Y(k) = [\mathbf{y}_0(k), \ldots \mathbf{y}_{N-1}(k)]$, and $\mathbf{y}_s(k)$ as the N-length vector of received signal value during HTLTF-s on subcarrier k. Following symbol synchronization and frequency offset correction, the receiver receives the HTLTF fields. It removes the cyclic prefix and performs the FFT to extract the values $Y(k)$ for each subcarrier. Given that $P^{-1} = \frac{1}{M}P^T$, we can estimate $H(k)$ (for any k that $L(k) \neq 0$):

$$\hat{H}(k) = \frac{1}{ML(k)}Y(k)P^T \tag{6}$$

E.g., for $M = 2$, the top left element of $\hat{H}(k)$ is

$$\hat{h}_{0,0}(k) = \frac{1}{2L(k)}(y_{0,0}(k) - y_{0,1}(k)). \tag{7}$$

C. Channel equalization

We describe how, in fields other than the HTLTF, the receiver equalizes the channel to estimate the transmitted symbols from its measurements $\mathbf{y}(k)$ as given in (1). To recover the transmitted symbols $\mathbf{x}(k)$, the receiver must invert the channel matrix, using channel equalization. As we already calculate a channel estimate $\hat{H}(k)$ in (6), our next step is to invert the response of the channel to recover the transmitted data $\mathbf{x}(k)$. One computationally efficient method of equalization is the zero forcing (ZF) method. The ZF method is a linear equalizer that corresponds to minimizing the inter-symbol interference.

To estimate $\hat{\mathbf{x}}(k)$, we need to find a matrix $W(k)$ that will invert $\hat{H}(k)$, that is, $W(k)\hat{H}(k)$ is the identity matrix. The ZF method uses the pseudo inverse,

$$W(k) = (H^H(k)H(k))^{-1}H^H(k), \tag{8}$$

where superscript H is used to denote the Hermitian matrix. The transmit signal estimate $\mathbf{x}(k)$ can then be obtained from

$$\hat{\mathbf{x}}(k) = W(k)\mathbf{y}(k). \tag{9}$$

Note that ZF is not optimal in terms of noise reduction. While inverting the channel, the ZF method may unduly amplify the noise at frequencies where the rank of channel response is low. To overcome this problem, other equalization algorithms can be used to improve performance at the expense of computational complexity. Our MoRTr method is not dependent on any equalization method. Using a more robust $W(k)$ will improve the payload demodulation performance of both the eavesdropper and desired receiver. In our implementations, we use a ZF equalizer for testing and comparing the performance of payload demodulation at the legitimate receiver before and after our proposed modification is applied; better equalization methods would equally improve both cases. The same channel equalization method is applicable even if we modify the transmit signal as discussed in next section.

D. Payload demodulation

Once the channel equalization is complete, a legitimate receiver is ready to extract the remaining part of the frame, the payload. The payload, in addition to the data symbols, contains pilot symbols which are used by the receiver for phase estimation and correction of data symbols.

The final step is to demodulate the data. The receiver applies (9) to estimate the transmitted symbol values $\mathbf{x}(k)$ for each transmit antenna and subcarrier k. These complex values are then used in demodulation. In our experiment, we implement QPSK demodulation on each subcarrier and stream as a proof of concept quantification of MoRTr's minimal impact on demodulation performance. However, we note that the modulation is typically chosen adaptively based on the available SNR and will have little effect on MoRTr.

E. Channel estimation to measure a human activity

With the procedure described in Section IV-B, an attacker estimates the channel coefficients for all (or any) subcarrier(s) to understand how CSI varies over time. We use $\hat{H}(k, t)$ to refer to the channel estimate on subcarrier k for packet t. The

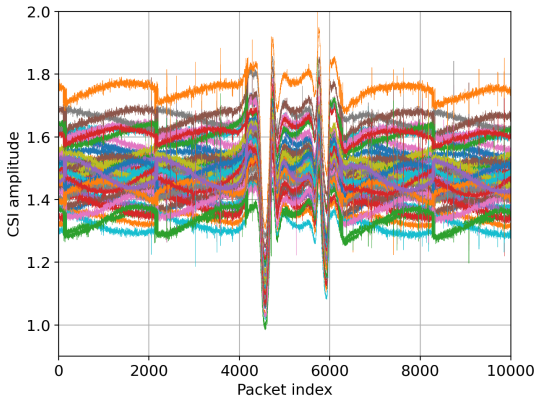


Fig. 3. Estimated CSI amplitude over time for all subcarriers (each plotted with a different color) while a person moves their hand twice vertically.

