WiCloak: Protect Location Privacy of WiFi Devices

Jinyan Jiang, Jiliang Wang[✉], Yihao Liu, Yijie Chen, Yunhao Liu
Tsinghua University, Beijing, P.R. China
{jiangjy23, liu-yh23, cyj20}@mails.tsinghua.edu.cn, {jiliangwang, yunhao}@tsinghua.edu.cn

ABSTRACT

The rapid development of WiFi localization poses a serious privacy threat, as eavesdroppers can locate WiFi devices without their consent. In this paper, we present WiCloak, the first system that protects WiFi device location privacy while supporting normal WiFi communication simultaneously. The high-level idea of WiCloak is to inject a fake channel into WiFi CSI at the transmitter, which renders the CIR and time information obtained by eavesdroppers meaningless. We mathematically prove that the injected fake channel is effective in any wireless environment and can strictly protect the location privacy of WiFi devices. To simultaneously support communication for commercial WiFi receivers, we propose a method to cancel out the fake channel impacts in decoding and prove that the method should not impact communication performance. WiCloak can work on commercial WiFi devices without any hardware modification. We evaluate the communication performance of WiCloak on commercial WiFi receivers (e.g., MacBook and Mac Studio) and demonstrate that it achieves the same packet reception rate as normal WiFi. We show that WiCloak increases the localization error by 22× to normal WiFi.

1 INTRODUCTION

Recently, WiFi chips are becoming ubiquitous. An increasing number of devices, such as mobile phones, laptops, and other smart home devices, are connected by WiFi. The shipment of WiFi 6 chips has already reached 3.8 billion and will reach 5.2 billion in 2025 [1]. On the one hand, WiFi provides a convenient way to connect devices. On the other hand, with the rapid development of wireless localization techniques [2–20], the location of WiFi devices can also be inferred by simply overhearing their packets. This poses a significant privacy concern, as an eavesdropper outside the wall can know the location of WiFi devices. For instance, an eavesdropper in the neighborhood can infer your location, behavior, and the location of devices in your home. Moreover, such eavesdroppers are hard to detect as they require no interaction with target devices and do not generate any traffic. Even worse, the available WiFi bandwidth keeps increasing, e.g., the bandwidth of 802.11ax reaches 160 MHz and that of 802.11be can even reach 320 MHz. This provides decimeter-level or even higher localization accuracy. There are already some tools to use WiFi to locate the user's location. It causes a lot of attention in the media [21, 22]. This paper aims to address this privacy concern and answer the critical question: *can we protect the location privacy of existing WiFi devices while supporting normal communication simultaneously*?

To answer this question, we first review existing WiFi localization techniques. Basically, there are Angle of Arrival (AoA) based localization approaches, and Time of Flight (ToF) based approaches. AoA-based approaches calculate the AoA of signals by antenna arrays. To obtain accurate localization results, they usually use a large antenna array. For example, ArrayTrack [23] uses an array of 16 antennas. Such a large antenna array is conspicuous, and thus it is not suitable for eavesdropping attacks. In addition, some methods [24, 25] use controllable reflectors or relay transmitters to interfere with the AoA localization. Thus, the focus of this paper is on dealing with the other category of ToF-based attacking methods. ToF-based approaches are developing rapidly recently [2, 6, 26–28]. Those approaches infer the CIR (Channel Impulse Response) of the signal by collecting the CSI (Channel State Information) on different subcarriers with only a single antenna. The first peak in the CIR corresponds to the ToF (with a certain shift) of the LoS (Line-of-Sight) path. For example, ToneTrack [26] calculates the CIR by splicing multiple 20 Mhz WiFi channels to generate a wider-band measurement and then calculates the TDoA (Time Difference of Arrival) by using two synchronized receivers. MonoLoco [9] calculates the geometric relation between LoS and reflected path, and achieves a median error of 0.5 m. SpotFi [8] achieves a median location accuracy of 0.4 m. We assume the attacker has multiple synchronized receivers and can estimate the CSI and CIR of the received packets. As the most covert means of attack at present, currently, no effective methods can defend against such ToF-based attackers. With the ongoing expansion of WiFi bandwidth, ToF-based attackers can achieve higher accuracy with imperceptible single antenna devices. We are determined to deal with this specific problem in this paper. There are also localization approaches based on RSSI [29–32], which rely on either pre-trained model or extensive data collection. Recently, Wi-Peep [33] measures the round-trip time by deceiving the target to reply to its packet. This can be defended by adding a random time delay while replying to the attacker [33].

In this paper, we focus on the very widely used ToF-based localization techniques, which can provide high accuracy with low eavesdropping overhead. We present *WiCloak*, the first approach that protects the location privacy of WiFi devices while supporting normal WiFi communication simultaneously. The basic idea of WiCloak is to manipulate the transmitted WiFi packet to make the calculated CSI at the eavesdropper useless for localization. To support packet transmission at legitimate receivers, WiCloak also compensates for the CSI change in the payload field and therefore enables standard WiFi decoding on COTS devices. We theoretically prove that WiCloak effectively conceals the localization from eavesdroppers while supporting normal communication.

The design of WiCloak consists of the following modules to address practical challenges.

(1) How to manipulate the CSI without affecting packet reception? The CSI indicates the attenuation and delay of the channel, which is obtained from the preamble field for each sub-carrier. It is usually estimated by comparing the pre-defined transmitted preamble and the actual received preamble propagating through the channel. For

[✉] Jiliang Wang is the corresponding author.

a normal receiver, the CSI is used to adjust the phase and amplitude of the payload symbols for packet decoding. To manipulate the CSI, we modify the preamble of a packet and add extra *fake channel* (i.e., extra amplitude and phase) to each symbol. As a result, any receiver will obtain the changed CSI based on the preamble. However, the CSI with this fake channel does not match the actual channel and cannot be used in decoding the payload of WiFi packets, resulting in WiFi communication failure. By studying the specification of each field in WiFi packets, we compensate the fake channel in the payload so that the manipulated CSI can be used to decode the payload. Meanwhile, we only change the phase of the channel for each subcarrier and do not change the amplitude or the orthogonality among them. We prove that our method can effectively manipulate the CSI while not affecting normal WiFi communication.

(2) How to effectively obfuscate the CIR information for the eavesdroppers? We assume eavesdropping receivers can effectively calculate the CSI based on the received preamble. Our goal is to find the most effective fake channel to add in order to conceal the actual channel information for various channel environments. We propose a method to determine the fake channel for different subcarriers. The method can make the eavesdropper obtain a random CIR without any noticeable peak. We also mathematically prove that the fake channel can work effectively in different environments.

(3) How to void fake channel being canceled? By obfuscating the CIR, the state-of-the-art approaches [26, 34, 35] cannot obtain useful ToF information anymore. We further assume an eavesdropper has multiple synchronized receivers capable of exchanging the overheard CSI. The eavesdropper can filter the fake channel by conjugating and multiplying the received CSI at two synchronized receivers. We show that in such a scenario the eavesdropper cannot differentiate accurate multipath and thus cannot obtain accurate location. Under a special case with a determined clear LoS to all receivers, the eavesdropper has the chance to remove the impact of the fake channel with the expense of quadratic increased multiple paths. We further present how to utilize WiFi beamforming to conceal the device location in multipath.

Main contributions and results:

- We show that WiFi devices face the threat of location leakage for ToF-based localization approaches and existing methods cannot effectively defend this. We propose WiCloak, the first location privacy protection system while supporting normal WiFi communication simultaneously. The basic idea of WiCloak is to manipulate the transmitted WiFi packets to add the fake channel in each sub-carrier. The eavesdropper cannot obtain the useful CSI and thus cannot locate WiFi devices. We also compensate the fake channel for the payload so that the legitimate WiFi receiver can decode the packet.
- We theoretically prove that WiCloak can effectively obfuscate the CIR on eavesdroppers and conceal the localization from eavesdroppers under various wireless channel. We also show that WiCloak can effectively support normal WiFi communication.
- We implement WiCloak by modifying the transmitted WiFi packets without any hardware modification. We conduct extensive experiments in different environments. The evaluation results show that the packets of WiCloak can be successfully decoded by commercial WiFi devices (e.g., MacBook, Mac Studio and Windows PC). We also implement and evaluate different types of attackers. The evaluation results show that WiCloak increases the localization error by 22×.

2 ATTACKING MODEL

A WiFi device sends WiFi packets through a wireless channel, as shown in Fig. 1(a). We assume the eavesdropper has a strong capability, which can intercept the traffic passively, and then decode the channel information, *i.e.,* CSI. The WiFi packet is modulated with OFDM and the CSI on each subcarrier is a complex value, including amplitude and phase, as shown in Fig. 1(b). We assume the eavesdropper is equipped with single antenna receivers and can monitor the WiFi channel and collect the CSI. The receivers are time synchronized. Besides, we also consider the eavesdropper with advanced counter method against WiCloak in § 4. The goal of eavesdroppers is to use the collected CSI to infer the location of the target device. The eavesdropper obtains the CIR by applying IFFT to the CSI sequence, and then infer the target location through the time delay in CIR, as shown in Fig. 1(c). Next, we explain the channel model and how the eavesdropper calculates the target location.

