Preamble Forgery and Injection in Wi-Fi Networks: Attacks and Defenses

Zhengguang Zhang, *Student Member, IEEE,* Marwan Krunz, *Fellow, IEEE*

Abstract—In Wi-Fi networks, the preamble plays a crucial role in frame detection, synchronization, and channel estimation. It also ensures compatibility and interoperability across devices that operate different versions of Wi-Fi (e.g., IEEE 802.11a/g/n/ac/ax/be). Despite its significance, the preamble lacks authenticity and confidentiality guarantees, relying solely on weak integrity protection. In this paper, we introduce novel Preamble Injection and Spoofing (PrInS) attacks that exploit these vulnerabilities. Specifically, we show how an adversary can inject forged preambles without payloads to disrupt legitimate receptions or force legitimate users to defer transmissions. We demonstrate the impact of PrInS attacks both via experiments using software-defined radios (SDRs) and via system-level simulations. Our results show that the adversary can almost silence the channel, degrading the throughput of a legitimate user down to 2% of its normal throughput. Even at 30 dB less power than the legitimate signal, the adversary still causes 87% reduction in throughput. Even when the attacker targets only a fraction of legitimate frames, the average packet latency and packet loss rate significantly increase. As a countermeasure, we propose preamble customization and randomization using group keys and timestamps, along with preamble authentication in the receive state machine. Our countermeasure detects forged preambles with nearly 100% accuracy while maintaining low false alarm rates in most scenarios. Most importantly, it remains backward-compatible with existing 802.11 standards and does not impact the synchronization and frame error rates of the Wi-Fi system.

Index Terms—Wi-Fi networks, forgery attacks, spoofing attacks, physical-layer security, denial-of-service.

1 Introduction

In the past few years, Wi-Fi experienced unprecedented growth, with over 22.2 billion Wi-Fi devices reported worldwide in 2022 [2]. Such ubiquitous deployment raises security concerns about Wi-Fi networks. Several medium access control (MAC) layer attacks were identified in the literature, including address spoofing [3], downgrade and dictionary attacks against WPA3 [4], and beacon announcement forgery [5]. Likewise, Physical (PHY)-layer vulnerabilities that lead to privacy leakage [6] and jamming [7] were identified. In response to the growing security concerns, various techniques were proposed to enhance Wi-Fi security, including beacon protection [8], friendly jamming [9], and PHY encryption [10]. Most of these techniques focus on the payload of the PHY frame, with little attention given to the protection of the preamble.

Fundamentally, the vulnerability of the frame preamble is rooted in the fact that it is publicly known and is decodable by any Wi-Fi device. As shown in Fig. 1, in Wi-Fi standards that use orthogonal frequency division multiplexing (OFDM), the preamble is composed of Training and Signal (SIG) fields. The preamble's primary purpose is to facilitate the reception and interpretation of the payload. It also conveys the frame duration, which is needed to reserve the channel. Thus, an attack on the preamble has far-reaching implications on the Wi-Fi network performance. In [11] the authors presented an attack that disrupts frame timing by jamming the Training field with continuous noise or false Training symbols. The jamming signal of the Training field in [12] was carefully designed to spoof the receiver into incorrect carrier frequency offset (CFO) estimation, hence corrupting the data. However, because the Training fields arrive in the first few microseconds of a frame, jamming them reactively is practically challenging. The above attacks incur high energy/exposure because they involve persistent jamming, or they are hard to implement in practice due to the need to act reactively with precise timing.

In contrast to prior attacks that aim at undermining functions of the preamble, our focus in this paper is on attacks that *exploit* these functions. The significance of these latter attacks is accentuated by the fact that more system-level information is being conveyed in the SIG fields.

In this paper, we demonstrate novel and practical attacks on the Wi-Fi preamble. Our attacks exploit inherent vulnerabilities in the preamble, along with the PHY-layer receive state machine and the capture effect. Specifically, we present Preamble Injection and Spoofing (PrInS) attacks in which the adversary injects a preamble with forged SIG fields to deceive neighboring devices and make them either defer channel access or receive packets incorrectly. These PrInS attacks are driven by malicious goals to disrupt mission-critical Wi-Fi applications in two aspects. First, the throughput reduction caused by PrInS attacks can lead to denial-of-service (DoS). In a Wi-Fi based healthcare application, DoS prevents data exchange between wearable devices, mobile medical equipment, patients, and healthcare staff [13], [14]. In a Wi-Fi mesh network deployed for public safety, real-time broadcast of critical alerts and mobile access to criminal data [15] will be negatively impacted by PrInS attacks. DoS can also impose severe threats on industrial IoT [16], [17]. Secondly, PrInS attacks dramatically increase the packet latency of legitimate users to up to several sec-

• *Zhengguang Zhang and Marwan Krunz are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721. E-mail: {zhengguangzhang,krunz}@arizona.edu*

An abridged version of this paper was presented at the IEEE GLOBECOM 2021 Conference, Dec. 7-11 2021, Madrid, Spain [1].
Manuscript received 8 March 2023, revised 5 Sep. 2023, and 27 Feb. 2024.

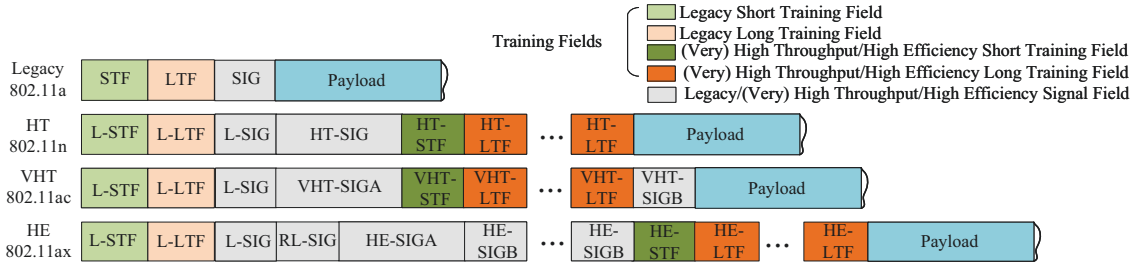


Fig. 1. Frame preambles in IEEE 802.11a/n/ac/ax standards.

onds. Such high latency is unacceptable in time-critical Wi-Fi applications such as remote surgery and extended reality (XR). At the same time, the adversary gains from PrInS attacks. Due to PrInS attacks, legitimate devices struggle to access the channel or they experience high frame error rates. As a result, these devices are compelled to switch to an alternate channel. This action effectively relinquishes the original channel to the adversary, substantially improving her throughput.

We study three PrInS attacks that can silence the channel, corrupt frame detection, falsify received data, and potentially drain the batteries of receiving devices. These attacks are effective irrespective of the Wi-Fi version used by the adversary or the targeted device, because they are based on vulnerabilities in the preamble and receive state machine that are universal in IEEE 802.11 versions. Furthermore, the impact of PrInS attacks extends beyond Wi-Fi systems to other systems, e.g., IEEE 802.11p/bd for vehicle networking.

Compared to jamming and injecting malicious MAC frames, PrInS attacks are more stealthy in the sense that they consume less power and last for a shorter duration. They are also more elusive because existing security mechanisms cannot detect PHY-layer threats and the induced errors are often misattributed to poor channel quality (blockage, fading, and interference). Furthermore, such attacks are easier to implement using cheap software-defined radios (SDRs) by manipulating, or capturing and replaying overheard preambles.

To defend against PrInS attacks, we propose a novel approach for preamble customization and randomization using existing group keys and timestamps. We further develop preamble authentication in the receive state machine to detect and filter out maliciously injected preambles. Our defense mechanism is compatible with all OFDM-based Wi-Fi systems and does not impact the synchronization and frame error rate of the Wi-Fi system. Our simulation results show that this defense mechanism detects forged preambles with nearly 100% probability while guaranteeing low false alarm rates in most scenarios.

The main contributions of this paper are as follows:

- We investigate the inherent vulnerability of Wi-Fi systems to PrInS attacks due to insecure preamble, receive state machine exploitation, and capture effect;
- We introduce three PrInS attacks and study their impact on Wi-Fi devices;
- We conduct extensive SDR experiments and simulations to demonstrate the efficacy and power efficiency of the proposed attacks;
- We propose a defense mechanism based on customization, randomization, and authentication of the preamble, as well as an enhanced receive state machine.
- We demonstrate the effectiveness of the defense mechanism through theoretical analysis and simulations.