attacker observes CSI for packets $t \in \{0, \ldots, T-1\}$. The variations in $\hat{H}(k,t)$ over time are due to noise, interference, measurement error, and any gesture or activity performed in the vicinity of the transceivers. For RF sensing, $\hat{H}(k,t)$ is filtered over all subcarrier and over all packet t to reduce the noise and error, and to determine the changes in the CSI from its long-term or static environment values [8].

Different activities lead to distinct changes in CSI amplitude and phase [23] that are correlated over space, frequency, and time. We model these changes in the channel as follows.

$$H(k,t) = \bar{H}(k,t)Q(k,t), \tag{10}$$

where $\bar{H}(k,t)$ is the static channel (without any activity) and $Q(k,t)$ is the multiplicative activity impact function. This multiplicative activity impact function allows us to model variations in both phase and amplitude of the signal.

We show an example of our measurements in Figure 3, where the CSI amplitude is plotted for all subcarriers, for each packet received over the time when a hand gesture is performed. Here, two deep fades are observed, first when the person moves their hand in an upward direction, and second, when they bring it back down. A CSI-based activity recognition method first collects a large amount of labelled CSI data to train a classifier. Once trained, common human gestures can be recognized from eavesdropped measured CSI data. In our work, We implement fake gesture patterns to confuse an eavesdropper.

F. CSI Denoising

As discussed in Section IV-B, the estimated raw CSI is extremely noisy. A receiver or an eavesdropping node measuring CSI must *denoise* the signal first to extract the information that it requires to detect a specific gesture. There are various methods to remove noise from the estimated CSI. These include moving average [24], median filter [25], principal component analysis (PCA) [26], wavelet filter [27], Hampel filter [28] and many more. We follow the two steps described in [8] to remove the noise. First, we use a Butterworth low pass filter (B-LPF) to eliminate high frequency noise and then, use PCA to reduce the dimensions of the CSI and capture the most important components.

V. Modification of Radio Training

Wireless channels are broadcast media by nature. We acknowledge the fact that an eavesdropper can record a transmitted MIMO-OFDM signal. Furthermore, we acknowledge the need for sending training symbols from a transmitter so that the desired receiver can estimate the CSI and thus successfully perform demodulation. These two facts make it possible for an adversary, even without authorization, to estimate and monitor the CSI for activity. In this section, we describe how we modify the transmit signal to falsify the CSI measured by an adversary while preserving the ability of the desired receiver to demodulate the payload.

The intuition behind MoRTr is that any measurement of the received training symbol is a convolution of the transmitted training symbol and the channel. By modifying the training symbol at the transmitter, we impose changes in the CSI any receiver will measure. Furthermore, if we vary these changes randomly over time in a way unknown to the eavesdropper they would appear to them to be changes due to the channel. If these changes are indistinguishable from the complex-valued spatial, temporal, and frequency-domain statistics of changes caused by actual human activity, the eavesdropper would not be able to distinguish a faked channel variation from a real one. The introduced ambiguity thus makes the detection and classification of genuine human activity unreliable.

We consider a legitimate transmitter, a legitimate receiver, and an eavesdropper node, with M, N and E number of antennas, respectively. Let H^{tr} be the channel between the legitimate transmitter and the intended legitimate receiver and H^{te} be the channel between transmitter and eavesdropper. The transmit signal vector for each antenna can be represented as $\mathbf{x}(k) = [x_0(k), \ldots, x_{M-1}(k)]^T$. Then, the received signal at the eavesdropper $\mathbf{y}_e(k) = H^{te}\mathbf{x}(k)$ and the desired received signal at the legitimate receiver $\mathbf{y}_r = H^{tr}\mathbf{x}(k)$ are a function of the choice of $\mathbf{x}(k)$ at the transmitter. We exploit this modification of $\mathbf{x}(k)$ over time to maximize the confusion at the eavesdropper.