2.1 Signal Propagation Model

To send a WiFi packet, the target device first generates an OFDM-modulated signal at the baseband. Given N subcarriers in total, we first focus on how the i^{th} subcarrier (the middle subcarrier in the gray box of Fig. 1(a)) propagates in space. On baseband, the i^{th} subcarrier is $s(t) = e^{j2\pi f_i t}$. The target then up-converts the baseband signal to $S(t)$ in the carrier:

$$S(t) = s(t) \cdot e^{j(2\pi f_c t + \theta_{Tx})} \tag{1}$$

where f_c and θ_{Tx} are the frequency and initial phase of the carrier. We first consider the signal propagates through one path with delay τ and attenuation α. Then, the eavesdropper down-converts the overheard signal to the baseband $r(t)$:

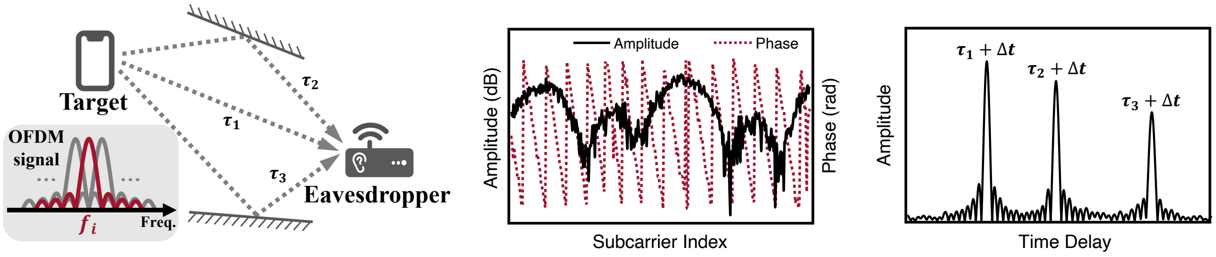
$$\begin{aligned} r(t) &= \alpha \cdot S(t-\tau) \cdot e^{j(-2\pi f_c t - \theta_{Rx})} \\ &= \alpha \cdot s(t-\tau) \cdot e^{-j2\pi f_c \tau} \cdot e^{j(\theta_{Tx} - \theta_{Rx})} \end{aligned} \tag{2}$$

where f_c and θ_{Rx} are the frequency and initial phase of the carrier signal at the eavesdropper. Here, we assume that the eavesdropper can eliminate the CFO (Carrier Frequency Offset) between the eavesdropper and target device by training symbols. Because the eavesdropper and target device are not synchronized, the eavesdropper uses a window offset by Δt to process the baseband signal $r(t)$:

$$r(t-\Delta t) = \alpha \cdot s(t-\tau-\Delta t) \cdot e^{-j2\pi f_c \tau} \cdot e^{j(\theta_{Tx} - \theta_{Rx})} \tag{3}$$

Because of the orthogonality of OFDM signals, the eavesdropper can obtain the CSI h_i on the i^{th} subcarrier by FFT and checking the output complex value of the i^{th} point:

$$\begin{aligned} h_i &= FFT_i\left[r(t-\Delta t)\right] \\ &= \alpha \cdot e^{-j2\pi f_i(\tau+\Delta t)} \cdot e^{-j2\pi f_c \tau} \cdot e^{j(\theta_{Tx} - \theta_{Rx})} = A \cdot e^{-j2\pi f_c^i(\tau+\Delta t)} \end{aligned} \tag{4}$$



(a) (b) (c)

Figure 1: (a) Signal from target device arrives at eavesdropper along three different paths. (b) Phase and amplitude of the measured CSI. (c) Three multi-paths are indicated in the derived CIR.

where $A = \alpha \cdot e^{j(\theta_{Tx} - \theta_{Rx})} \cdot e^{j2\pi f_c \Delta t}$ and $f_c^i = f_c + f_i$. The term A in Eq. 4 remains unchanged in each subcarrier. The phase of CSI $e^{-j2\pi f_c^i(\tau+\Delta t)}$, increases linearly with the subcarriers.

2.2 Infer CIR from CSI

In a real wireless environment, signals will propagate along different paths due to reflections such as walls. As shown in Fig. 1(a), the signal reaches the eavesdropper along three different paths. Generally, assume there are L different paths with time delay $[\tau_1, \tau_2, .., \tau_L]$ and attenuation $[\alpha_1, \alpha_2, .., \alpha_L]$. These multipath signals linearly add in the air and will jointly affect CSI. Because of the linear additivity of the FFT operation, the resulting CSI is also linearly affected by these multipath signals. Eq. 4 should be generalized as:

$$h_i = \sum_{l=1}^{L} A_l \cdot e^{-j2\pi f_c^i(\tau_l + \Delta t)} \tag{5}$$

CSI sequence $[h_1, h_2, .., h_N]$ can be collected on N subcarriers, as shown in Fig. 1(b). We obtain the CIR c_t by applying IFFT to this sequence:

$$c_t = \frac{1}{N} \sum_{i=1}^{N} e^{j2\pi f_c^i t} \cdot h_i \tag{6}$$

c_t will achieve a peak value when $t = \tau_l + \Delta t$ in Eq. 6. When the bandwidth is large enough, c_t will be 0 under other values. In this case, we have:

$$c_t \approx \sum_{l=1}^{L} A_l \cdot \delta\left[t - (\tau_l + \Delta t)\right] \tag{7}$$

where $\delta(\cdot)$ is the delta function. As shown in Fig. 1(c), after IFFT, we obtain three peaks corresponding to three multipath in Fig. 1(a).

2.3 Calculate Location from CIR

Existing ToF methods calculate location based on CIR. We also assume the eavesdropper has this capability. The eavesdropper finds the first peak in the CIR, which corresponds to the LoS path to the target. However, this peak is subjected to a time shift Δt due to the eavesdropper's lack of synchronization with the target. To address this, we assume the eavesdropper has two synchronized receivers (*e.g.,* Rx1 and Rx2) to process the same data packet. This synchronization can be achieved by cables [2], or wireless synchronization techniques [36]. By extracting the first peak, the two receivers of the eavesdropper obtain the delay $\tau_1^1 + \Delta t$ and $\tau_1^2 + \Delta t$ for the LoS path, respectively. Then, subtracting those two values yields $(\tau_1^1 + \Delta t) - (\tau_1^2 + \Delta t) = \tau_1^1 - \tau_1^2$, allowing the eavesdropper to determine that the target is on a hyperbola with Rx1 and Rx2 as the focus. With one more Rx, another hyperbola can be determined and the intersection of two hyperbolas is the location of the target.

2.4 Generality of the Attacking Model

The above attacking model is the foundation for ToF-based localization approaches. For example, ToneTrack [26] and Splicer [27] use multiple 20 MHz CSI measurements to achieve high-bandwidth localization. MonoLoco [9] uses the ToF differences to construct the geometric relationship. SpotFi [8] identifies the LoS path using the first peak of the CIR. μLocate [2] and Owll [6] implement ToF-based localization in LoRa and LoRa backscatter. SAIL [37] and Chronos [7] exchange packets and analyze CSI/CIR for localization. ISLA [35] enables ToF-based localization in a 5G cellular network. The design of WiCloak can be generalized to these methods.

3 WICLOAK DESIGN

The main working flow of the eavesdropper is to infer CIR from CSI based on the packets sent by target devices. Thus, the goal of WiCloak is to change the calculated CSI at the eavesdropper and make the result of CIR meaningless. Meanwhile, a normal WiFi receiver should be able to receive the WiFi data packets. We first introduce how WiCloak can manipulate the CSI. Then, we show how WiCloak can conceal the location information. Finally, we introduce how to ensure normal data decoding in WiCloak.

3.1 Manipulate CSI

For an eavesdropper, it first detects the packet and estimates the CSI. More specifically, given a pre-defined symbol X_i at the i^{th} subcarrier, an eavesdropper will receive $Y_i = h_i \cdot X_i$ after passing through channel h_i, as shown in Fig. 2(a). Thus, the eavesdropper calculate h_i as:

$$\frac{Y_i}{X_i} = \frac{h_i \cdot \cancel{X_i}}{\cancel{X_i}} = h_i \tag{8}$$

To conceal the location, it is important to obfuscate the CSI at the eavesdropper. The basic idea is to embed an additional fake channel $M_i = a_i e^{j\theta_i}$ to the preamble. In other words, instead of sending X_i directly in the preamble, we send $M_i \cdot X_i$. After propagation through the channel h_i, the eavesdropper receive $\hat{Y}_i = h_i \cdot M_i \cdot X_i$. The eavesdropper does not know the value of M_i as it is generated by the Tx. It uses Eq. 8 for CSI estimation, and calculates $\hat{h}_i$ at subcarrier i as:

$$\hat{h}_i = \frac{\hat{Y}_i}{X_i} = \frac{h_i \cdot M_i \cdot \cancel{X_i}}{\cancel{X_i}} = h_i \cdot M_i \tag{9}$$

Next, we show how to set M_i to hide the location of Tx from the eavesdropper.

We need to determine the value of fake channel M_i so that the eavesdroppers cannot recover the real CIR in different wireless channel environments. Suppose the real channel for different subcarrier is $[h_1, \ldots, h_i, \ldots, h_N]$ as described in Eq. 5. We set the fake channel to $[M_1, \ldots, M_i = e^{j\theta_i}, \ldots, M_N]$, where $\theta_i \sim U(0, 2\pi)$ and θ_i is independent with θ_j $(i \neq j)$. Here, we set the amplitude of the fake channel to be one in order not to impact the signal SNR. We can add the extra fake channel M_i to a WiCloak packet. Thus, the calculated CSI for this packet at the receiver now is $[\hat{h}_1, \ldots, \hat{h}_i = h_i{\cdot}M_i, \ldots, \hat{h}_N]$.By applying the IFFT operation described in Eq. 6 with this obfuscated CSI sequence, we obtain the CIR $\hat{c}_t$. We show that the obtained CIR $\hat{c}_t$ is a random sequence without noticeable peaks. More specifically, we theoretically prove the *CIR obfuscation effectiveness*: the CIR cannot be recovered in any wireless environment as long as the θ_i in M_i conforms to a uniform distribution in 0 and 2π. We will prove this in two steps: (1) we first prove it for the wireless channel containing only a single path, and then (2) we extend the proof to practical wireless channels with multipath.