2 Preliminaries

2.1 Frame Preamble

In OFDM-based Wi-Fi standards (802.11a/n/ac/ax/be), every Wi-Fi frame starts with a legacy (802.11a) preamble that has three fields: Legacy Short Training Field (L-STF), Legacy Long Training Field (L-LTF), and Legacy SIG (L-SIG) field. The first two fields are fixed waveforms that are used for frame detection, synchronization, and channel estimation. The L-SIG field is mainly used for two purposes. First, it signals the amount of time allocated for transmission over the channel. Second, it indicates the frame format and other information needed for frame decoding (e.g., rate and length). The information conveyed in the preamble is also used by neighboring devices to defer their own transmissions and automatically filter out unintended frames.

A Wi-Fi network often includes a heterogeneous mix of devices that conform to different Wi-Fi versions. Specifically, current access points (APs) can seamlessly serve 802.11a/n/ac/ax stations (STAs) in the 5 GHz band and 802.11 b/g/n/ax STAs in the 2.4 GHz band. To remain interoperable with older Wi-Fi standards, the preambles in recent Wi-Fi standards prepend a legacy (802.11a) preamble to newly added Training and SIG fields. For example, in the high throughput (HT) preamble (802.11n), in addition to the legacy preamble, HT-training and HT-SIG fields are introduced, as shown in Fig. 1. The HT-SIG field conveys the bandwidth, the modulation-and-coding scheme (MCS), and other necessary information for HT operation. Generally, the duration, content, and modulation of non-legacy SIG fields vary depending on the specific Wi-Fi standard.

2.2 PHY Carrier Sense and Receive Procedure

Carrier sense multiple access with collision avoidance (CSMA/CA) is the fundamental MAC mechanism used in Wi-Fi networks. According to this mechanism, a device that wants to transmit must first perform carrier sense (CS) to determine whether the channel is busy or idle. If the channel is idle, the device performs random backoff before transmitting. CSMA/CA uses both physical and virtual CS. Physical CS is formally known as clear channel assessment (CCA). It determines whether the channel is busy or idle based on energy detection (ED) and preamble detection (PD). Virtual CS, on the other hand, reserves the channel based on the network allocation vector (NAV) set by the Duration field in the MAC headers of frames such as the request-to-send

(RTS)/clear-to-send (CTS). To differentiate between the two CS mechanisms, Physical CS is commonly referred to as CS/CCA In our paper, we only consider CS/CCA.

A 20 MHz channel is determined to be busy during CS/CCA: (1) if ED detects any signal (Wi-Fi or non-Wi-Fi) whose power exceeds -62 dBm, or (2) if PD identifies a Wi-Fi preamble whose power exceeds -82 dBm. CS/CCA for a wider channel is executed on the primary and secondary channels separately, and the values of the corresponding ED and PD thresholds can be found in [18, §21.3.18.5]. The CS/CCA mechanism is augmented by predicting the duration of the frame from the Length and Rate fields in the L-SIG of a decoded preamble. This reserves the channel for the current frame even if the frame payload is corrupted, whereas NAV reserves the channel for subsequent frames.

A simplified state machine for the PHY receive (RX) procedure is shown in Fig. 2 (based on [18, Fig. 21-37]). Upon sensing a busy channel through CS/CCA, the receiver detects the SIG fields and determines the format of the detected frame. If the format is supported, the receiver proceeds to decode and check the content of the SIG fields. In this step, the SIG fields are also validated through even parity and cyclic redundancy check (CRC). If the SIG fields are valid and all the announced modes (e.g., multi-user) are supported, the receiver proceeds to set up the hardware accordingly to decode incoming symbols. The receiver shuts off its PHY processing at the end of the frame duration predicted from L-SIG. If no error is encountered, the receiver switches back to CS/CCA once reception ends.

However, there are certain scenarios in which the receiver has to prematurely terminate reception but still wait until the predicted frame duration has elapsed before returning to CS/CCA: (1) an unsupported format, (2) an unsupported mode, or (3) lost carrier. These scenarios, elucidated in Section 4, are at the core of the PrInS attacks presented in this paper.

Note that the procedure outlined in Fig. 2 is applicable to both collision-free and collision scenarios. However, in the event of a collision, the receiver sticks to the procedure to receive the first frame or restarts the procedure to receive the strongest frame that the receiver locks on to due to the capture effect (see Section 3.3).

3 Preamble Vulnerability

3.1 Weak Information Security

Despite its importance, information security of the Wi-Fi preamble is not adequate, which may compromise the functions of the preamble. First, Wi-Fi devices never check the authenticity of received preambles. Consequently, an adversary can deceive its neighbors by sending forged preambles with forged SIG fields appended to the publicly known training fields. Secondly, L-SIG is protected by extremely weak even-parity that only detects odd numbers of bit errors. As for non-legacy SIG fields, their 8- or 4-bit CRCs fail to detect error patterns that are multiples of the generator polynomial, and their 4-bit CRCs cannot detect any errors in the last two bits of the SIG field [19]. Such weak integrity protection opens the door for preamble forgery by flipping a few bits of eavesdropped SIG fields. Eavesdropping is quite feasible as the preamble is neither encrypted nor scrambled.

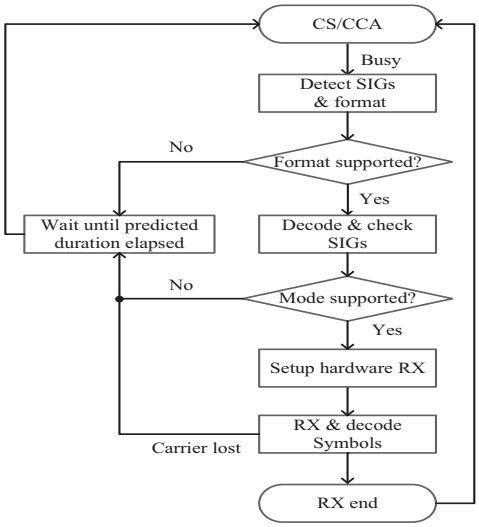


Fig. 2. PHY RX state machine in a standard-compliant Wi-Fi device.

To make matters worse, measurements in [19] show that some SIG fields are easily predictable. Prior knowledge gained via eavesdropping and prediction allows the forgery of dedicated preambles to intensify the effects of spoofing attacks.

3.2 Nonuniform Formats and Optional Modes

Wi-Fi standards support numerous preamble formats, which vary depending on the specific Wi-Fi version, the number of users, the number of antennas, etc. [20, §27]. A Wi-Fi device must detect the format of a received preamble, and hence the frame format, by examining the duration and modulation schemes of the SIG fields, as well as the value of the Length field in the L-SIG [20, Fig. 27-63]. Even if two preambles have the same format, they may announce in their SIG fields support for different optional modes, such as multiple-input-multiple-output (MIMO) and space–time block code (STBC). However, according to Wi-Fi standards [18, §19,§21] and industrial white papers [21], MIMO and STBC are *optional* in 802.11n/ac devices, especially non-AP STAs. Table 1 depicts the MIMO and STBC capabilities of examples of commercial Wi-Fi cards. Though some APs and laptops support both MIMO and STBC, some devices do not support them or only support MIMO without STBC. When format and mode variations occur in the same network (i.e., a so-called mixed network), mis-detection and false detection may occur [22], [23]. The probability of a detection error in the frame format increases dramatically if an adversary injects preambles of specific formats. Problems also arise if a device detects an unsupported frame format or an unsupported mode within the frame of a supported format. If that happens, the device will abort reception immediately, but will still wait for the predicted frame duration before initiating a new CS/CCA action (see Fig. 2).

3.3 Capture Effect

While Wi-Fi standards do not specify the receiver's actions following a collision [18, Fig. 21-37], due to the capture effect modern Wi-Fi devices often switch to decoding a stronger signal during the reception of a weaker one. In a

TABLE 1
MIMO and STBC capabilities in five commercial Wi-Fi cards.

Device	Wi-Fi card	Standards	MIMO	STBC
Linksys EA6350V3 AP	Atheros IPQ4018	a/b/g/n/ac	Yes	Yes
Dell Latitude	Intel N6300AGN	a/g/n	Yes	Yes
HP Pavilion	Intel AC3168	b/g/n/ac	No	No
iPhone8	Broadcom BCM4357	a/b/g/n/ac	Yes	No
Vivo X7	MediaTek MT6625	a/b/g/n	No	No

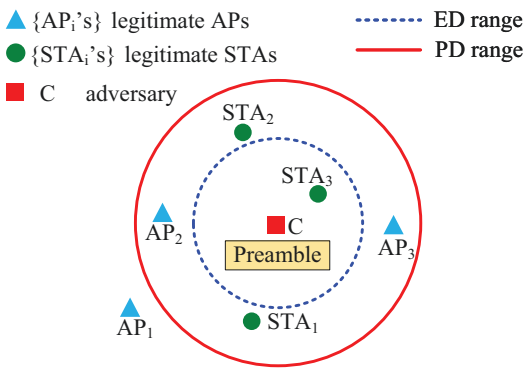


Fig. 3. Wi-Fi network with an adversary.

preamble/preamble collision scenario, if the second preamble arrives within a time offset less than a threshold, say Δt_{cap}, and has a power ratio higher than a threshold γ_{cap} compared to the first preamble, the receiver will abort the ongoing reception of the first preamble and will initiate the reception of the second preamble. Conversely, if the second preamble is insufficiently strong (i.e., power ratio lower than γ_{cap}) or arrives too late (i.e., time offset exceeds Δt_{cap}), the receiver will treat it as interference and continue receiving the first preamble. In this case, the receiver may still be able to correctly decode the first preamble if the signal-to-interference power ratio (SIR) exceeds a threshold γ_{dec}. Generally, $0 < \gamma_{dec} < \gamma_{cap}$ [24]. Because the preamble is critical for frame detection and synchronization, it is more susceptible to the capture effect than the payload. Indeed, experimental studies [3], [25], [26] show that many smartphone models that use Intel, Qualcomm, and Broadcom chipsets are all prone to the preamble capture effect, and some Qualcomm chipsets are even prone to the frame payload capture effect. Exploiting the capture effect, an adversary can gain a disruptive advantage.