While measuring the CSI of a true gesture, we observe that it affects both the amplitude and the phase of the received signal due to constructive and destructive interference. A temporal change is observed on each frequency subcarrier component with slightly different channel frequency responses on each spatial stream of TX-RX antenna pairs. Therefore, to create a fake gesture, we modify the transmit symbols in the temporal, frequency, and spatial domains. We need a pre-defined statistical model for a normal gesture; with that model we can generate a random change that would simulate a gesture.

A. Building a Statistical Model

To create a statistical model for a gesture, MoRTr assumes that measurements of CSI while people perform various actions and/or gestures, G, are available a priori. MoRTr fakes

TABLE I
SYMBOLS USED IN EQUATIONS AND THEIR DESCRIPTION

Symbols	Description	Symbols	Description
M	Number of transmit antennas	N	Number of receive antennas
K	Number of subcarriers	$\mathbf{y}(k)$	Received signal vector on subcarrier k
$\mathbf{x}(k)$	Transmit signal vector on subcarrier k	$\mathbf{z}$	Additive noise
$h_{M,N}(k)$	Channel coefficient between Mth transmit and Nth receive antenna on subcarrier k	$H(k)$	Channel response matrix on subcarrier k
t	Packet index	a	Amplitude gain
τ	Propagation delay	f_k	Center frequency
$L(k)$	HTLTF symbol on subcarrier k	P	Orthogonal mapping matrix
s	Number of HTLTF	$W(k)$	Pseudo inverse matrix
$Q(k,t)$	Multiplicative activity impact function	g_t	Data for all subcarriers due to gesture for packet t
b^r	Data for all subcarriers and for all packets for a single gesture	R	Total number of experiments for a single gesture
T	Total number of packets	$\mu_{\mathbf{r}}$	Sample mean
C_r	Co-variance matrix	U	Left eigenvector
S	Diagonal matrix of eigenvalues	J	Fake gesture pattern
d	Vector of v independent zero mean Gaussian random variable	$\Delta(k)$	Modified transmit symbol on subcarrier k

these gestures to fool an eavesdropper. The measurements of CSI corresponding to these gestures can be done via offline experimentation prior to deployment of any network using MoRTr, or these can be done online for a particular deployed network in the environment of interest. All these measurements are then stored in a database, and we assume the legitimate transmitter has access to it over some secure channel that the eavesdropper cannot access.

In our work, we record fifty CSI measurements for each of four real gestures: **hand movement** in the vertical direction, action of a **punch**, **pick an object** up from the ground, and action of **push** between the transceiver nodes. Each fifty of these unique gestures present similar but slightly different patterns of variation of the measured CSI, as we now show.

To estimate the multiplicative effect of a gesture on the wireless channel, during our experiments, we record CSI measurements on a link starting from before the start of the gesture through the end of the gesture, $H(k,t)$, over frequency subcarrier k and time t, where H is the $M \times N$ channel matrix. Denoting the CSI before the gesture starts as $\tilde{H}$, we estimate the multiplicative effect of the gesture on the channel as,

$$\hat{Q}(k,t) = \left[H^H(k,t)H(k,t)\right]^{-1} H^H(k,t)\tilde{H}(k,t). \quad (11)$$

The matrix $\hat{Q}(k,t)$ is an $M \times N$ matrix like H but it estimates the multiplicative change in the channel due to the gesture performed at time t on subcarrier k. We also define $\mathbf{q}(k)$ as a $MN \times 1$ vector in which we linearize the MN elements of $\hat{Q}(k)$ into a column vector. Next, we stack these channel vectors to consolidate the channels measured at different frequencies k and samples t:

$$\mathbf{g}_t = \begin{bmatrix} \mathbf{q}(0) \\ \vdots \\ \mathbf{q}(K-1) \end{bmatrix}, \quad \mathbf{b}^r = \begin{bmatrix} \mathbf{g}_0 \\ \vdots \\ \mathbf{g}_{T-1} \end{bmatrix}, \quad (12)$$

where $\mathbf{g}_t$ is the data for all subcarriers due to the gesture at time sample t, and $\mathbf{b}^r$ is the data for all time $t = 0, \ldots T-1$, the entire duration of the gesture. Thus, $\mathbf{b}^r$ contains measurements for a single gesture experiment r, with CSI across space, frequency, and time.