3.1.1 Single Path Scenario. As shown in Fig. 2(a), we first analyze the CIR obfuscation effectiveness for the single path scenario. According to Eq. 4, the real channel $h_i = A \cdot e^{-j2\pi f_c^i(\tau+\Delta t)}$. According to Eq. 6, the CIR after injecting the fake channel is:

$$\hat{c}_t = \frac{1}{N}\sum_{i=1}^{N} e^{j2\pi f_c^i t} \cdot h_i \cdot M_i \quad = \frac{1}{N}\sum_{i=1}^{N} F_t^i \cdot M_i \tag{10}$$

where $F_t^i = A \cdot e^{j2\pi f_c^i(t-\tau-\Delta t)}$. The additional obfuscation data M_i is now added to the calculation of CIR. To show the effectiveness of CIR obfuscation, we first show how $\hat{c}_t$ is distributed. As $M_i = e^{j\theta_i}$, where θ_i is uniformly distributed between 0 and 2π, the probability density function of θ_i is:

$$p_{\theta_i}(x) = \frac{1}{2\pi}, 0 \leq x < 2\pi \tag{11}$$

The mean and variance of M_i are:

$$\mathbb{E}[M_i] = \int_0^{2\pi} p_{\theta_i}(\theta)e^{j\theta}\mathrm{d}\theta = \frac{1}{2\pi j}\left. e^{j\theta}\right|_0^{2\pi} = 0 \tag{12}$$

$$\mathrm{Var}[M_i] = \mathbb{E}[M_i \cdot M_i^*] - \mathbb{E}[M_i]^2 = \mathbb{E}[e^{j\theta_i} \cdot e^{-j\theta_i}] - 0 = 1 \tag{13}$$

We find the random variable $\hat{c}_t$ in Eq. 10 can be expressed as a linear combination of the random variable sequence $[M_1, \ldots, M_i, \ldots, M_N]$. The CIR $\hat{c}_t$ at different t has different coefficients F_t^i in this linear summation process. Therefore, we can use the mean $\mathbb{E}[M_i]$ and variance $\mathrm{Var}[M_i]$ of M_i (derived from Eq. 12 and 13) to describe the distribution of $\hat{c}_t$. We calculate the mean of $\hat{c}_t$:

$$\mathbb{E}\left[\hat{c}_t\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} F_t^i \cdot M_i\right] = \frac{1}{N}\sum_{i=1}^{N} F_t^i \cdot \mathbb{E}\left[M_i\right] = 0 \tag{14}$$

Then, we calculate the variance of $\hat{c}_t$:

$$\begin{aligned}\mathrm{Var}\left[\hat{c}_t\right] &= \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} F_t^i \cdot M_i\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\left|F_t^i\right|^2 \mathrm{Var}\left[M_i\right] + \frac{2}{N^2}\sum_{p,q} F_t^p F_t^q \,\mathrm{Cov}(M_p, M_q)\end{aligned} \tag{15}$$

where $1 \leq p < q \leq N$. In Eq. 15, The amplitude of F_t^i is always α regardless of i and t. Here we normalize the attenuation α to 1. As the random variable M_i on different subcarriers are independent, we have $\mathrm{Cov}(M_p, M_q) = 0$ when $p \neq q$. By substituting these two values into Eq. 15, we have:

$$\mathrm{Var}\left[\hat{c}_t\right] = \frac{1}{N^2}\sum_{i=1}^{N} 1^2 \cdot 1 + \frac{2}{N^2}\sum_{p,q} F_t^p F_t^q \cdot 0 = \frac{1}{N} \tag{16}$$

Eq. 14 and 16 demonstrate that the mean and variance of $\hat{c}_t$ are constant, regardless of the value of t. In contrast, the original real CIR c_t reflects the correlation between the selected time t and the channel delay τ. When time t is equal to τ, c_t should exhibit an obvious peak (ignoring the time asynchronous Δt between transceivers). However, the estimated CIR $\hat{c}_t$ is a random variable, and its mean and variance do not change with the value of t, including when t is exactly equal to τ. The randomly distributed phase of WiCloak shown in Fig. 2(b) turns to be randomly distributed CIR shown in Fig. 2(c). Consequently, the CIR derived from WiCloak lacks an obvious peak as in Fig. 1(c). Thus, the obfuscated CSI cannot be used for localization in this scenario.

3.1.2 Multipath Scenario. Suppose the l^{th} path experiences an additional time τ_l and amplitude attenuation α_l compared to the LoS path. Therefore, the CIR corresponding to the l^{th} path, denoted as $\alpha_l \hat{c}_{t-\tau_l}$, is time-shifted by τ_l and is scaled by α_l to $\hat{c}_t$ obtained in Eq. 10. The multipath signals are linearly added to the air and processed by the receiver. As described in Eq. 5, the CIR $\hat{C}_t$ is the linear addition of respective CIRs of each path as the IFFT operation is also linear:

$$\hat{C}_t = \sum_{l=1}^{L} \alpha_l \hat{c}_{t-\tau_l} \tag{17}$$

We have demonstrated that the mean and variance of $\hat{c}_t$ are independent of t. As a result, in Eq. 17, $\mathbb{E}\left[\hat{c}_{t-\tau_l}\right] = \mathbb{E}\left[\hat{c}_t\right] = 0$, and $\mathrm{Var}\left[\hat{c}_{t-\tau_l}\right] = \mathrm{Var}\left[\hat{c}_t\right] = \frac{1}{N}$. We observe that $\hat{C}_t$ is essentially a linear combination of L random variables, similar to $\hat{c}_t$ in Eq. 10. According to Eq. 14 and 15, the mean of $\hat{C}_t$ is:

$$\mathbb{E}\left[\hat{C}_t\right] = \sum_{l=1}^{L} \alpha_l \mathbb{E}\left[\hat{c}_{t-\tau_l}\right] = 0 \tag{18}$$

Similarly, the variance of $\hat{C}_t$ is:

$$\mathrm{Var}\left[\hat{C}_t\right] = \sum_{l=1}^{L} |\alpha_l|^2 \,\mathrm{Var}\left[\hat{c}_{t-\tau_l}\right] + 2\sum_{p,q} \alpha_p \alpha_q \,\mathrm{Cov}(\hat{c}_{t-\tau_p}, \hat{c}_{t-\tau_q}) \tag{19}$$

where $1 \leq p < q \leq L$. In Eq. 19, the only unknown quantity is $\mathrm{Cov}(\hat{c}_{t-\tau_p}, \hat{c}_{t-\tau_q})$, *i.e.,* the covariance between $\hat{c}_{t-\tau_p}$ and $\hat{c}_{t-\tau_q}$. We set $t_1 = t - \tau_p$ and $t_2 = t - \tau_q$. Now, we need to calculate the value of $\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2})$ when $t_1 \neq t_2$. This is equivalent to finding

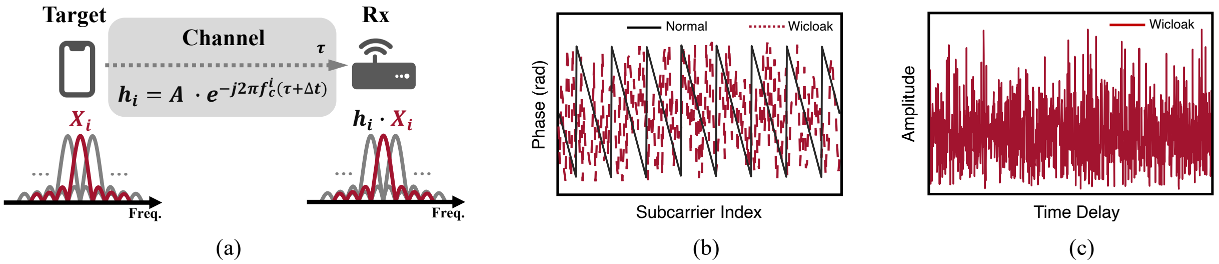


Figure 2: (a) X_i becomes $h_i \cdot X_i$ after passing through the channel h_i. (b) The phase of CSI from the WiCloak packet becomes random in WiCloak. (c) CIR derived from the WiCloak packet has no noticeable peak.

the correlation between any two different points in the CIR $\hat{c}_t$ for a single path. In Appendix A, we prove $\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2}) = 0$ as any two different points in $\hat{c}_t$ have a correlation of 0. Substituting this result into Eq. 19, we can calculate the variance of $\hat{C}_t$:

$$\mathrm{Var}\left[\hat{C}_t\right] = \sum_{l=1}^{L} |\alpha_l|^2 \frac{1}{N} + 2\sum_{p,q} \alpha_p \alpha_q \cdot 0 = \frac{1}{N}\sum_{l=1}^{L} |\alpha_l|^2 \qquad (20)$$

In Eq. 18 and 20, we find that the mean and variance of the random variable $\hat{C}_t$ are also constant, regardless of the value of t. Therefore, in the real-world multipath scenario, $\hat{C}_t$ has no obvious peak after obfuscation. Since any channel can be expressed as a combination of multiple single-path channels, as shown in Eq. 17, the obfuscation can be extended to various wireless channels.

3.2 WiFi Communication

To decode the payload, the receiver uses the obtained h_i to restore the OFDM symbol after passing through the channel. Specifically, suppose a WiFi transmitter sends $dataX_i$ on the i^{th} subcarrier, and the received data is $dataY_i = h_i \cdot dataX_i$ in the coherence time of the channel. Then, the original $dataX_i$ can be obtained as:

$$\frac{dataY_i}{h_i} = \frac{\cancel{h_i} \cdot dataX_i}{\cancel{h_i}} = dataX_i \qquad (21)$$

Normal Communication with WiFi NICs. For normal communication, the receiver also calculates CSI as $h_i \cdot M_i$. Obviously, due to the existence of the fake channel M_i, the receiver cannot correctly decode the packet payload. To recover the data field correctly, an intuitive approach is to share the value of M_i between the transmitter and the legitimate receiver. The receiver recovers the correct CSI h_i by taking $M_i \cdot X_i$ into the denominator of Eq. 9. A feasible way is to use pre-shared key to generate M_i between transmitter and receiver. However, it requires protocol updates for those receivers with simple communication purposes. To address this, we take a different way to decode the packet. Instead of restoring the correct h_i, we change the transmitted packet payload at the Tx to neutralize the impact of the fake channel. We multiply $dataX_i$ by M_i and transmit the resulting data $\widehat{dataX}_i = M_i \cdot dataX_i$. For the legitimate receiver, the obtained CSI is also distorted as $\hat{h}_i = h_i \cdot M_i$. The received payload becomes $\widehat{dataY}_i = h_i \cdot \widehat{dataX}_i = h_i \cdot M_i \cdot dataX_i$. Using Eq. 21, the original payload can be decoded as $\frac{\widehat{dataY}_i}{\hat{h}_i} = dataX_i$. This process is seamless for commercial WiFi NICs. Descrambling, mapping, checking and other processes will not be affected. Commercial NICs treat packets in WiCloak format as normal 802.11 packets. Our experiment in § 6.1 also shows that WiCloak can work well with current 802.11-compliant NICs.