4 PrInS Attacks

4.1 Overview

For illustration purposes, we consider the Wi-Fi network in Fig. 3, which depicts three legitimate AP-STA pairs, (AP$_i$, STA$_i$, $i = 1, 2, 3$) that operate in the presence of an adversary C. This adversary injects forged preambles with no payloads. She does not strictly adhere to the CSMA/CA procedure. Instead, She strategically injects preambles at specific times. APs/STAs within the PD range of C can detect her forged preamble unless they experience high interference or are transmitting (TX). As a result, such nodes can be spoofed to take wrong actions according to the injected preamble. Three different PrInS attacks can be launched by C, as shown in Fig. 4. The type of attack depends on the timing of injection and the relative power of the injected preamble. These attacks can silence the channel, mislead frame detection, falsify received data, and potentially drain the batteries of receiving devices. Though the injected preamble is short, its impact on the victims can last significantly longer than the duration of a legitimate frame.

4.2 Channel Silencing Attack

Fig. 5(a) depicts the channel silencing attack, where a forged preamble is injected but it does not collide with any legitimate frames. In this case, all legitimate APs/STAs in Fig. 3 except AP$_1$ (which is out of the PD range) detect the injected preamble. Due to backward compatibility, these APs/STAs will decode the legacy portion of the injected preamble correctly regardless of its format and will predict the frame duration from the Length and Rate field in the L-SIG. However, a receiving AP/STA will ultimately realize the absence of the payload carrier and report a PHY error code "Carrier Lost". Following the RX state machine, the AP/STA will wait for the anticipated duration of the nonexistent payload. Thus, an injected preamble, which could be as short as $20\,\mu s$, can reserve the channel for a maximum of $5.484\,ms$ (the longest duration of a PHY frame [18]). In addition, the AP/STA needs to wait for an extended inter-frame spacing (EIFS)[1] before trying to transmit again. To prolong the channel silencing, the adversary can announce the lowest rate and the largest payload length in the forged preamble. To make matters worse, if the announced bandwidth is wide, multiple Wi-Fi channels will be silenced, significantly hampering network throughput. Effectively, the preamble injection without collision maliciously silences the channel by deferring channel access of victim devices.

It is worth noting that a low-power injected preamble can still succeed as long as its power is above the PD threshold of the victim STAs. This is the main advantage of the channel silencing attack over conventional PHY-layer jamming attacks. Only $20\,\mu s$ (roughly 2 slots) are required to inject a forged preamble. Such a short duration makes it feasible to launch this attack between two Wi-Fi frames, which include various interframe spacings (IFSs) and random backoffs ranging from 0 to 1023 slots. Although we present the channel silencing attack in a collision-free setting, we later show in Section 4.5 that this attack can still be realized even if the injected preamble collides with one or multiple legitimate frames.

4.3 Frame Detection Attack

This attack involves a collision between the injected preamble and a legitimate frame. In contrast to the attacks in [11]

1. EIFS is the total duration of a short interframe spacing (SIFS), an acknowledgment (ACK) frame, and a distributed coordination function (DCF) interframe spacing (DIFS) shown in Fig. 4.

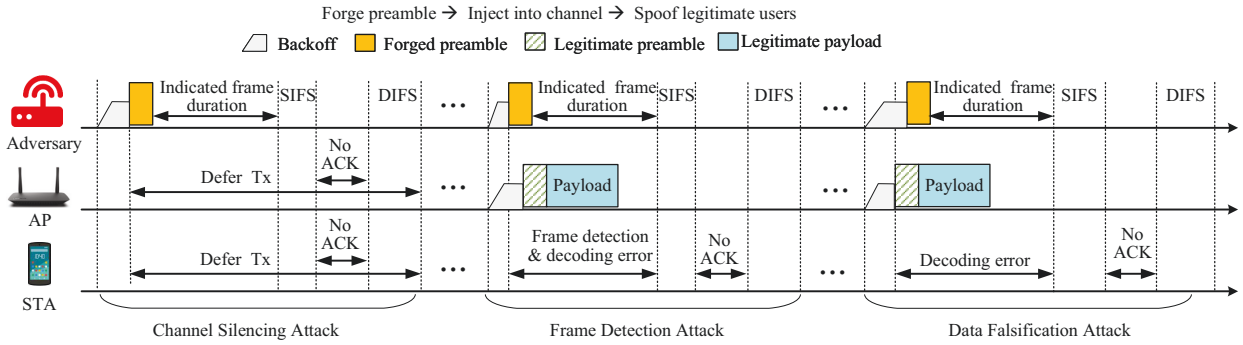
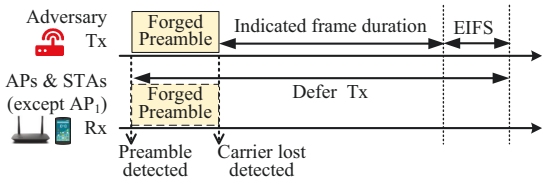
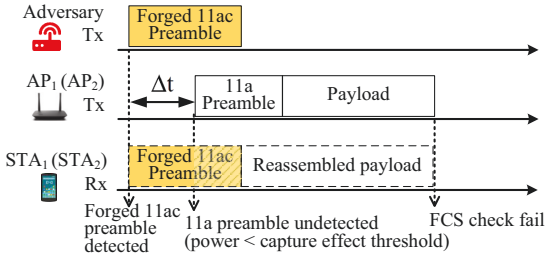


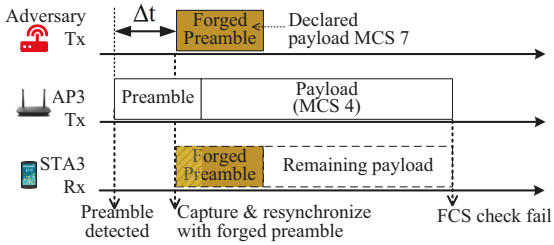
Fig. 4. Overview of three PrInS attacks in the downlink scenario (the attacks can also be performed during uplink transmissions). The indicated frame duration can be $10\times$ to $100\times$ of the legitimate frame duration.



(a) Channel silencing attack. Injected preamble does not collide with any legitimate frame.



(b) Frame detection attack. Injected preamble arrives first and collides with the legitimate frame.



(c) Data falsification attack. Injected preamble arrives last and collides with the legitimate frame.

Fig. 5. Details of PrInS attacks at different timings of injection and power levels (the darker the color of the injected preamble, the higher its power).

that also target a legitimate preamble to disrupt frame detection, our attack relies on frame format spoofing.

At first glance, it seems challenging to inject the forged preamble at a specific time just before the start of a legitimate frame, as devices are supposed to perform random backoff before any transmission. However, ACK, CTS, Response, and Data frames in a contention-free transmission opportunity (TXOP) are sent after a fixed duration from their preceding frames. Therefore, the start of such "follow-up" frames can be accurately predicted by the adversary. This allows the adversary to inject a forged preamble a few microseconds ahead of the legitimate frame to guarantee success.