Now, we must build a statistical model of this gesture $\mathbf{b}^r$ over R repeated experiments (i.e., $r = 0, \ldots, R-1$), so that we can later generate a random simulation of the channel changes in a way that accurately represents this gesture. Any failure to model the true correlation structure across multiple experiments of the same gesture will give the eavesdropper the chance to distinguish between the fake and the true gesture.

Using the data recordings, we calculate the sample mean $\mu_{\mathbf{r}} = \frac{1}{R}\sum_{r=0}^{R-1} \mathbf{b}^r$, and the sample covariance matrix, $C_{\mathbf{r}}$:

$$C_{\mathbf{r}} = \frac{1}{R-1}\sum_{r=0}^{R-1}(\mathbf{b}^r - \mu_{\mathbf{r}})(\mathbf{b}^r - \mu_{\mathbf{r}})^H \quad (13)$$

To decompose the covariance matrix, we use singular value decomposition (SVD),

$$C_r = \tilde{U}S\tilde{U}^H, \quad (14)$$

where $\tilde{U}$ is the left eigenvector matrix and $S = \text{diag}(\lambda_0^2, \ldots, \lambda_{MNTK-1}^2)$ is matrix of eigenvalues. Note that $C_{\mathbf{r}}$ has at most $v = \min(MNTK, R)$ non-zero eigenvalues. Typically, we expect $R < MNTK$ since the product $MNTK$ will be large. Therefore, $\tilde{U}$ is a $MNTK \times v$ matrix of the eigenvectors with positive eigenvalues. We store $\tilde{U}$ in the transmitter for later generation of a fake gesture.

B. Online Generation of a Fake Gesture

To create a fake gesture we generate $\mathbf{d}$, a vector of v independent zero mean Gaussian random variables, where the ith element has λ_i variance. Then, we generate data for a fake gesture as:

$$J = \tilde{U}\mathbf{d} + \mu_r. \quad (15)$$

J represents the pattern that matches the spatial, temporal, and frequency variations of an experimentally recorded gesture. We then reshape the vector J in the reverse manner as in equation (12), to create TK different matrices $J(t,k)$, essentially, one $M \times N$ matrix for each subcarrier k and time sample t.

The matrices $\{J(t,k)\}_k$ are used at time t to alter the original transmit signal vectors $\{\mathbf{x}(k)\}_k$, that is, to alter the complex-valued signals on the M transmit antennas being sent to convey the data in the packet sent at time t. In MoRTr, instead of $\{\mathbf{x}(k)\}_k$, the transmitter sends a modified transmit symbol, which we denote $\Delta(k) = [\Delta_0(k), \ldots, \Delta_{M-1}(k)]^T$. $\Delta(k)$ is an M length vector containing the complex amplitude sent on each antenna,

$$\Delta(k) = J(t,k)\mathbf{x}(k). \tag{16}$$

Once the eavesdropper receives the modified symbol, it measures the received vector of $H^{te}J(t,k)\mathbf{x}(k)$ rather than $H^{te}\mathbf{x}(k)$. While $\mathbf{x}(k)$ is known during the training symbols, the eavesdropper estimates $H^{te}J(t,k)$ as the CSI; there is no way for the eavesdropper to recover H^{te} itself. Note that there is no way for the legitimate receiver to estimate the true CSI either. However, during equalization, eavesdropper (and legitimate receiver) can estimate the pseudo-inverse of $H^{te}J(t,k)$ (and $H^{tr}J(t,k)$) and thus obtain the sent data. Even when the eavesdropper obtains the sent data, it is unable to distinguish between a fake and a real gesture.

VI. Hardware and Software Implementation

We implement MoRTr in order to quantify its performance in real-world situations. We need to demonstrate that the modification is transparent to the desired receiver, as well as that an eavesdropper will be unable to distinguish a fake gesture created with MoRTr from a genuine human gesture. To be as realistic as possible, we implement MoRTr to run on top of IEEE 802.11n. We choose 802.11n because it is certainly a common MIMO-OFDM standard in use, and additionally, it is the basis for the PHY layer in 802.11ac and 802.11x. There are no fundamental changes that would be needed to implement MoRTr with higher M, N possible in these newer generations of 802.11.