No SNR Loss. We further show that WiCloak introduces no SNR loss. Theoretically, multiplying both preamble and payload with $M_i = a_i e^{j\theta_i}$ brings SNR loss. To be specific, if the normalized amplitude of M_i varies among subcarriers, it is equivalent to varying the transmit power of subcarriers with different coefficients. This may reduce the power at certain subcarriers and decrease the SNR of those subcarriers. To avoid affecting normal communication, we only change the phase information in the data packet. In other words, WiCloak sets $a_i = 1$ in fake channel M_i and thus $M_i = e^{j\theta_i}$. By doing so, the power on each subcarrier does not change and thus there is no SNR loss in decoding.

Impact on High-order Modulation. To enable higher data rates, WiFi utilizes higher-order modulation and coding methods, indicated by Modulation and Coding Schemes (MCS). WiCloak can still decode $dataX_i$ by synchronously changing its phase at the transmitter, as described in Eq. 21.

4 DEAL WITH ADVANCED EAVESDROPPERS

We have demonstrated that WiCloak obfuscates the CIR of potential eavesdroppers without compromising WiFi communication, rendering them not able to detect any peaks in the CIR using non-interactive localization techniques. Existing localization approaches such as ToneTrack [26] and Splicer [27] cannot work under the protection of WiCloak. However, if attackers become aware of the existence of WiCloak, they may enhance their attack methods. We show how WiCloak can work under different scenarios and how to avoid the fake channel being canceled.

4.1 Advanced Attacking Model

The basic design of WiCloak ensures that the CIR at eavesdroppers becomes a random sequence. Assume an advanced eavesdropper notices the existence of WiCloak, e.g., by observing the absence of any noticeable peak in the CIR. The eavesdropper can then develop a new attacking model to bypass the effect of WiCloak to deal with the obfuscated CIR.

Localization methods based on ToF require identifying the first peak in the CIR and obtaining the time delay difference between different receivers. After obfuscation, the CIR of each receiver alone has no peaks. By exploiting multiple Rxs, we can multiply the measured CSI by its complex conjugate at different synchronized receivers to remove the fake channel. Here we assume the advanced eavesdropper can share the CSI value among receivers. Specifically, given the obfuscation coefficient $M_i = e^{j\theta_i}$ on the i^{th} subcarrier, the CSI measured by Rx1 is $\hat{h}_i^1 = h_i^1 \cdot M_i$, as shown in Eq. 9. Similarly, the CSI measured by Rx2 is $\hat{h}_i^2 = h_i^2 \cdot M_i$. The random coefficients M_i on both Rx1 and Rx2 are the same. By conjugated multiplication of the obfuscation CSI information on two Rxs, we have:

$$h_i^{1,2} = \hat{h}_i^1 \cdot \left[\hat{h}_i^2\right]^* = h_i^1 \cdot e^{j\theta} \cdot \left[h_i^2\right]^* \cdot e^{-j\theta} = h_i^1 \left[h_i^2\right]^* \tag{22}$$

The injected random phase ($e^{j\theta}$) in the fake channel M_i can be eliminated. So, the eavesdropper will perform the IFFT operation on the conjugated-multiplied CSI $h_i^{1,2}$ to obtain the obfuscation-free CIR. However, the LoS delay is now hidden in the multipath. The number of multipaths is also increased by square, which reduces the localization accuracy. To understand this, we explain it through a simple example.

Assume that the target sends a signal that reaches Rx1 through two paths with delays of 4 ns and 7 ns, and with attenuations of a_1^1 and a_2^1, respectively. Similarly, the signal also reaches Rx2 through two paths with delays of 1 ns and 3 ns, and attenuations of a_1^2 and a_2^2, respectively. We assume the eavesdropper can distinguish all multipaths. For Rx1, according to Eq. 5 and Eq. 9, the CSI measured at the i^{th} subcarrier is $\hat{h}_i^1 = \left[a_1^1 e^{-j2\pi f_c^i \cdot 4} + a_2^1 e^{-j2\pi f_c^i \cdot 7}\right] \cdot M_i$. For Rx2, the measured CSI is $\hat{h}_i^2 = \left[a_1^2 e^{-j2\pi f_c^i \cdot 1} + a_2^2 e^{-j2\pi f_c^i \cdot 3}\right] \cdot M_i$. Here, we ignore the time and phase shift between transceivers. And we assume the eavesdropper can eliminate this shift. The eavesdropper multiplies those two CSI measurements by their complex conjugates and has $h_i^{1,2}$ as:

$$\begin{aligned} h_i^{1,2} &= \left[a_1^1 e^{-j2\pi f_c^i \cdot 4} + a_2^1 e^{-j2\pi f_c^i \cdot 7}\right] \cdot \left[a_1^2 e^{-j2\pi f_c^i \cdot 1} + a_2^2 e^{-j2\pi f_c^i \cdot 3}\right]^* \\ &= b_1 e^{-j2\pi f_c^i \cdot 1} + b_2 e^{-j2\pi f_c^i \cdot 3} + b_3 e^{-j2\pi f_c^i \cdot 4} + b_4 e^{-j2\pi f_c^i \cdot 6} \end{aligned} \tag{23}$$

where $b_1 = a_1^1 a_2^2$, $b_2 = a_1^1 a_1^2$, $b_3 = a_2^1 a_2^2$ and $b_4 = a_2^1 a_1^2$. The original two paths of Rx1 and Rx2 are now combined into four paths. If the eavesdropper perform IFFT on $h_i^{1,2}$, four peaks will appear in CIR $c_t^{1,2}$ when $t = 1$, 3, 4 and 6 ns. To identify the LoS path with the CIR derived from the channel h_i^1 or h_i^2 measured by a single receiver, the eavesdropper finds the first peak in the CIR, which corresponds to the smallest delay. However, this no longer works for the CIR $c_t^{1,2}$ derived from $h_i^{1,2}$. The eavesdropper cannot identify the LoS path by the peak with the shortest delay. Instead, the first peak with the smallest delay 1 ns is the shortest (LoS) delay received by Rx1 (4 ns) minus the longest delay received by Rx2 (3 ns). Similarly, the delay 6 ns corresponding to the last peak is the longest delay received by Rx1 (7 ns) minus the shortest (Los) delay received by Rx2 (1 ns). For the other two delays in the middle of $c_t^{1,2}$ (3 ns and 4 ns), the eavesdropper cannot determine which of them corresponds to the shortest (LoS) delay of Rx1 minus the shortest (LoS) delay of Rx2. Now, the TDoA of the target should be 4 ns (LoS delay) minus 1 ns (LoS delay) which equals 3 ns. However, it is also possible that the two delays received by Rx1 are 5 ns and 7 ns, while the two delays received by Rx2 are 1 ns and 4 ns. In this case, there are also four peaks exactly the same as in $c_t^{1,2}$, but the target TDoA is actually 4 ns.

This problem is even more severe in real indoor wireless channels. Typically, there are 4 ∼ 5 main propagation paths [7]. When these paths are combined in pairs to form the CIR $c_t^{1,2}$, the resulting CIR contains 16 ∼ 25 peaks. However, the eavesdropper can only reliably determine the difference of the shortest and longest delay in the $c_t^{1,2}$, leaving the target TDoA hidden among the remaining peaks. Furthermore, this approach also causes a quadratic increase in the number of multipaths. It requires more bandwidth to separate those peaks. In our example before, the closest delay between the peaks is $3 - 1$ =2 ns in the respective CIR of Rx2, while in $c_t^{1,2}$, the two closest peaks are $4 - 3$ =1 ns apart. According to the principle that the time measurement accuracy is inversely proportional to the bandwidth, this doubles the bandwidth requirement to separate the peaks that are 1 ns apart.

In an ideal case for the eavesdropper, it can still use a best-effort approach. Assume that the LoS path is the strongest for both Rx1 and Rx2. In $c_t^{1,2}$, the eavesdropper can find the strongest peak that corresponds to the correct TDoA. For example, if $a_1^1 \geq a_2^1$ and $a_1^2 \geq a_2^2$, b_2 will be the strongest peak. However, this method has significant limitations. Typically, the eavesdropper has no strong LoS path to the target; otherwise, it should be able to directly know the target position. For example, if the user is at home, the eavesdropper is generally located outside the wall, and the LoS signal can be significantly attenuated. As long as one receiver does not have the strongest LoS path to the target, the eavesdropper cannot work. Additionally, it is difficult for the eavesdropper to determine whether the LoS is the strongest and when to use the strongest peaks. Even if there is a clear LoS path between the eavesdropper and target device, it is not guaranteed that the eavesdropper has the strongest LoS path due to factors such as antenna polarization, lobe width, device orientation, device position, etc. [38, 39].

4.2 Leveraging Beamforming

If the WiFi device supports beamforming, we can further improve WiCloak. Even when all receivers of an eavesdropper always have a LoS path with the strongest amplitude to the target (which should not be the practice case), we can still address this. We can leverage beamforming techniques in WiFi communication to enhance signal strength in specific directions. After detecting the training sequence sent by legitimate AP, a WiFi device equipped with multiple antennas will add different phase coefficients to different antennas to amplify the signal strength toward the direction of AP. We can utilize beamforming to amplify any reflected signal to eavesdroppers so that the highest peak in $c_t^{1,2}$ does not always correspond to the correct TDoA. For example, in Fig. 4(a), there are five paths. By enhancing the reflection signal in the direction corresponding to τ_2 or τ_3 while weakening the LoS signal corresponding to τ_1, the error increases in the estimation of the target TDoA.

For example, we put two WiFi transceivers at a distance of 6 m in a room 7 m × 12 m, as shown in Fig. 4(a), and plot CIR when LoS path exists. As shown in Fig. 3(a), we can observe five distinct peaks. In this scenario, we only need to slightly increase any reflection path to exceed the intensity of the LoS path. After the Tx generates a beam towards the legitimate Rx as shown in Fig. 4(a) with two

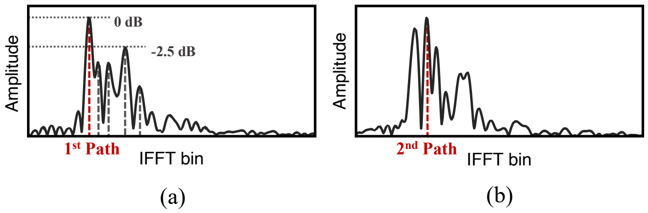


Figure 3: (a) CIR of a typical indoor environment. (b) WiCloak increases the signal strength of the 2^{nd} path.