Fig. 5(b) shows an injected forged 802.11ac preamble that arrives at STA_1 before the arrival of a legitimate 802.11a frame at STA_1 (also STA_2). This situation arises when the legitimate transmitter erroneously detects the channel as idle, which can occur in two possible scenarios: (1) for STA_1, the legitimate transmitter AP_1 is outside the PD range of C; (2) for STA_2, the legitimate transmitter AP_2 is busy in RX state and, therefore, does not invoke PD while at the same time, the power of the injected preamble is below the ED threshold. Here, we assume that STA_1 and STA_2 support the 802.11ac standard and are backward-compatible with the 802.11a standard. We denote the time offset between the injected preamble and the legitimate preamble as Δt, the signal-to-jamming power ratio (SJR) in dB as γ, and the respective duration of the legitimate and injected preambles as T_p and T'_p. A successful frame detection attack requires $-T'_p < \Delta t < 0$ and $\gamma < \gamma_{cap}$. Because the victim receivers (i.e., STA_1 and STA_2) are already in the process of decoding the forged 802.11ac preamble, they will not detect the legitimate 802.11a preamble as the SJR γ is smaller than γ_{cap} of the capture effect. Since $\gamma_{cap} > 0$, such an attack can potentially succeed with $\gamma > 0$. In this case, the effectiveness of the attack could be compromised as the injected preamble is not fully decodable due to the interference from the stronger legitimate frame. If the injected preamble is sufficiently strong such that $\gamma < -\gamma_{dec} < 0$, its decoding will not be impacted much by interference from the legitimate frame. Consequently, the legitimate 802.11a frame will be mis-detected as an 802.11ac frame at incorrect timing. Then, STA_1 (also STA_2) will mistakenly reassemble the non-overlapping portion of the legitimate frame as payload data for the forged preamble and decode it as an 802.11ac frame. Eventually, the frame check sequence (FCS) will declare a decoding error. A victim STA will start its EIFS timer at the end of the payload (Fig. 5(b)) or when the frame duration indicated in the forged preamble has elapsed (see the example in Fig. 4), whichever comes later. So a victim STA may defer channel access for a longer time than in the case of the frame detection attacks in [11].

4.4 Data Falsification Attack

Fig. 5(c) illustrates the other collision case, where the injected preamble arrives at STA_3 during the reception of a legitimate frame from AP_3, i.e., $0 < \Delta t < \Delta t_{cap} \leq T_p$. There is still a chance for the adversary to succeed because of the capture effect. More specifically, STA_3 will switch

to synchronize with and decode the injected preamble if the injected preamble overpowers the legitimate one and $\gamma < -\gamma_{cap}$. Note that the adversary does not necessarily need a higher transmitting power than AP_3 to launch the attack if she is much closer to STA_3 than AP_3 is. Because $-\gamma > \gamma_{cap} > \gamma_{dec}$, the strong forged preamble is decoded successfully despite the interference from the weak legitimate frame. As a result, STA_3 sets up the hardware according to the SIG fields of the forged preamble. The remaining portion of the legitimate payload will be received and decoded with incorrect parameters. Hence, the forged preamble manages to spoof STA_3 into receiving data with rogue signaling information (e.g., incorrect MCS). For example, STA_3 may decode the payload with MCS 7 (64-QAM at $3/4$ code rate) announced by the adversary, while the actual MCS is 4 (16-QAM at $1/2$ code rate) for the legitimate frame.

Based on the above attack, the received data will be falsified and will ultimately not pass the FCS check, which in turn, leads to a high packet loss rate.

4.5 Discussions

4.5.1 Special Channel Silencing Attack

So far, we have assumed that the injected preamble can be completely decoded by the legitimate APs/STAs. But what if the injected preamble belongs to an unsupported format or announces in its SIG fields an unsupported mode?

Consider a scenario where the adversary injects an 802.11ac preamble into a network comprised solely of 802.11a-capable devices. According to Section 2.1, a legacy (802.11a) preamble is always prepended to the dedicated preamble fields of an 802.11n/ac/ax/be frame. Therefore, an 802.11a device can still detect and decode the legacy portion of the injected 802.11ac preamble. Nevertheless, such a device lacks the capability to decode the non-legacy portion of the preamble. In such instances, the device would report an unsupported frame format. Similarly, the adversary can inject an 802.11ac preamble indicating STBC mode, when the legitimate receiving 802.11ac devices do not support this mode. Although legitimate devices could decode the entire preamble correctly, they have to report an unsupported mode and terminate their reception. Upon detecting an unsupported format or mode, the victim device does not immediately transition to the CS/CCA state. Instead, it recognizes the presence of an 802.11 frame in the air, whose duration is derived from the L-SIG. So, it must wait until the end of this duration. As a result, the channel will be silenced for all victim devices within the vicinity of the adversary.

Most importantly, such a channel silencing attack occurs *irrespective of whether or not collisions occur* between the injected preamble and legitimate frames, provided that the legitimate devices lock on to the injected preamble. In other words, if a forged preamble, unsupported by the legitimate AP and STA, is utilized in any of the three scenarios shown in Fig. 4, as long as the associated timing and power requirements are met, all devices not in the TX state will defer their transmissions.

4.5.2 Impact on Packet Latency

By hindering channel access, the channel silencing attack significantly increases the latency of packets. Although the other two attacks are not aimed primarily at hindering channel access, they still produce a forged preamble that corrupts the targeted legitimate frame and extends the duration of its channel occupation to up to 5.484 ms, plus an EIFS (see Fig. 4), and a random backoff due to the collision. Additionally, the APP packet carried in the corrupted frame needs retransmissions. Eventually, the packet latency under such attacks is increased. The overall packet latency is further increased if a PrInS attack is persistently repeated. Due to its low power and short duration, the channel silencing attack is stealthy. Consequently, if the adversary manages to inject a preamble during idle periods, she can periodically (approximately every 5 ms) inject subsequent preambles without raising suspicion. As for the other two attacks, repeating the attack every few packets and alternating targets among different users can make these attacks more stealthy and sustainable. At the same time, these repeated attacks cause a backlog of packets at the queues of legitimate transmitters, prolonging the average packet latency and increasing the likelihood of packet loss due to buffer overflow. These intuitive conclusions will be corroborated later in our simulations.

4.5.3 Effectiveness in Complex Scenarios

Multi-user Collisions: It is possible for the injected preamble to collide with multiple legitimate preambles, although the likelihood of such occurrences is low given the short duration of the preamble. As long as none of the legitimate preambles is captured by the receiver, all three types of PrInS attacks can still succeed. We elaborate by considering three possible scenarios. If the injected preamble arrives first, the frame detection attack or the special channel silencing attack will succeed, provided that none of the subsequent legitimate preambles involved in the collision meet the capture effect thresholds (for both time offset and SJR) compared to the injected preamble. Conversely, if the injected preamble arrives last, the data classification attack or special channel silencing attack will succeed, provided that the injected preamble meets the capture effect thresholds relative to the preceding legitimate preambles. In a more complex scenario where the injected preamble arrives later than some legitimate preambles but earlier than others, one or a combination of the three attacks will succeed when both conditions for the previous two scenarios are satisfied. In any of the above three scenarios, even when the conditions for a successful PrInS attack are not fully met, an injected preamble that collides with legitimate preambles can still corrupt the legitimate frames if the interference is significant.

MU Operations: If the underlying Wi-Fi network supports multi-user (MU) operations, i.e., MIMO and/or orthogonal frequency-division multiple access (OFDMA), the PrInS attacks can still succeed and will impact multiple links simultaneously. In the following, we take the channel silencing attack as an example. Consider a downlink (DL) scenario where the AP sends a single DL-MU frame to multiple STAs. If a forged preamble is injected before the transmission of the DL-MU frame, both the AP and STAs will be silenced. If this DL-MU frame collides with the injected preamble, the special channel silencing attack outlined in Section 4.5.1 can be realized. In uplink (UL) scenarios, consider the transmis-

TABLE 2
Timing, power, and impact of PrInS attacks.

Attack	Timing	SJR (dB)	Error			
			Invalid SIG	Carrier Lost	FCS Failure	Format Violation
Channel Silencing Attack	Idle	$\gamma > 0$	$-$	$+$	$-$	$-$
Frame Detection Attack	$-T'_p < \Delta t < 0$	$\gamma < \gamma_{cap}$	$+$	$+$	$+$	$+$
Data Falsification Attack	$0 < \Delta t < \Delta t_{cap} \leq T_p$	$\gamma < -\gamma_{cap}$	$+$	$+$	$+$	$+$

sion of UL-MU frames from multiple STAs following a Trigger frame from the AP. If the adversary injects a preamble (of at least 20 µs) during the SIFS (16 µs) between the Trigger frame and UL-MU frames, then as long as STAs (whether or not they intend to transmit in this UL-MU schedule) can detect the injected preamble, they have to defer their transmissions. In other words, these STAs will be silenced. Note that STAs that are hidden from the adversary may proceed to transmit their UL-MU frames as scheduled. To ensure channel silencing, the adversary must strategically determine her location and transmit power so that all targets are within its coverage. For instance, a plausible approach for the adversary is to situate herself on a drone that enables her to achieve her coverage goal.