We implement MoRTr using Python on a laptop connected to a universal software radio peripheral (USRP) National Instruments / Ettus Research B-210 software-defined radio [29]. Each of the B210 is installed with two antennas. We experiment with one device as transmitter, one as receiver, and one as eavesdropper. MoRTr interacts with the USRP through hardware drivers (UHDs).

VII. Experimentation and Performance Analysis

In this section, we experimentally evaluate the performance of MoRTr using our implementation. Our goal is to understand the capability of our system in a real-world environment while an eavesdropper carries out a radio window attack. Therefore, to evaluate our system's performance, we focus on:

1) MoRTr's ability to deceive an eavesdropper attempting to reliably observe a gesture; and
2) The effect of the transmit symbol modification on a legitimate receiver's ability to demodulate a packet's payload.

We test MoRTr's ability to fake more than one gesture type. During each experiment, we use the offline method as described in V-A. We carry out the experiments in an indoor environment with the transmitter and the receiver node at a similar height, but 6 meters apart. We separately record the CSI at the eavesdropping node during multiple trials of the four real and four fake gestures. After applying CSI denoising, we plot the CSI magnitude. Figure 4(a) plots four real hand gesture CSI measurements over time. Figure 4(b) plots the spatio-temporal signals generated by MoRTr to simulate a hand gesture, and Figure 4(c) plots the measured CSI at the eavesdropper of a fake gesture over the four antenna pairs (for a single subcarrier). Although space limitations prevent us from plotting genuine and MoRTr-generated CSI streams for all subcarriers, gestures, and antenna pairs, we observe that MoRTr does capture the general time frequency characteristics of a true gesture. Figure 5 compares the real and fake gestures in the eavesdropper's measured CSI for a single channel for different types of gestures.

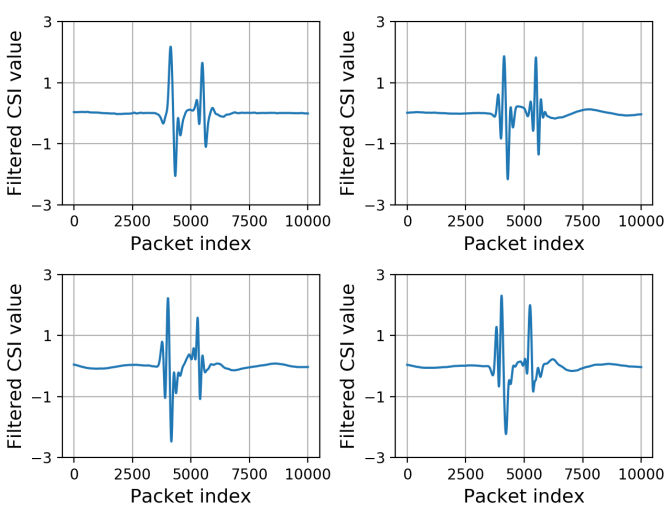
A. Distinction & Classification of Real and Fake Gestures

Once an eavesdropper node estimates the CSI with methods presented in Section IV, it can now attempt two things - first, distinguish between real and fake gestures; second, correctly classify the different human activities.

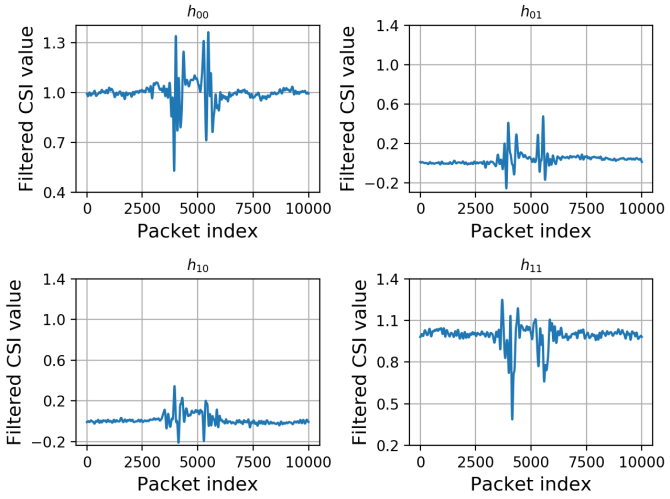
To evaluate the eavesdropper's ability to perform these two tasks, we assume that the attacker follows a supervised learning based approach to perform gesture recognition and classification. We also assume that the eavesdropper is aware of MoRTr and can generate fake CSI modifications just as the legitimate transmitter can. Learning algorithms in general can recognize activities by comparing a test set with a training set. There are a number of supervised learning algorithms that provide different levels of accuracy based on the quality of acquired signals. In our experiment, we use and compare four such algorithms, support vector machine (SVM), logistic regression (LR), decision tree (DT), and naive Bayes (NB).