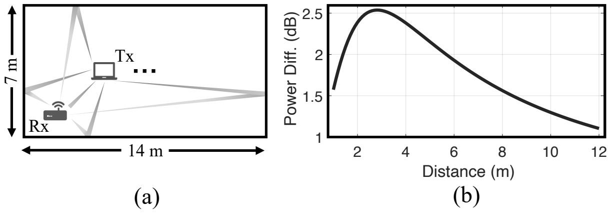


Figure 4: (a) Propagation model of the field. (b) Power differences of the LoS and strongest reflected path.

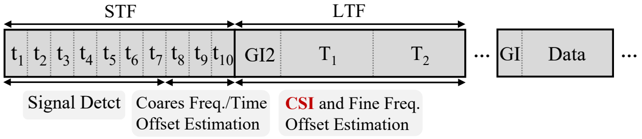

STF
LTF
t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
GI2
T1
T2
...
GI
Data
...
Signal Detct
Coares Freq./Time Offset Estimation
CSI and Fine Freq. Offset Estimation

Figure 5: WiFi packet structure.

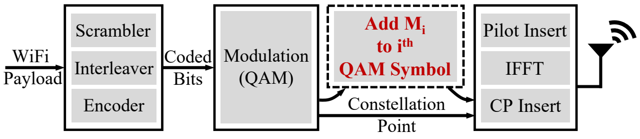


Figure 6: WiCloak transmitter design.

antennas, the amplitude of the second path is amplified and exceeds that of the LoS path, as shown in Fig. 3(b). To further investigate the signal propagation model in this scenario, we analyze it based on the ray-tracing method [40]. As shown in Fig. 4(a), we put the Rx on one side of the room, and move the Tx from the left-hand side to the right-hand side in the middle of the room. We estimate the power difference between the LoS path and the most potent reflected path. Here, we consider four walls as the main obstacles as shown in Fig. 4(a). As illustrated in Fig. 4(b), the propagation paths change as the Tx gradually moves away from the Rx. Consequently, the power difference first increases and then decreases. The maximum is a difference of around 2.5 dB, which is consistent with our observation in the first experiment. Furthermore, we found that even a two-antenna beam exhibits sufficient azimuth sensitivity. For a 2.5 dB power difference, it requires at most 18° angular offset in the generated beam.

In practice, the beam directions are fixed to the targeted APs who have a limited number. However, we find that WiCloak does not necessarily increase the reflected signal received by each Rx. For example, when three Rxs calculate two TDoAs to determine their intersection point, a strong reflection signal at one of the Rxs can cause errors in one TDoA estimation, resulting in the failure of final localization results. Therefore, WiCloak only needs to generate a random beam for eavesdroppers and does not need to steer it in any arbitrary directions. We note that It is worth noting that the obfuscating in the phase of CSI is essential even with multiple equipped antennas on the Tx. If we only randomly generate beams, Rx can still infer the LoS path by searching for the first peak (although the first peak has become lower), as shown in Fig. 4(b). WiCloak can utilize the beamforming in WiFi to provide comprehensive protection in extreme cases.

5 IMPLEMENTATION

WiCloak packets generation. The main fields of an 802.11 data packet have three parts as shown in Fig. 5 The packet starts with the STF (Short Training Field), which consists of ten identical short OFDM symbols, each lasting 0.8 µs. The first seven short symbols are used for packet detection. The remaining three are used for coarse frequency and time alignment. Following the STF is the LTF (Long Training Field), which consists of two long OFDM symbols, each lasting 4 µs. The LTF is used to estimate CSI. To generate PHY I/Q points, we use the *wlanWaveformGenerator* tools in Matlab and adjust the phase of each subcarrier in the LTF and Payload fields simultaneously. We introduce a random value M_i with phase between 0 and 2π. The change to a conventional WiFi transmitter is as shown in Fig. 6. After encoding and modulation, conventional transmitters directly apply IFFT to constellation points in the frequency domain. However, WiCloak takes one more step after modulation to add extra value M_i to i^{th} QAM.

Attacking methods. We develop two kinds of attacking methods. The vanilla one (as described in § 2) is based on a general model widely used in various ToF-based systems [2, 6, 7, 26–28]. The eavesdropper applies IFFT on the CSI sequence of multiple synchronized receivers and calculates the ToA based on the first-time impulse. Then it records TDoA and calculates the target location. We then develop an advanced attacking method to cancel the random value M_i in CSI by conjugated-multiplying the measurements of two receivers, as described in § 4.1. We evaluate the WiCloak for both the vanilla and the advanced attacking method.

Hardware. We implement the WiCloak transmitter on USRP X310 [41]. By default, the transmitter uses a two-antenna array, each link with a transmitting power of 17 dBm. We implement the WiCloak receiver on various COTS devices, such as the Broadcom BCM7BF and BCM4387 NIC [42] on a MacBook pro laptop and a Mac studio desktop, and the Intel AX210 NIC [43] on an x86 PC. We use the AirPort Utility on MacOS to collect WiCloak packets. We use the Picoscenes tool [44] on Linux to collect the CSI of WiCloak packets received by AX210.

6 EVALUATION

6.1 Compatibility with COTS Devices

6.1.1 Basic WiFi Communication. We first measure the packet reception rate (PRR) of WiCloak packets on a MacBook and compare it with that of normal WiFi packets. Both types of packets are

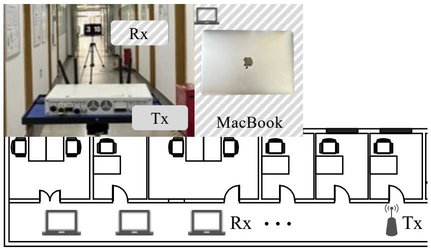


Figure 7: Experimental field and equipment.

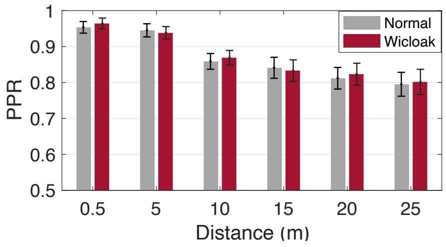


Figure 8: PRR comparison with different communication distance.

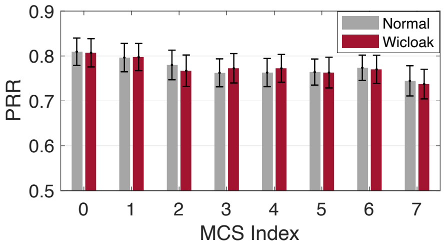
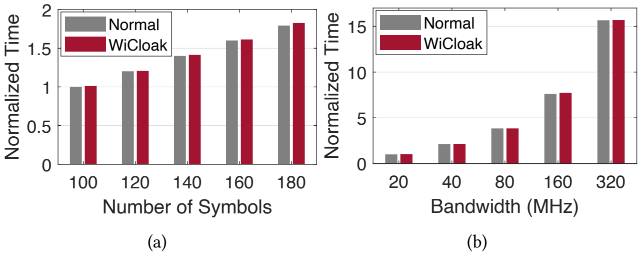


Figure 9: PRR comparison with different MCS.

nonHT 20 MHz WiFi beacons transmitted using USRP with the same parameters: single antenna, transmitting power of 20 dBm, BPSK modulation, and coding rate of 1/2. Here we don't use the generated beams to evaluate because in practice, WiCloak can always steer the beam in the direction of the legitimate receiver. In this case, there will be an SNR gain, which is unfair for normal WiFi packets. In this section, we intend to only evaluate the influence of concurrently adding the obfuscation phase in both training and payload fields. We evaluate the PRR with different communication distances in an indoor corridor with a size of 28×2.3 m, as shown in Fig. 7. We put the WiCloak Tx on one end of the corridor and move the Rx (MacBook) away from 0.5 m to 25 m. At each location, we send 3,000 WiCloak and normal packets and record the PRR on the MacBook. At different distances, we receive both WiCloak and normal packets. We find that the content of the received packet is the same as the transmitted one. As shown in Fig. 8, WiCloak achieves a PRR similar to that of normal packets. This is because WiCloak can cancel out the impact of the fake channel. When Tx and Rx are 0.5 m apart, both normal packets and WiCloak data packets achieve a PRR above 95%. The PRR finally falls to about 80% at 25 m away.

6.1.2 Different MCS. Then, we investigate the impact of different modulation and coding schemes (MCS) on PRR. The 802.11 standard allows for four modulation methods in a nonHT format packet: BPSK, QPSK, 16QAM, and 64QAM. Each modulation method can go with a 1/2 or 3/4 coding rate (CR) (except for 64QAM, which goes with CR 2/3 or 3/4), resulting in a total of 8 possible combinations. We keep Tx and Rx 14 m apart and measure the PRR of both WiCloak and normal packets under these 8 parameters. The index of MCS is the same as that in the 802.11 standard [45], and a larger index means a higher-order modulation and a greater coding rate. Fig. 9 shows that WiCloak achieves similar PRRs to normal packets across different MCSs. When the MCS index goes up, the PRR decreases. This is because higher-order modulation and coding schemes need higher SNR, so they are more likely to experience packet losses. Nevertheless, our experiment proves that WiCloak works well with commercial WiFi NIC in different scenarios.

6.1.3 Impact on Other Devices. To evaluate WiCloak 's impact on a real network with diverse devices, including non-WiCloak WiFi devices and devices with other protocols, we conduct an experiment to measure interference caused by WiCloak packets and normal packets. We generate non-HT 20 MHz WiCloak and normal packets and continuously transmit them via an X310 on WiFi channel 8. We measure the interference on BLE traffic on the overlapped BLE channel 39. An NRF52840 module [46] captured emitted BLE packets at a distance of 8 meters from a USRP N210 that concurrently transmitted BLE advertising packets alongside the WiFi interference. We change the transmission power of the interference. Both WiCloak and normal packets exhibited comparable PRR for BLE traffic in different power settings. Subsequently, we assess interference on non-HT packets transmitted by a co-located N210 on WiFi channel 9. It shows there is no additional interference caused by WiCloak when compared to normal packets.