4.5.4 Comparison of PrInS Attacks

In all three PrInS attacks, the forged preambles can be based on the same Wi-Fi standard or different Wi-Fi standards. However, compared to the frame detection attack where the adversary can forge any SIG field in the injected preamble, the SIG fields of the injected preamble used in the other two PrInS attacks should be designed to guarantee the most harmful channel silencing and data falsification. In Table 2, we compare three PrInS attacks in terms of timing, power, and impact on other error metrics[2].

The channel silencing attack during idle intervals leads to a"Carrier Lost" error and requires the lowest power, which, according to our experimental results in Section 5.3.1 can be 30 dB lower than the legitimate signal. The second energy-efficient attack is the frame detection attack, for which $\gamma < \gamma_{cap}$, implying a positive γ is sufficient. The data falsification attack, which requires $\gamma < -\gamma_{cap} < 0$, consumes more energy than the other two attacks. However, under certain conditions, both the frame detection attack and data falsification attack may lead to errors other than the expected "FCS Failure". First, as shown in Fig. 4, the legitimate frame may end before the frame duration predicted from the SIG fields of the injected preamble, triggering a "Carrier Lost" error. Secondly, if the injected preamble is detected (which could be a mis-detection due to interference) to be a format unsupported by the victim device, the "Format Violation" error will be reported. Lastly, if the injected preamble is not sufficiently strong, i.e., $-\gamma < \gamma_{dec}$, then even though the attacked device can still capture the injected preamble, it will report the "Invalid SIG" error due to decoding errors in forged SIG fields caused by severe interference from the legitimate frame.

2. Per the standard [18], four possible error codes listed in Table 2 could be reported at reveiver's PHY during the reception of a frame.

Among these errors, an "Invalid SIG" error results in terminating frame reception and immediately switching back to CS/CCA. This is less problematic than the other three types of errors. A "Carrier Lost" or "Format Violation" error reported during the reception of the injected preamble can silence the channel for a while but not trigger the reception of the payload at the victim devices. The "FCS Failure" or "Carrier Lost" error can be reported by the victim device after receiving and incorrectly decoding the non-overlapping portion of the legitimate frame. Because demodulation and decoding consume relatively high power, the PrInS attacks that lead to these two errors can cause rapid battery depletion.

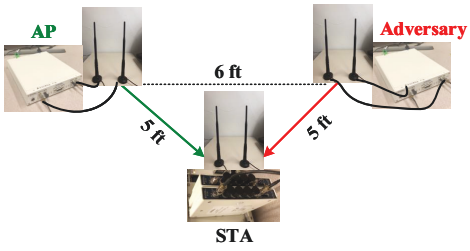
5 Evaluation of PrInS Attacks

5.1 Evaluation Metrics

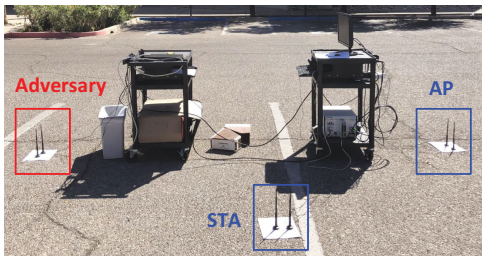
Our key metric for assessing the impact of PrInS attacks is the *throughput ratio*, defined as the ratio of the throughput when such an attack is present to the throughput when there is no attack. We evaluate this ratio at various SJRs, which reflect the energy efficiency of our attacks. Additionally, the *frame error rate (FER)* is used to determine the percentage of frames received with errors. We also measure the *latency* from the time an application (APP) packet is generated until its delivery to the receiver. Besides, we assess the *packet loss ratio*, which is the percentage of APP packets that overflow in the STAs' queues.

5.2 Experimental Setup

Due to the infeasibility of manipulating the PHY layer of commercial Wi-Fi cards, we conduct our experiments using the standard-compliant National Instruments (NI) LabVIEW 802.11 Application Framework [27]. This SDR platform implements both 802.11a (legacy) and 802.11ac standards, with a CS/CCA mechanism slightly different from the standard one in Section 2.2. Specifically, it reports a busy channel upon detecting a Wi-Fi preamble without strictly checking the -82 dBm PD threshold. Moreover, it allows the ED threshold for CCA to be configured at a higher value than the noise floor. As a result, in a CS/CCA process, a busy channel is mostly determined by PD unless there is a non-Wi-Fi transmission. In our indoor and outdoor experiments, we first identify an interference-free channel. We then measure the noise floor and preamble power at each device (i.e., AP, STA, adversary) and set a common ED threshold around the average preamble power. Due to differences in channel conditions among various setups, the ED threshold setting ranges from -65 to -60 dBm. Note that the CS/CCA mechanism may not operate effectively



(a) Indoor.



(b) Outdoor.

Fig. 6. Experimental setup with SDRs and antennas.

during attacks in Section 5.3.2 and 5.3.3. To force collisions of preambles in these attack experiments, we disable CCA by blocking the AP and the adversary and prevent doubling of the contention window. Yet, they still can execute CS/CCA effectively in response to the STA's transmissions.

We use NI USRP-2944 SDRs to emulate one legitimate AP and one adversary. One STA is implemented on an NI FlexRIO 7975 platform with an NI 5791 adaptor module. Our experiments are conducted in both realistic indoor and outdoor settings, as depicted in Fig. 6. In the indoor setup, the AP and the adversary are placed 6 ft apart from each other, and both are placed 5 ft from the STA. All these distances are tripled for the outdoor setup. For simplicity, devices are configured to operate in a SISO mode. The AP and STA run the Application Framework to exchange downlink data and uplink ACK packets. Their communication is attacked by the adversary, who injects forged preambles. We configure the AP to transmit 802.11a/ac data packets of fixed length (1024 bytes) and MCS 4 (16-QAM with $1/2$ code rate). Two packet generation rates (λ) are used: $\lambda = 2000$ packets/s (light load) and $\lambda = 10000$ packets/s (heavy load). The adversary is configured to transmit preambles with specifically manipulated SIG fields. The channel is centered at 2.457 GHz, with a bandwidth of 20 MHz.

5.3 Effectiveness and Efficiency of PrInS Attacks

5.3.1 Channel Silencing Attack

We implement the channel silencing attack in the indoor setting shown in Fig. 6(a). To silence the channel, the adversary is configured to inject 802.11a preambles at a target rate of 1000 preambles/s. These preambles are sent without payloads, so no ACK frames will be returned to the adversary by the STA. To avoid a dramatic increase in the adversary's contention window, we fix its backoff period for channel access to 8 slots. This value is the mean of the random backoff of legitimate users whose initial contention window size is $CW_{\min} = 15$. Because of the short preamble duration (20 µs) and the low preamble injection rate, the probability of a collision between an injected preamble and a legitimate frame is very small. Thus, the contention window size of the legitimate user remains $CW_{\min}$ almost always despite the attack. This setup guarantees fair channel access. To exclude the impact of MAC layer contention resolution mechanisms, the RTS/CTS and retransmissions are disabled on all three devices. The log file in Fig. 7 confirms that the perceived frame length at the PHY layer of the STA is quite large (4028 bytes) even though no frame is actually being detected (i.e., frame body length is 0) by the MAC layer of the same STA. This demonstrates that the channel silencing attack is not caused by a collision (otherwise, the log entries for the PHY and MAC will be consistent). We also notice that the MAC layer reports a "FCS check fail" error around 1.34 ms (anticipated frame duration) after the PHY RX start, which is much longer than the preamble duration (20 µs). We proceed to evaluate the impact of the declared frame length (payload size) in the forged preamble (denoted by L_{forged} when SJR = 0 dB and the announced MCS index is 4 (same as the actual value in the legitimate frame). Fig. 8(a) shows that as L_{forged} increases from 0 to 4000 bytes, the throughput ratio decreases from $76\% \sim 86\%$ to $10\% \sim 20\%$. Although a large L_{forged} should be announced to achieve an effective channel silencing attack, we intentionally add the special case of $L_{\text{forged}} = 0$ to show that the throughput reduction is not caused by the wasted channel time taken by the injected preamble itself. Rather, the throughput reduction is proportional to the air time reserved by the attacker's announced L_{forged}. Expectedly, the throughput reduction at heavy load is more severe than at light load. Both legacy (802.11a) and 802.11ac STAs are impacted by the forged 802.11a preamble, but the throughput ratio is worse (smaller) for an 802.11 ac STA when L_{forged} is large. This suggests that non-legacy traffic that utilizes aggregations is more susceptible to the channel silencing attack.