To prepare for privacy attacks, the eavesdropper measures a large number of both real and fake gestures. The eavesdropper labels each gesture as fake or real and stores it in its training database. The eavesdropper uses this database to train its classifiers. We collect a test set for the eavesdropper and use it to train each learning classifier.

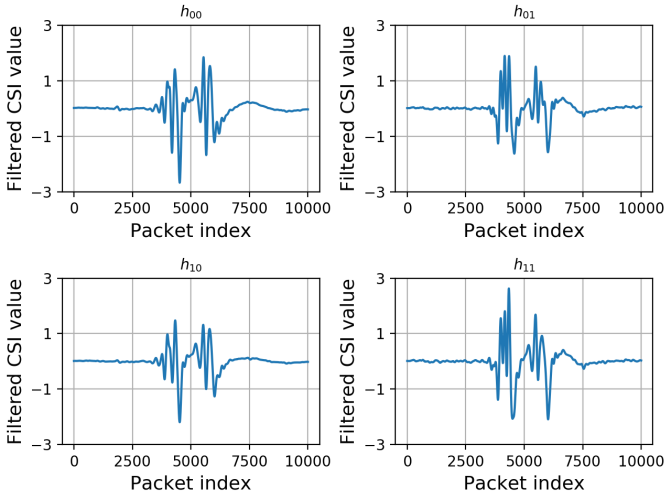
During the attack, the eavesdropper measures the CSI and applies its learning algorithm to distinguish between real and fake gestures of each possible type. Figure 6 shows the performance, with a confusion matrix, using the SVM algorithm, which has approximately the same performance as the other algorithms (LR, DT, and NB) we evaluate. It can be seen from the confusion matrix that an attacker is largely unable to distinguish between a real and fake human gesture, for any of the four gestures that we test. For example, a real punch is classified as either a fake or a real punch, equally likely, and is almost never confused with a different gesture and the activities are correctly classified with more than 80% accuracy.



(a)

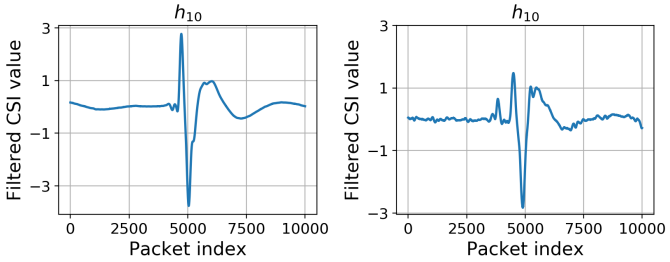


(b)

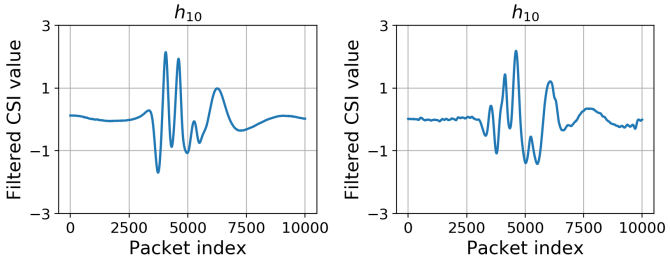


(c)

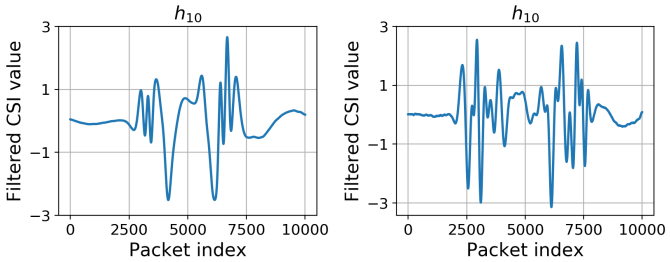
Fig. 4. (a) Measured CSI during real hand gestures, (b) MoRTr-generated transmit modification signals for fake hand gestures, (c) Eavesdropper-measured CSI during fake hand gestures.