(a)

(b)

Figure 10: Time consumption for generating packets

6.2 Computation Overhead

As described in § 5, there is more extra step needed for generating WiCloak packets than normal packets, which is multiplying the obfuscation phase to the QAM symbol after scrambler, interleaving, etc. It will slow the generation of packets. In this section, we evaluate this extra time delay and whether it affects the throughput. We find that in WiFi protocol, there is a Short Interframe Space (SIFS) defined. For example, when sending data packets, the transmitter will wait for the time of SIFS to receive the ACK packet and send the next packet. Thus, as long as the duration of producing a WiCloak packet is less than SIFS, it shouldn't cause WiFi throughput degrades. During the generation of the WiFi packet, the most time-consuming step is IFFT, which has a time complexity of $O(n \cdot log(n))$. Most else steps such as scramble and mapper only have a complexity of $O(n)$. Luckily, the extra step of multiplying the obfuscation phase token by WiCloak also with a complexity of $O(n)$. So, it won't cause much extra delay for WiFi packet generation.

We first measure the time consumption for generating the waveform of normal and WiCloak packets in Matlab. We use the timer functions *tic* and *toc* in Matlab to record the time it takes to generate

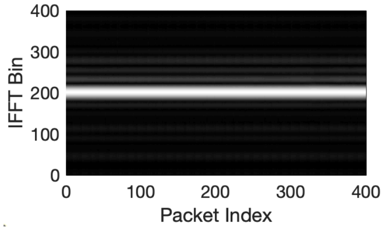


Figure 11: CIR for normal packets.

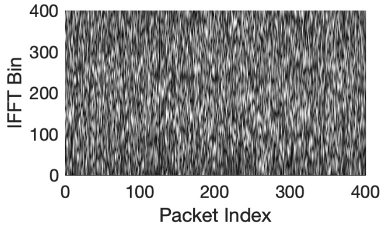


Figure 12: CIR for WiCloak packets.

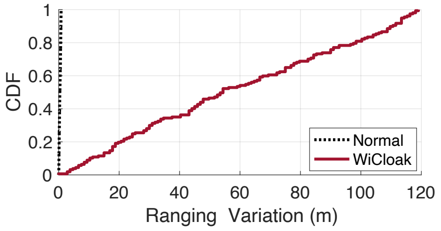


Figure 13: Ranging variation.

VHT 20 MHz WiFi packets in both normal and WiCloak formats under different packet lengths. The result is shown in Fig. 10(a). We normalize the time consumption to 1 for generating a normal packet with 100 symbols. The time consumption for generating WiCloak is only slightly larger than that for normal packets. Overall, WiCloak only increases the time consumption by 3%. We also measure the time consumption for generating the LTF field under different bandwidths. We normalize the time consumption to 1 for generating a normal packet with 20 MHz. The results are shown in Fig. 10(b). A higher bandwidth signal contains more sampling points to process, which incurs higher time consumption. WiCloak packets also reach a similar time consumption with normal packets with an average increment of only 2.5%. In an open-source repertory related to WiFi packet generation Sora [47], the end-to-end time delay to compose a 20 MHz WiFi packet with a 128-byte payload is only 360 ns. Even when considering the extra time computations by WiCloak , it is far less than the SIFS, which is 16 μs in the current WiFi protocol. Thus, WiCloak won't cause throughput degradation.

6.3 CIR Obfuscation

To verify the performance of CIR obfuscation, we use the AX210 NIC to collect the CSI of both normal and WiCloak packets. We send 400 normal packets and 400 WiCloak packets. Each WiCloak packet contains a different and uncorrelated obfuscation sequence. We set Rx and Tx 3 m apart and ensure a LoS path. We do not change the wireless environment of the test area during the experiment. We use ToneTrack [26] to remove time misalignment among packets in this experiment. Then, we calculate the CIR using the IFFT operation based on the collected CSI. Normal packets should have a noticeable peak in the CIR that indicates the LoS path; WiCloak packets should have a random CIR without any noticeable peak.

We show the CIR of normal data packets in Fig. 11. The horizontal axis in this figure represents the index of the packet, and the vertical axis represents the IFFT bin index in the CIR of each data packet. We observe that CIRs of normal packets have a clear peak at the same position, as illustrated by a white line. However, as shown in Fig. 12, WiCloak packets have no obvious peak. The CIRs among different packets are not correlated as the embedded fake channel is random. The IFFT bins of all packets are also randomly distributed.

At this point, the eavesdropper cannot obtain any valid information from a single Rx due to the absence of a clear peak in the CIR. To quantify the effectiveness of WiCloak, we compare the ranging results corresponding to the highest peak value of normal packets and WiCloak packets in CIRs. The CIR of each WiCloak packet is a random distribution sequence with equal mean and variance, and each IFFT bin has an equal probability of being a peak. Consequently, the ranging results of WiCloak are uniformly distributed across the entire delay range, as shown in Fig. 13. In contrast, normal data packets have a clear and consistent peak value, resulting in a ranging result with a small error.

6.4 End-to-end Localization Performance

We set up a test bed in the indoor office environment as shown in Fig. 17. Three APs with AX210 NICs are placed in Room 1. We test the performance of WiCloak under different experimental settings, such as LoS or NLoS scenarios and various bandwidths. We find that the multiple antennas of the AX210 are clock-synchronized, which enables measuring the location of the target device through TDoA. The target device and APs operate in the 5.25 GHz band and use 160 MHz packets in HESU format. In this experiment, we assume the SNR is enough for at least one legitimate AP to receive regardless of the beam direction. We manually adjust the phase differences of the two antenna arrays to generate beams. We have shown that the CIR from a single AP cannot be used for localization in § 6.3. Eavesdroppers can only follow the advanced method which assumes there is a LoS path, and locates the target by multiplying CSI from different Rxs. We show WiCloak can also work under this kind of eavesdropper.

6.4.1 Basic Performance. We place the target device at the center of Room 1, as indicated by the black cross in Figure. 17. We conduct two sets of experiments with and without WiCloak . Without WiCloak, the eavesdropper can achieve precise localization results with an error of only 48 cm , as illustrated by the green point in Fig. 17. Thus, the eavesdropper can infer fine-grained user information and engage in malicious activities. With WiCloak, the eavesdropper can only use the CIR obtained after conjugated multiplication. It cannot find the LoS path and causes a larger localization error.

We test how WiCloak works in real environments more. We first fix the phase differences between the two antennas to 0 and create a beam straight ahead. Then, we turn the orientation of the device from 0° to 180° and increase the angle by 20° each time. Figure. 17 illustrates the results. WiCloak gives wrong localization results for all nine target device orientations. This is because, in indoor environments, the number of multipaths increases after conjugated multiplication. The eavesdropper can generate many possible locations. These nine positions exhibit substantial errors. The smallest error is larger than 4 m and the biggest one is 12 m. We also observe that the localization accuracy varies with the orientation. The orientation to the biggest error is only 40° different

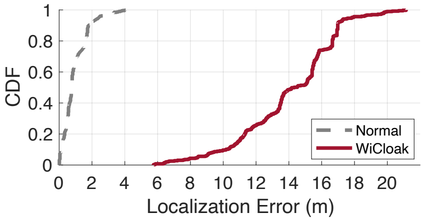


Figure 14: Error in NLoS deployment.

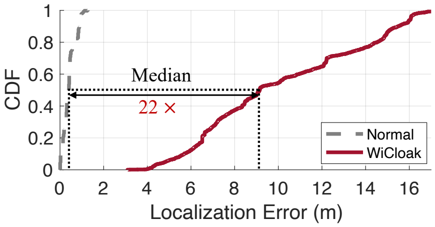


Figure 15: Error in LoS deployment.

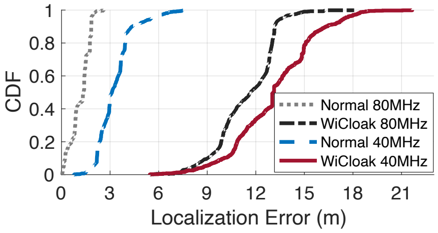


Figure 16: Error for different BW.

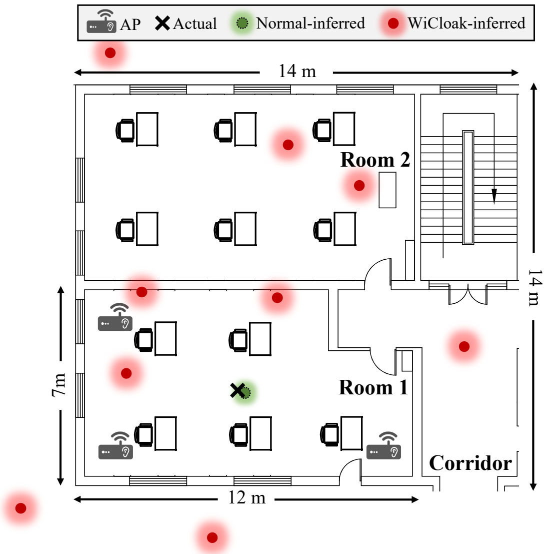


Figure 17: Localization test bed.

from the orientation of the smallest error. Practically, the random orientation of the target device causes eavesdroppers not to be able to obtain the correct location.

6.4.2 Localization error. We evaluate the overall performance of WiCloak in Room 1. We divide Room 1 into 5 areas equally, each has an area of about 4 m by 3 m. We measure the localization error by placing the target device at three random locations in each area. We run the experiments by steering the beam of the target device to two opposite directions at each position. We ensure that there is always no LoS path between the target device and one of the gateways. We collect the CSI of 600 data packets at each position and calculate the mean localization error. The receivers are placed in three corners as shown in Fig. 17. Fig. 14 shows that the median localization error of normal packets is 0.75 m. WiCloak increases the median localization error to 14.56 m in this case even when the eavesdropper can conduct advanced localization method in § 4.1. This is because it is difficult to guarantee a LoS path to every Rx with the strongest signal strength. In practice, the eavesdropper may be in a more complicated environment, e.g., outside of a wall. The localization error for WiCloak can be even larger.