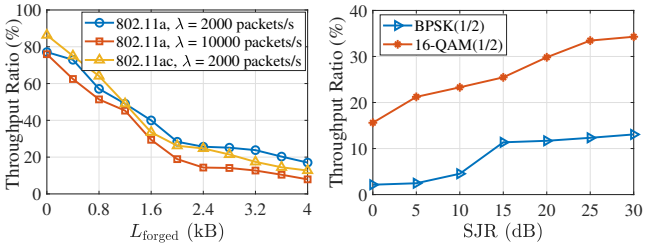
Next, we study the impact of the MCS indicated in the forged preamble. For this experiment, we let $L_{\text{forged}} = 4000$ bytes and consider light load ($\lambda = 2000$ packets/s). Recall that the MCS index for the legitimate frame is set to 4. As shown in Table 3, the lower the MCS index declared in the forged preamble, the lower the throughput ratio. For instance, when the adversary declares MCS for her fictitious payload as BPSK at $1/2$ code rate, the throughput ratio is about 2%. A similar low ratio is also observed when the injected preamble is unsupported by legitimate devices (see Table 4). In the first attack, the adversary randomly injects 802.11ac preambles that are of unsupported formats by the 802.11a AP and STA. Fig. 9 presents the log of an attacked 802.11a STA. The first three log entries are PHY RX start, PHY RX end, and MAC RX indications for a successfully received frame. Since the PHY RX start indication is issued only when the received preamble is valid [18, Fig. 17-19] [27], the STA does not issue such an indication in the fourth entry as it cannot validate the unsupported 802.11ac preamble. Instead, the PHY layer of the STA immediately issues an RX end indication and reports a "FormatViolation" error. So the MAC RX is not triggered at all. However, the STA still defers channel access until the expiration of the announced frame duration. As shown in Table 4, launching this attack at a rate of 2000 preambles/s effectively brings the STA's throughput down to 2.35% of its normal through-

```
5:58,805.3519 [INF]   FPGA-PHY RX start indication PhyRxStart.ind format: NON_HT_OFDM, bandwidth: 20 MHz, MCS: 4, PSDU length: 4028
5:58,805.3720 [INF]   FPGA-PHY RX end indication PhyRxEnd.ind RX error: CarrierLost, format: NON_HT_OFDM, bandwidth: 20 MHz, MCS: 4,
                      PSDU length: 4028, non HT bandwidth: 20 MHz, dynamic bandwidth support: 0
5:58,806.6971 [INF]   FPGA-MAC MPDU RX indication MAC RX indication type: Management, subtype: Association request, to DS: 0, from DS: 0,
                      HT: 0, sequence nr.: 0, fragment nr.: 0, MPDU frame body length: 0, A-MPDU De-Aggregation successful, FCS check fail,
                      MPDU length check not executed, Disassembly not executed, recipient address :  0:xx:xx:xx: 0: 0
```

Fig. 7. PHY and MAC logs of a legitimate STA when receiving a forged preamble that declares $L_{\text{forged}} = 4000$ bytes. The PSDU length of 4028 bytes includes the 28 bytes for a MAC header in addition to L_{forged}.

TABLE 3
Throughput ratio vs. attacker's announced MCS index under the channel silencing attack, $L_{\text{forged}} = 4000$ bytes, $\lambda = 2000$ packets/s, SJR $= 0$ dB, MCS index of the legitimate frame is 4.

MCS Index	0	1	2	3	4	5	6	7
Modulation	BPSK		QPSK		16-QAM		64-QAM	
Code Rate	1/2	3/4	1/2	3/4	1/2	3/4	2/3	3/4
Throughput Ratio	2.15%	4.39%	6.34%	11.43%	15.60%	24.39%	28.45%	34.94%



(a) (b)

Fig. 8. (a) Throughput ratio vs. announced frame length L_{forged} at SJR $=$ 0 dB; (b) Throughput ratio vs. SJR when $L_{\text{forged}} = 4000$ bytes with two different announced MCS's in the forged preamble.

TABLE 4
Throughput ratio under PrInS attacks using unsupported preambles.

SJR (dB)	Unsupported Format	Unsupported Mode
0	2.35%	2.94%
10	3.18%	3.27%

put. In the other attack, the forged preamble announces the STBC mode when the legitimate AP and STA do not support it. The extremely low throughput ratio caused by these two attacks is still observed even at SJR as high as 10 dB.

To quantify the energy efficiency of the channel silencing attack, we vary the transmit power for both the AP and the adversary to achieve an SJR in the range $[0, 30]$ dB. We set $\lambda = 2000$ packets/s and $L_{\text{forged}} = 4000$ bytes. The results in Fig. 8(b) demonstrate that even at SJR $= 30$ dB, the adversary brings down the throughput of a legitimate link to 13% of its normal throughput by announcing the lowest MCS (BPSK with $1/2$ code rate) in its forged preamble. When the injected preamble has the same MCS (16-QAM with $1/2$ code rate) as legitimate frames, the throughput ratio at 30 dB SJR is still low (34%).

5.3.2 Frame Detection Attack

As discussed before, the frame detection attack occurs when the AP and the adversary cannot sense each other. To study such an attack, we conduct outdoor experiments, where we physically ensure that the AP and adversary do not sense each other's transmission. This is done by inserting large metal objects between the two devices[3]. At the same time,

3. In an indoor setting, rich scattering makes it quite difficult to prevent the AP from sensing the adversary even when the line-of-sight (LOS) is blocked.

TABLE 5
Throughput ratio under the frame detection attack ($\Delta t < 0$).

SJR (dB)	$\lvert\Delta t\rvert$	
	9 µs (1 slot)	18 µs (2 slots)
0	9.96%	9.13%
10	72.49%	18.25%

we allow both devices to be sensed by the STA. To impose a negative time offset, i.e., $\Delta t < 0$ (see Fig. 5(b)), we fix the backoff duration at the adversary to 1 slot (i.e., 9 µs) while fixing the backoff duration at the AP to 2 or 3 slots. Both the AP and the adversary have saturated traffic: 1024-byte packets modulated by MCS 4 at the AP and preamble-only transmissions at the adversary. At SJR $= 0$ dB, the late-arriving legitimate preamble is undetectable by the STA because its power is not sufficient for a capture effect, resulting in a throughput ratio of around 9% for both values of $|\Delta t|$ (see Table 5). However, at 10 dB SJR, $|\Delta t| = 1$ slot (9 µs) leads to a throughput ratio of 72.49%, in contrast to 18.25% when $|\Delta t| = 2$ slots. This can be explained by the fact that in Wi-Fi systems, the STA performs frame detection and synchronization in the first 16 µs of a received preamble, so at a high SJR, legitimate frames that arrive during this period have a high chance of being recaptured by the receiver.

Link-level Simulation Setup: The LabVIEW Application Framework does not allow us to set Δt below 1 slot. Thus, we resort to link-level simulations to show the impact of small Δt values on the effectiveness of the frame detection attack. We implement the legitimate link and the attack based on a standard-compliant Wi-Fi toolbox [28]. To emulate the preamble capture effect in [24], [26], we set $\Delta t_{cap} = 16$ µs and $\gamma_{cap} = 6$ dB. As γ increases from 6 to 8 dB, the probability of a successful capture increases from 0.2 to 1. Both legitimate frames and forged preambles are in 802.11a format and have random MCSs (index 0 to 7) and frame lengths (400 to 1600 bytes).

As shown in Fig. 10, even when the injected preamble has significantly lower power than the legitimate one (e.g., SJR $= 8$ dB), as long as the injected preamble arrives 16 µs earlier (i.e., $\Delta_t < -16$ µs), the frame detection attack can cause an FER of 100%. Such high FER is consistently achieved, irrespective of Δ_t, when SJR $= 4$ dB. This is because the frame detection attack always succeeds when SJR

```
3:32,477.9262 [INF]   FPGA-PHY RX start indication PhyRxStart.ind format: NON_HT_OFDM, bandwidth: 20 MHz, MCS: 4, PSDU length: 1052
3:32,478.2793 [INF]   FPGA-PHY RX end indication PhyRxEnd.ind RX error: NoError, format: NON_HT_OFDM, bandwidth: 20 MHz, MCS: 4,
                      PSDU length: 1052, non HT bandwidth: 20 MHz, dynamic bandwidth support: 0
3:32,478.2794 [INF]   FPGA-MAC MPDU RX indication MAC RX indication type: Data, subtype: Data, to DS: 0, from DS: 0, HT: 0, sequence nr.: 2575,
Successful Data Frame fragment nr.: 0, MPDU frame body length: 1024, A-MPDU De-Aggregation successful, FCS check pass, MPDU length check pass,
                      Disassembly successful, recipient address : 46:xx:xx:xx:6D:62
3:32,479.8016 [INF]   FPGA-PHY RX end indication PhyRxEnd.ind RX error: FormatViolation, format: NON_HT_OFDM, bandwidth: 20 MHz, MCS: 0,
Unsupported preamble format PSDU length: 1052, non HT bandwidth: 20 MHz, dynamic bandwidth support: 0
```

Fig. 9. PHY and MAC logs of a legitimate STA when successfully receiving a data frame followed by a forged preamble of an unsupported format.