(a)



(b)



(c)

Fig. 5. CSI measured at eavesdropper during real (left) and fake (right) for the gestures : (a) Punch, (b) Push and (c) Pickup

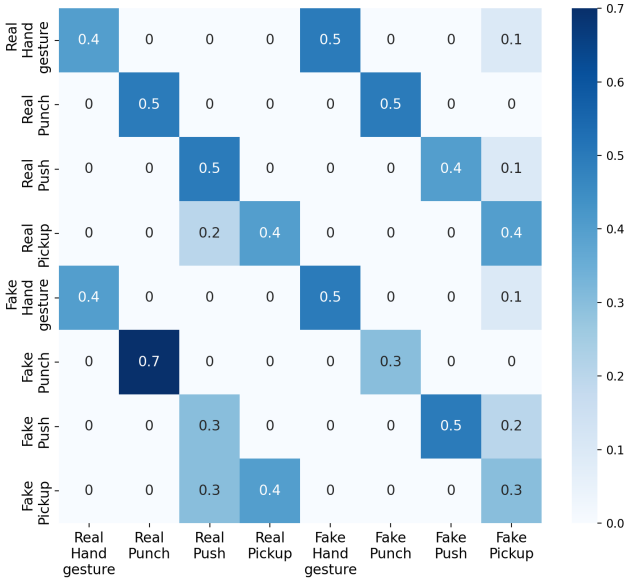


Fig. 6. Confusion matrix of different real and fake gestures based on support vector machine algorithm.

TABLE II
COMPARISON OF MEAN BER DURING PERFORMING REAL GESTURE AND WHILE MODIFICATION IS APPLIED

	Bit Error Rate		
	w/out MoRTr	**w/ MoRTr**	p**-value**
Hand Gesture	0.00048	0.00049	0.66
Punch	0.00039	0.00044	0.85
Push	0.00043	0.00046	0.76
Pickup	0.00042	0.00042	0.61

B. Demodulation Performance

To confirm that MoRTr does not degrade normal communication, we measure the bit error rate at the legitimate receiver both with and without MoRTr. We compare the received bits in error with each packet's total number of bits and compute the mean error rate over 10000 packets, around 500 bits each.

The bit error rate (BER) values presented in Table VII-B are high for both cases, with and without MoRTr, in comparison to generic wireless network performance. A wireless node operating using an 802.11 WiFi protocol incorporates a forward error correction (FEC) mechanism [30] which corrects any bits that are received in error and improves the overall performance. In our framework, we did not implement FEC, as our goal was not to improve the received bits accuracy but to fairly compare the performance when the symbols are modified. However, Table VII-B shows that no significant degradation is observed in the mean BER when comparing demodulation performance with and without MoRTr. We also compute the p-value with a null hypothesis that the two methods have identical BER vs. alternate hypothesis that the two are different. All p-values are greater than 0.05, indicating that there is no significant difference between the two bit error rates.

VIII. CONCLUSION

To defend against radio window attacks, we have developed and experimentally demonstrated a novel idea to secure CSI by altering the transmit signal in time, frequency and space to generate random human gestures that match the statistics of actual gestures in order to thwart an eavesdropper from being able to know when the actual gestures occurred. We present B210 USRP test setup, implement the PHY of a WiFi MIMO-OFDM link, measure the channel state information, and implement our proposed modification MoRTr at the transmitter. Our experimental results show that, with this modification, an eavesdropper is unable to distinguish between real and fake human gestures. Simultaneously, the legitimate receiver is able to extract and demodulate the payload of MoRTr-generated packets without any significant performance degradation.

ACKNOWLEDGEMENT

This material is based upon work supported by the Army Research Office under Grant No. W911NF-17-1-0457.