6.4.3 Ideal case with LoS to all receivers. In the experiment, we put the target device on a shelf and ensure a LoS path from the target to all Rxs. Fig. 15 shows the distribution of localization errors. For normal packets, the median localization error is only 0.41 m due to the accurate CIR results of the large bandwidth. However, WiCloak packets completely obfuscate the CIR, rendering large location errors. WiCloak increases the median error to 9.14 m, which is about 22× the normal results. Considering that the size of Room 1 is only 12 m by 7 m, the localization error caused by WiCloak prevents eavesdroppers from obtaining meaningful results. In the worst case, the localization error caused by WiCloak is still more than 3 m. Further checking the positions of the worst case, we find that a slight change in beam direction can significantly increase the localization error. This is consistent with observations in § 6.4.1. In most cases, diverse generated beams will cause the eavesdropper to obtain two completely different localization results with large errors. Note that this experiment for the LoS scenario ensures that the target device has a LoS path to all Rxs used for localization. This is an ideal attack scenario for eavesdroppers.

6.4.4 Impact of bandwidth. In this experiment, we evaluate the impact of bandwidth on localization accuracy. The overall experimental setup is the same as described in § 6.4.3. We conduct measurements with 80 MHz and 40 MHz bandwidth in different locations. As shown in Fig. 16, increasing the bandwidth of normal data packets results in a significant improvement in the localization accuracy. The median localization error decreased from 3.05 m for 40 MHz packets to only 1.33 m for 80 MHz packets. The error is reduced by 2.3 × by increasing the bandwidth from 40 MHz to 80 MHz. However, with WiCloak, the eavesdropper cannot obtain meaningful localization results for either 80 MHz or 40 MHz bandwidth. The median localization error is larger than 10 m in both cases. The localization error of 40 MHz is only 1.1× greater than that of 80 MHz. This experiment confirms that WiCloak is compatible with various settings of bandwidth, and it is effective for increased bandwidth.

7 RELATED WORKS

Wireless localization. ToneTrack [26] combines multiple 20 MHz narrow bandwidth WiFi signals to form up to 80 MHz high-accuracy measurements and deduce the location of WiFi devices through TDoA. MonoLoco [34] utilizes the ToF relationship between LoS and multipath information to get locations with a single AP. Spotfi [8] performs joint estimation of AoA and ToF through CSI of multiple antennas at 20 MHz. Splicer [27] derives high-resolution CIR by splicing the amplitude and phase of CSI from multiple bands. UAT [48] improves the reliability of AoA localization. FUSIC [49] combines Fine Timing Measurement (FTM) and MUSIC algorithms to improve the localization performance of NLoS. SAIL [37] combines the measured ToF with the smartphone dead-reckoning techniques and employs a geometric relationship to infer the location using a single AP. Chronos [7] exchanges packets between transceivers to remove phase uncertainty and derives an accurate CIR. RFind [10] utilizes a large bandwidth excitation signal to enable localization for RFID. RF-Chord [50] features a multisite-constructed wideband design to facilitate one-shot localization at scale. ISLA [35] enables ToF-based localization in the cellular network. MAVL [51] localizes sound sources using estimated AoA and room structure. TagFi [52] implements localization for WiFi backscatter tags.

Wireless sensing privacy. PhyCloak [25] uses a relay transmitter to confuse the frequency, phase, and other information in WiFi sensing. The attacking methods defended by PhyCloak needs to tracks the phase and amplitude differences among different packets or symbols to obtain the movement of the target. PhyCloak alters the phase and amplitude in the created reflection path and changes the setting among packets to protect the differences among packets. Within the duration between two configuration alterations (around 100 ms) in PhyCloak, a WiFi packet is transmitted and it's sufficient for the attacker to get the location. As the reflection path always has a longer propagation distance than the LoS path, an attacker still can simply check the first peak in the CIR to infer the location. It also may cause degradation in localization accuracy, as it creates more multipath in the environment. However, modern WiFi with high bandwidth can separate the reflection path and mitigate its influence. [3] and [53] add irreversible I/Q samples in the time domain to hide the signal reflected by the human body. CSI fuzzer [54] adds multipath in the CSI of openwifi [55]. However, the constructed multipath must be longer than the LoS path, so it cannot resist ToF-based eavesdroppers. SNOOPDOG [56] uses causality between wireless traffic and a trusted sensor to detect malicious sensors. [57] proposes recommendations for improving the privacy and security of in-car wireless sensor networks. IRShield [58] uses controllable metasurfaces to hide WiFi sensing data. Wi-Peep [33] confirms that WiFi devices can be induced to send data packets to calculate the signal round-trip delay and studies the methods against such attacks. RF-protect [59] constructs a multi-antenna reflector to hide the FMCW signal reflected by humans. Lumos [60] identifies and locates hidden IoT devices. There is currently no defense against ToF-based non-interactive localization.

8 CONCLUSION

In this paper, we present WiCloak, the first system to protect WiFi device location privacy while supporting normal WiFi communication simultaneously. WiCloak manipulates the packet at the transmitter by injecting a fake channel to change the CSI captured by eavesdroppers. Thus, eavesdroppers can only obtain meaningless CIR results. To support normal communication on commercial devices, WiCloak also changes the payload fields to cancel out the fake channel. Even for an ideal eavesdropper with multiple synchronized receivers having strong LoS to the target device. WiCloak utilizes beamforming in WiFi standards to amplify any reflection signal to make the eavesdropper difficult to extract useful location information. We conduct extensive experiments to evaluate the performance of WiCloak. The results show that WiCloak can communicate with commercial WiFi NICs (e.g., MacBook and Mac Studio) and achieve the same packet reception rate. WiCloak increases localization error of normal WiFi localization by 22×. It demonstrates that WiCloak provides strict protection for the location privacy of WiFi devices.

9 ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments to improve the quality of our work. This work is in part supported by National Key R&D Program of China 2022YFC3801300, National Natural Science Foundation of China (U22A2031, No. 61932013, 62172250).

A APPENDIX

Here we prove the covariance between different CIR $\hat{c}_{t_1}$ and $\hat{c}_{t_2}$ for a single path is equal to 0. The covariance can be written as:

$$\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2}) = \mathbb{E}\left[\left(\hat{c}_{t_1} - \mathbb{E}\left[\hat{c}_{t_1}\right]\right)\left(\hat{c}_{t_2} - \mathbb{E}\left[\hat{c}_{t_2}\right]\right)^*\right] \tag{24}$$

As we have proven $\mathbb{E}\left[\hat{c}_t\right] = 0$ in Eq. 14, it can be simplified as:

$$\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2}) = \mathbb{E}\left[\hat{c}_{t_1}\hat{c}_{t_2}^*\right] \tag{25}$$

According to the Eq. 10, we compute it as follows:

$$\begin{aligned}
&\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2}) \\
&= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} F_{t_1}^{i} \cdot M_i\right)\left(\frac{1}{N}\sum_{i=1}^{N} F_{t_2}^{i} \cdot M_i\right)^*\right] \\
&= \mathbb{E}\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} F_{t_1}^{i}\left(F_{t_2}^{j}\right)^* M_i M_j^*\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} F_{t_1}^{i}\left(F_{t_2}^{j}\right)^* \cdot \mathbb{E}\left[M_i M_j^*\right]
\end{aligned} \tag{26}$$

Note that we have previously computed $\mathbb{E}\left[M_i\right] = 0$ in Eq. 12 and $\mathrm{Var}\left[M_i\right] = 1$ in Eq. 13. For those terms where $i = j$, we have:

$$\begin{aligned}
&\frac{1}{N^2}\sum_{i=1}^{N} F_{t_1}^{i}\left(F_{t_2}^{i}\right)^* \cdot \mathbb{E}\left[M_i M_i^*\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{N} F_{t_1}^{i}\left(F_{t_2}^{i}\right)^* \cdot \mathbb{E}\left[\left(M_i - \mathbb{E}\left[M_i\right]\right)\left(M_i - \mathbb{E}\left[M_i\right]\right)^*\right] = 0
\end{aligned} \tag{27}$$

For those terms where $i \neq j$, since M_i and M_j are independent of each other, we have:

$$\mathbb{E}\left[M_i M_j^*\right] = \mathbb{E}\left[\left(M_i - \mathbb{E}\left[M_i\right]\right)\left(M_j - \mathbb{E}\left[M_j\right]\right)^*\right] = 0 \tag{28}$$

Again, the sum of these terms is equal to 0. Therefore, we prove $\mathrm{Cov}(\hat{c}_{t_1}, \hat{c}_{t_2}) = 0$, which means that different CIR $\hat{c}_{t_1}$ and $\hat{c}_{t_2}$ for a single path are uncorrelated to each other.

REFERENCES

[1] WiFi Alliance. WiFi 6 shipments. https://www.WiFi.org/beacon/the-beacon/WiFi-6-shipments-to-surpass-52-billion-\by-2025.

[2] Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota. 3d localization for sub-centimeter sized devices. In *Proceedings of ACM Sensys*, 2018.

[3] Lorenzo Ghiro, Marco Cominelli, Francesco Gringoli, and Renato Lo Cigno. Wifi localization obfuscation: An implementation in openwifi. *Elsevier Computer Communications*, 2023.

[4] Guanglin Zhang, Anqi Zhang, Ping Zhao, and Jiaxin Sun. Lightweight privacy-preserving scheme in wifi fingerprint-based indoor localization. *IEEE Systems Journal*, 14(3):4638–4647, 2020.

[5] Zhe Chen, Guorong Zhu, Sulei Wang, Yuedong Xu, Jie Xiong, Jin Zhao, Jun Luo, and Xin Wang. M^3: Multipath assisted wifi localization with a single access point. *IEEE Transactions on Mobile Computing (TMC)*, 20(2):588–602, 2019.

[6] Atul Bansal, Akshay Gadre, Vaibhav Singh, Anthony Rowe, Bob Iannucci, and Swarun Kumar. OwLL: Accurate LoRa Localization using the TV Whitespaces. In *Proceedings of ACM/IEEE IPSN*, 2021.