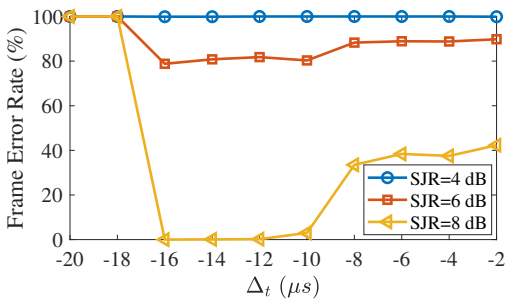


Fig. 10. Frame error rate vs. Δt in the frame detection attack.

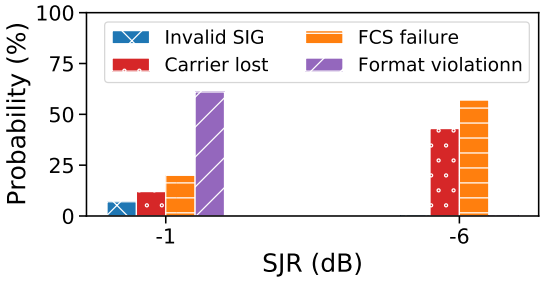


Fig. 11. Probability of different error types vs. SJR under the data falsification attack.

< 6 dB. However, an FER between 80% and 90% is observed at 6 dB SJR, as the legitimate frame may be captured by the receiver with a probability of 0.2. At an SJR of 8 dB, even though the legitimate frame meets the capture effect thresholds when $\Delta_t \geq -16\,\mu$s, the FER varies significantly. When $-16 \leq \Delta_t \leq -10\,\mu$s, the attack barely impacts the decoding and FER of legitimate frames because only part of the L-STF of the legitimate frame interferes with the injected preamble. But when the legitimate frame arrives within the first $10\,\mu$s of the injected preamble, several of its fields are corrupted by the injected preamble. This leads to errors in CFO estimation, channel estimation, and L-SIG decoding, hence, errors in decoded payload. In this case, the FER is around 40%.

5.3.3 *Data Falsification Attack*

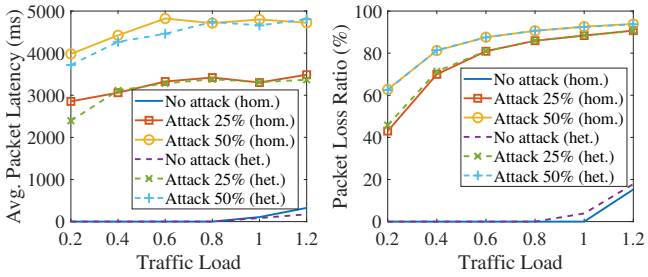
To study this attack, we use the same experimental setup used for the frame detection attack except that the forged preamble is injected $9\,\mu$s later than the legitimate frame and at much higher power (SJR $= -10$ dB). Such a setup allows the STA to capture the forged preamble during the reception of the legitimate preamble. The adversary announces MCS 3 (QPSK at $3/4$ code rate) in its preamble, whereas the actual frame sent by the AP uses MCS 4 (16-QAM at $1/2$ code rate). As a result, around 75% to 80% packets are decoded by the STA but with the wrong MCS. Such packets eventually fail the FCS check. The FER under this attack is close to 80%, which is much higher than the 10% FER required for reliable Wi-Fi communication [18]. On average, we observe a throughput ratio of 27.4% under this attack.

Furthermore, we examine the types of errors caused by the data falsification attack to elaborate its impact at different SJRs. We employ the same simulator and configurations as used in Section 5.3.2, with the exception that Δ_t is set within the range of 0 to 16 μs. As depicted in Fig. 11, all four error types are observed at an SJR of -1 dB. In this regime, the "Format Violation" error occurs with the highest probability, approximately 60%. At the lower SJR of -6 dB, the forged preamble is so strong that its SIG fields are valid and its format is correctly detected. Thus, there are no "Invalid SIG" or "Format Violation" errors, but only "Carrier Lost" and "FCS Failure" errors. Moreover, because random MCSs and lengths in the legitimate and forged preambles result in fewer scenarios where the estimated frame duration exceeds the actual frame duration, "Carrier Lost" occurs less frequently than "FCS Failure".

System-level Simulation Setup: To evaluate the impact of PrInS attacks on latency, we conduct *system-level* simulations, where we consider a Wi-Fi network that consists of one AP and four legitimate STAs with UL traffic. A standard-compliant 802.11ax implementation based on the example in [29] is used but modified to incorporate our proposed attacks. In this implementation, the PHY, MAC, and Application (APP) layers are abstracted. The Residential Path-loss Propagation channel model [30] is assumed. For simplicity, we fix the size and inter-packet times of generated APP packets at a given station. For all four STAs, we set the MAC payload size to 1500 bytes and the MCS index to 7 (64-QAM with code rate 5/6). The adversary launches a data falsification attack once every few legitimate frames using a forged preamble with an announced *Length* subfield in L-SIG of 65535 bytes[4]. We define the traffic load as the channel time allocated to a packet (including the PHY frame duration, contention time, SIFS, and ACK frame duration) divided by the time between the generation of two subsequent APP packets. We vary the total load from 0.2 to 1.2 by adjusting the APP-layer data rate. Both homogeneous and heterogeneous traffic scenarios are studied. For the homogeneous case, all STAs have the same load, which varies from 0.05 to 0.3 per STA. For the heterogeneous case, the loads for the four STAs are, respectively, 10%, 20%, 30%, and 40% of the total load. Details of the computation of the data rates for our simulations are provided in Appendix C.

As shown in Fig. 12(a), for both homogeneous and heterogeneous cases, the attack significantly increases packet latency even when only a fraction of the legitimate frames is being attacked. Under the attack, the average latency is higher than 2400 ms for all examined loads. As expected, the

4. In principle, if the same forged preamble is used in the frame detection attack, the packet latency will be comparable to that of the data falsification attack. For brevity, we do not include simulations for the frame detection attack.



(a) Average APP packet latency. (b) APP packet loss ratio.

Fig. 12. APP layer performance vs. total traffic load under no attack and under a data falsification attack for homogeneous (hom.) and heterogeneous (het.) traffic. When an attack is launched, the attacker targets a fraction (25% or 50%) of legitimate frames.

more persistent the attack, the higher the latency. Note that each simulation experiment is run for 10 sec. Extending the simulation time beyond 10 sec will result in higher latency. Such latency is unacceptable for time-critical applications. In fact and depending on the specific use case, mission-critical extended reality (XR) applications such as remote surgery and defense require a latency of one to tens of milliseconds [31]. In addition to increased latency, the attack causes a significant buildup in the packet queue at each STA, leading to a high buffer overflow rate (we observed packet loss rates from 43% to 94% when the buffer size was set to 256 packets).

6 Defense Mechanism

6.1 Challenges and Solution Space

The success of PrInS attacks hinges on the fact that the publicly known preamble sent by any device is trustworthy. An intuitive defense against such attacks would be based on authenticating the source device or the preamble.

6.1.1 Limitation of Existing Approaches

In [32], the authors proposed a frame-by-frame preamble authentication scheme to thwart relay and spoofing attacks against the Wi-Fi connection establishment. This scheme uses a hash-based message authentication code (HMAC) derived from the AP's MAC address, a symmetric key, and a sequence number (SN). The HMAC is partitioned into segments and integrated into the preambles of successive frames transmitted by the AP. The STA verifies the authenticity of a received preamble by comparing the received HMAC segment with the locally generated one.

However, extending this scheme to authenticate the preamble of a randomly received frame faces challenges. First, authenticating preambles from a diverse set of devices (including APs and STAs) using pairwise symmetric keys incurs significant overhead even when considering a modest number of devices. An alternative that uses a common shared key (such as the AP's public key or a group key) to generate an HMAC per beacon interval is susceptible to replay attacks. Secondly, the complexity of actual frame exchanges within Wi-Fi networks surpasses that of the connection establishment of a single link, where management frames follow a sequential order. In practice, many devices contend to transmit and certain devices may not be heard by all neighboring devices. Multi-user transmissions, retransmissions, and collisions further complicate frame exchanges. As a result, tracking the SN and HMAC segments, as required by the authentication scheme in [32], becomes extremely hard.