REFERENCES

[1] A. Banerjee, D. Maas, M. Bocca, N. Patwari, and S. Kasera, "Violating privacy through walls by passive monitoring of radio windows," in *ACM WiSec*, 2014.

[2] L. Gong, W. Yang, Z. Zhou, D. Man, H. Cai, X. Zhou, and Z. Yang, "An adaptive wireless passive human detection via fine-grained physical layer information," *Ad Hoc Networks*, 2016.

[3] C. Han, K. Wu, Y. Wang, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," in *IEEE INFOCOM 2014*, 2014.

[4] Y. Gu, J. Zhan, Y. Ji, J. Li, F. Ren, and S. Gao, "Mosense: An rf-based motion detection system via off-the-shelf wifi devices," *IEEE Internet of Things Journal*, 2017.

[5] S. Arshad, C. Feng, Y. Liu, Y. Hu, R. Yu, S. Zhou, and H. Li, "Wichase: A wifi based human activity recognition system for sensorless environments," in *IEEE WoWMoM*, 2017.

[6] D. S. et al., "Wibecam: Device free human activity recognition through wifi beacon-enabled camera," in *Workshop on Workshop on Physical Analytics*, 2015.

[7] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *ACM MobiCom*, 2013.

[8] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *ACM CCS*, 2016.

[9] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, 2011.

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

[11] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wi-fi," *IEEE Transactions on Mobile Computing*, 2019.

[12] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, 2017.

[13] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *ACM MobiCom*, 2015.

[14] Q. Zhou, J. Xing, and Q. Yang, "Device-free occupant activity recognition in smart offices using intrinsic wi-fi components," *Building and Environment*, 2020.

[15] Y. Gu, Y. Wang, Z. Liu, J. Liu, and J. Li, "Sleepguardian: An rf-based healthcare system guarding your sleep from afar," *IEEE Network*, 2020.

[16] H.-N. Dai, Q. Wang, D. Li, and R. C.-W. Wong, "On eavesdropping attacks in wireless sensor networks with directional antennas," *International Journal of Distributed Sensor Networks*, 2013.

[17] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Transactions on Wireless Communications*, 2008.

[18] J. Zhang, Z. Tang, M. Li, D. Fang, X. Chen, and Z. Wang, "Find me a safe zone: A countermeasure for channel state information based attacks," *Computers & Security*, 2019.

[19] J. Zhang, Z. Tang, R. Li, X. Chen, X. Gong, D. Fang, and Z. Wang, "Protect sensitive information against channel state information based attacks," in *IEEE CSE and IEEE EUC*, 2017.

[20] Y. Yao, Y. Li, X. Liu, Z. Chi, W. Wang, T. Xie, and T. Zhu, "Aegis: An interference-negligible rf sensing shield," in *IEEE INFOCOM*, 2018.

[21] Y. Qiao, O. Zhang, W. Zhou, K. Srinivasan, and A. Arora, "Phycloak: Obfuscating sensing from communication signals," in *In NSDI*, 2016.

[22] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac*, 2nd ed. USA: Cambridge University Press, 2013.

[23] Z. Wang, Z. Huang, C. Zhang, W. Dou, Y. Guo, and D. Chen, "CSI-based human sensing using model-based approaches: a survey," *Journal of Computational Design and Engineering*, 2021.

[24] I. E. Bagci, U. Roedig, I. Martinovic, M. Schulz, and M. Hollick, "Using channel state information for tamper detection in the internet of things," in *In ACSAC*, 2015.

[25] L. Cheng and J. Wang, "How can i guard my ap? non-intrusive user identification for mobile devices using wifi signals," in *ACM MoobiHoc*, 2016.

[26] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *In ACM MobiCom*, 2015.

[27] X. Liu, J. Cao, S. Tang, and J. Wen, "Wi-sleep: Contactless sleep monitoring via wifi signals," in *IEEE RTSS*, 2014.

[28] Y. Chen, W. Dong, Y. Gao, X. Liu, and T. Gu, "Rapid: A multimodal and device-free approach using noise estimation for robust person identification," *ACM IMWUT*, 2017.

[29] E. Research. (2020, April) B200/b210/b200mini/b205mini. [Online]. Available: https://kb.ettus.com/B200/B210/B200mini/B205mini

[30] "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016*, 2016.