[7] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-Level localization with a single WiFi access point. In *Proceedings of USENIX NSDI*, 2016.

[8] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of ACM SIGCOMM*, 2015.

[9] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. Multipath Triangulation: Decimeter-Level WiFi Localization and Orientation with a Single Unaided Receiver. In *Proceedings of ACM MobiSys*, 2018.

[10] Yunfei Ma, Nicholas Selby, and Fadel Adib. Minding the billions: Ultra-wideband localization for deployed rfid tags. In *Proceedings of ACM MobiCom*, 2017.

[11] Yaxiong Xie, Yanbo Zhang, Jansen Christian Liando, and Mo Li. Swan: Stitched wi-fi antennas. In *Proceedings of ACM MobiCom*, 2018.

[12] Haoyu Wang and Wei Gong. RF-Pen: Practical Real-Time RFID Tracking in the Air. *IEEE Transactions on Mobile Computing*, 20(11):3227–3238, 2021.

[13] Chenren Xu, Bernhard Firner, Yanyong Zhang, and Richard E Howard. The case for efficient and robust rf-based device-free localization. *IEEE Transactions on Mobile Computing*, 15(9):2362–2375, 2015.

[14] Qiang Yang and Yuanqing Zheng. Deepear: Sound localization with binaural microphones. In *Proceedings of IEEE INFOCOM*, 2022.

[15] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. Lifs: Low human-effort, device-free localization with fine-grained subcarrier information. In *Proceedings of ACM MobiCom*, 2016.

[16] Han Ding, Jinsong Han, Chen Qian, Fu Xiao, Ge Wang, Nan Yang, Wei Xi, and Jian Xiao. Trio: Utilizing tag interference for refined localization of passive rfid. In *Proceedings of IEEE INFOCOM*, 2018.

[17] Ge Wang, Chen Qian, Longfei Shangguan, Han Ding, Jinsong Han, Nan Yang, Wei Xi, and Jizhong Zhao. Hmrl: Relative localization of rfid tags with static devices. In *Proceedings of IEEE SECON*, 2017.

[18] Qin Shi, Sihao Zhao, Xiaowei Cui, Mingquan Lu, and Mengdi Jia. Anchor self-localization algorithm based on uwb ranging and inertial measurements. *Tsinghua Science and Technology*, 24(6):728–737, 2019.

[19] Jian-feng ZHU, Xin-sheng HE, and Ben-jian HAO. A hybrid localization technology for an aerial moving target based on tdoa of dual-satellite and doa of main satellite. *ACTA ELECTONICA SINICA*, 46(6):1378, 2018.

[20] Qiang Yang and Yuanqing Zheng. Aquahelper: Underwater sos transmission and detection in swimming pools. In *Proceedings of ACM Sensys*, 2023.

[21] The Hacker News. New whiffy recon malware triangulates infected device location via wi-fi every minute. https://thehackernews.com/2023/08/new-whiffy-recon-malware-triangulates.html.

[22] PCMag Middle East. New malware component can use wi-fi triangulation to determine pc's location. https://me.pcmag.com/en/security/18955/new-malware-component-can-use-wi-fi-triangulation-to-determine-pcs-location.

[23] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Proceedings of USENIX NSDI*, 2013.

[24] Paul Staat, Simon Mulzer, Stefan Roth, Veelasha Moonsamy, Markus Heinrichs, Rainer Kronberger, Aydin Sezgin, and Christof Paar. Irshield: A countermeasure against adversarial physical-layer wireless sensing. In *Proceedings of IEEE S&P*, 2022.

[25] Yue Qiao, Ouyang Zhang, Wenjie Zhou, Kannan Srinivasan, and Anish Arora. Phycloak: Obfuscating sensing from communication signals. In *Proceedings of USENIX NSDI*, 2016.

[26] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proceedings of ACM MobiCom*, 2015.

[27] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity wifi. In *Proceedings of ACM MobiCom*, 2015.

[28] Jinyan Jiang, Jiliang Wang, Yijie Chen, Yihao Liu, and Yunhao Liu. LocRa: Enable Practical Long-Range Backscatter Localization for Low-Cost Tags. In *Proceedings of ACM Mobisys*, 2023.

[29] Zheng Yang, Chenshu Wu, and Yunhao Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of ACM Mobicom*, 2012.

[30] Chenshu Wu, Jingao Xu, Zheng Yang, Nicholas D Lane, and Zuwei Yin. Gain without pain: Accurate wifi-based localization using fingerprint spatial gradient. In *Proceedings of ACM UbiComp*, 2017.

[31] Suining He, Tianyang Hu, and S-H Gary Chan. Contour-based trilateration for indoor fingerprinting localization. In *Proceedings of ACM Sensys*, 2019.

[32] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of ACM MobiCom*, 2009.

[33] Ali Abedi and Deepak Vasisht. Non-cooperative wifi localization & its privacy implications. In *Proceedings of ACM MobiCom*, 2022.

[34] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of ACM Mobisys*, 2018.

[35] Suraj Jog, Junfeng Guan, Sohrab Madani, Ruochen Lu, Songbin Gong, Deepak Vasisht, and Haitham Hassanieh. Enabling IoT Self-Localization Using Ambient 5G Signals. In *Proceedings of USENIX NSDI*, 2022.

[36] Kubra Alemdar, Divashree Varshney, Subhramoy Mohanti, Ufuk Muncuk, and Kaushik Chowdhury. Rfclock: timing, phase and frequency synchronization for distributed wireless networks. In *Proceedings of ACM MobiCom*, 2021.

[37] Alex T Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. Sail: Single access point-based indoor localization. In *Proceedings of ACM Mobisys*, 2014.

[38] Lili Chen, Wenjun Hu, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Jeremy Gummeson. Pushing the physical limits of iot devices with programmable metasurfaces. In *Proceedings of USENIX NSDI*, 2021.

[39] Robert F Linfield, RW Hubbard, and LE Pratt. *Transmission channel characterization by impulse response measurements*, volume 76. US Department of Commerce, Office of Telecommunications, 1976.

[40] Hank Weghorst, Gary Hooper, and Donald P Greenberg. Improved computational methods for ray tracing. *ACM Transactions on Graphics (TOG)*, 3(1):52–69, 1984.

[41] Etuss Research. USRP X310. https://www.ettus.com/all-products/x310-kit/.

[42] Broadcom. Broadcom WiFi NICs. https://www.broadcom.com/products/wireless/wireless-lan-bluetooth.

[43] Intel. AX210 WiFi NICs. https://www.intel.sg/content/www/xa/en/products/sku/204836/intel-wifi-6e-ax210-gig/specifications.html.

[44] Zhiping Jiang. PicoScenes. https://www.intel.sg/content/www/xa/en/products/sku/204836/intel-wifi-6e-ax210-gig/specifications.html.

[45] IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021.

[46] NORDIC. nRF52840. https://www.nordicsemi.com/products/nrf52840.

[47] Kun Tan, He Liu, Jiansong Zhang, Yongguang Zhang, Ji Fang, and Geoffrey M Voelker. Sora: high-performance software radio using general-purpose multi-core processors. In *Proceedings of USENIX NSDI*, 2009.

[48] Tzu-Chun Tai, Kate Ching-Ju Lin, and Yu-Chee Tseng. Toward reliable localization by unequal aoa tracking. In *Proceedings of ACM MobiSys*, 2019.

[49] Kevin Jiokeng, Gentian Jakllari, Alain Tchana, and André-Luc Beylot. When ftm discovered music: Accurate wifi-based ranging in the presence of multipath. In *Proceedings of IEEE INFOCOM*, 2020.

[50] Bo Liang, Purui Wang, Renjie Zhao, Heyu Guo, Pengyu Zhang, Junchen Guo, Shunmin Zhu, Hongqiang Harry Liu, Xinyu Zhang, and Chenren Xu. RF-Chord: Towards deployable RFID localization system for logistic networks. In *Proceedings of USENIX NSDI*, 2023.

[51] Mei Wang, Wei Sun, and Lili Qiu. MAVL: Multiresolution analysis of voice localization. In *Proceedings of USENIX NSDI*, 2021.

[52] Elahe Soltanaghaei, Adwait Dongare, Akarsh Prabhakara, Swarun Kumar, Anthony Rowe, and Kamin Whitehouse. Tagfi: Locating ultra-low power wifi tags using unmodified wifi infrastructure. In *Proceedings of ACM UbiComp*, 2021.

[53] Marco Cominelli, Felix Kosterhon, Francesco Gringoli, Renato Lo Cigno, and Arash Asadi. An experimental study of csi management to preserve location privacy. In *Proceedings of ACM WiNTECH*, 2020.

[54] Xianjun Jiao, Michael Mehari, Wei Liu, Muhammad Aslam, and Ingrid Moerman. openwifi csi fuzzer for authorized sensing and covert channels. In *Proceedings of ACM WiSec*, 2021.

[55] Xianjun Jiao, Wei Liu, Michael Mehari, Muhammad Aslam, and Ingrid Moerman. openwifi: a free and open-source ieee802. 11 sdr implementation on soc. In *Proceedings of IEEE VTC*, 2020.

[56] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani Srivastava. I always feel like somebody's sensing me! a framework to detect, identify, and localize clandestine wireless sensors. In *Proceedings of USENIX Security*, 2021.

[57] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and privacy vulnerabilities of In-Car wireless networks: A tire pressure monitoring system case study. In *Proceedings of USENIX Security*, 2010.

[58] Paul Staat, Simon Mulzer, Stefan Roth, Veelasha Moonsamy, Markus Heinrichs, Rainer Kronberger, Aydin Sezgin, and Christof Paar. Irshield: A countermeasure against adversarial physical-layer wireless sensing. In *Proceedings of IEEE S&P*, 2022.

[59] Jayanth Shenoy, Zikun Liu, Bill Tao, Zachary Kabelac, and Deepak Vasisht. Rf-protect: privacy against device-free human tracking. In *Proceedings of ACM SIGCOMM*, 2022.

[60] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. Lumos: Identifying and localizing diverse hidden IoT devices in an unfamiliar environment. In *Proceedings of USENIX Security*, 2022.