RF fingerprinting techniques, implemented with machine learning [33], [34] or without it [35], [36], have been explored for device authentication. These techniques have the potential to defend against PrInS attacks. However, such techniques are sensitive to channel impairments and are impractical in dynamic network topologies [37], [38]. They often require storing and updating pairwise RF fingerprints, including CFO and channel state information (CSI), between authenticated network devices. Therefore, they cannot authenticate devices joining the network for the first time and are only feasible when there are very few legitimate devices [37]. For a network of tens of devices, discerning unique RF fingerprints is challenging, and the associated storage and computational complexities become unmanageable [37], [38]. Furthermore, these techniques rely on the ability to identify the transmitter of a received signal. Achieving this necessitates the transmitter's identity or address obtained via prior control frames exchanges (e.g., RTS/CTS exchange), or decoding the current MAC header. Unfortunately, the RTS/CTS exchange is only enabled for large frames (the default threshold is 2347 bytes). The MAC header, which is not part of the injected preamble, is absent in collision-free PrInS attacks. For the two PrInS attacks that involve colliding with a legitimate frame, the MAC header of the legitimate frame does not appear at the expected location of the MAC header of the reassembled frame. Therefore, the victim device is unable to identify the source and authenticity of the received preamble.

Monitoring signal quality could potentially detect a data falsification attack. However, this approach could lead to false alarms. Such false alarms can occur when: (1) the transmitter is moving closer to the receiver, or (2) no attack is present but a capture effect takes place, wherein a stronger *legitimate* frame arrives in the middle of the first preamble. For instance, the first frame might originate from a distant device, followed by a stronger frame from a device closer to the receiver. If the receiver interprets the elevated received signal strength as a data falsification attack, it might erroneously discard the second frame.

Moreover, the above schemes cannot authenticate unknown devices in neighboring networks.

6.1.2 Solution Requirements

Above all, a feasible defense scheme against PrInS attacks should focus on authenticating preambles rather than devices in a Wi-Fi network. This authentication should then be used to enhance the RX state machine, mitigating the impact of PrInS attacks. Moreover, the scheme should be robust against the network dynamics in the number of devices, channel conditions, and Wi-Fi versions. Importantly, the complexity of storage, computation, and communication involved in the scheme should be manageable and practical.

Existing integrity checks in the SIG fields of the preamble can sometimes detect errors due to the collision of forged and legitimate preambles. However, such errors do not occur in the channel silencing attack or when $|\gamma| > \gamma_{dec}$

(let alone these integrity checks are weak as stated in Section 3.1). Therefore, a standard preamble is not suitable for authentication. A more effective approach would be to introduce a shared secret in the preamble so that all legitimate devices can verify its authenticity. Yet, securing each preamble with cryptography approaches (e.g., signature, HMAC) is infeasible due to the constraints on low computing resources and energy consumption at the PHY layer. A lightweight solution should directly discriminate between legitimate and forged preambles based on their distinct waveforms. This requires customization of the preamble waveform. Moreover, an adversary can record, modify, and/or replay eavesdropped legitimate preambles. To protect the preamble from replay attacks, the freshness of the customized preamble should be guaranteed.

Finally, any defense mechanism should be backward-compatible and not impact the critical functions of the preamble. The *extensible preamble modulation (eP-Mod)* scheme, proposed in [39], [40], can customize the STF fields of the preamble while maintaining its primary functions. The *eP-Mod* scheme can embed tens of bits into the STFs of a preamble with a probability of successful preamble demodulation (received error-free embedded bits) over 90%.

6.2 Preamble Authentication

To thwart PrInS attacks that exploit forged or replayed preambles, we propose to customize and randomize the STF fields of the preamble using *eP-Mod*, which embeds a seed for preamble authentication. The LTF and SIG fields remain intact so that channel estimation and PHY-layer signaling functions on devices equipped with our defense mechanism are not affected.

6.2.1 Preamble Customization and Randomization

For preamble customization, we first select an appropriate secret to ensure the authenticity of the preamble. Considering the broadcast nature of the preamble, in this paper, we use a beacon integrity group temporal key (BIGTK) because of its advantages which we will explain later. However, BIGTK is updated hourly. This time granularity is too large to combat preamble replay attacks. To address this, we additionally employ timestamps to randomize preambles over time. There is a readily available timestamp that indicates the time in microseconds since the AP has been active. This timestamp is obtained from the Beacon or Probe Response frame sent by the AP [18]. Wi-Fi devices synchronize their clocks to this timestamp every beacon interval of 100 time units (TUs) [41]. Given that one time unit is 1024 µs, a beacon interval is approximately 100 ms. For security purposes, it is better to set the preamble update interval to match the duration of the shortest frame, typically the ACK frame, which lasts for 30 µs to 100 µs, depending on the MCS index.

We propose to derive a pseudo-random sequence for every TU and segment it into multiple preamble seeds. Legitimate devices update the preamble every few tens of microseconds using a fresh seed to avoid replay attacks. The details of our proposed defense scheme are shown in Fig. 13. The AP broadcasts a beacon whose timestamp is t_k for the kth beacon interval and distributes the BIGTK along with other keys to the STA after MAC-layer authentication is

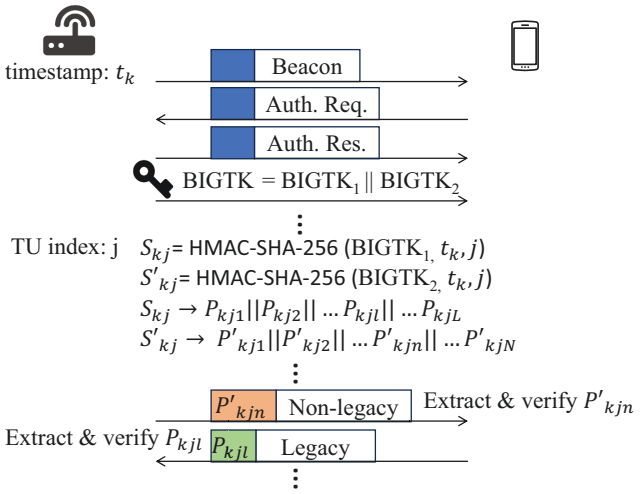


Fig. 13. Proposed defense scheme based on preamble authentication. The keys and preamble seeds are common among all legitimate devices.

TABLE 6
Parameters of preamble customization and randomization for legacy and non-legacy frames.

Frame type	key	Embedded bits/preamble	Update interval (µs)
Legacy	BIGTK_1	8	32
Non-legacy	BIGTK_2	16	64

completed. The root key, i.e., BIGTK, is split equally into two keys, BIGTK_1 and BIGTK_2, that are used separately to customize the preambles of legacy and non-legacy frames. At the beginning of the jth ($j \in \mathbb{Z}^+, 1 \leq j \leq 100$) TU within the kth beacon interval, both AP and STAs derive two pseudo-random sequences, S_{kj} and S'_{kj}, using a cryptographic primitive (e.g., HMAC-SHA-256) of Wi-Fi devices. Next, S_{kj} and S'_{kj} are respectively split equally into L segments P_{kjl}'s and N segments P'_{kjn}'s, $1 \leq l \leq L, 1 \leq n \leq N$. In our implementation, we use parameters listed in Table 6. A legacy frame embeds only an 8-bit preamble seed P_{kjl} in its L-STF, and a non-legacy frame embeds 8 bits of P'_{kjn} in its L-STF and 8 bits of P'_{kjn} in its non-legacy STF (e.g., HT-STF, VHT-STF, HE-STF). Given that the S_{kj} and S'_{kj} are 256-bit sequences generated by HMAC-SHA-256, L and N are 32 and 16, respectively. As a result, during every TU of 1024 µs, the STF of legacy frames can be updated 32 times, and the STFs of non-legacy frames can be updated 16 times. Thus, the preamble update interval for legacy and non-legacy frames are 32 µs and 64 µs, respectively.

The short preamble update intervals effectively prevent the adversary from replaying the preamble of a previous frame. To replay a legitimate preamble and pratially overlap with it, the adversary must first compensate for channel effects on L-STF and non-legacy STF with channel coefficients estimated from corresponding LTFs. As seen in Fig. 1, the adversary has at most the duration of one SIG field ($4 \sim 8$ µs) to replay the current preamble and overlap with it, which is infeasible considering typical processing times of channel compensation and RX-to-TX turnaround [27]. Furthermore, since the preamble seeds for legacy and non-legacy frames are different, the adversary cannot replay the legacy portion of a non-legacy preamble and overlap with

the tail of a non-legacy preamble to spoof legitimate users into thinking that there is an ongoing legacy frame.

We further consider two practical issues and propose our solutions to them.

What preambles should be used in frames sent before the establishment of BIGTK, and how to authenticate such preambles? An STA establishes a secure connection with the AP through a 4-way handshake [18, §4.10], during which the BIGTK is distributed in the third message. Consequently, pre-BIGTK frames (i.e., frames sent before the third message in a 4-way handshake) cannot customize their preamble as described above. Therefore, pre-BIGTK frames should still use the standard L-STF and non-legacy STF waveforms in their preambles. For instance, the beacon, authentication request (Auth. Req.), and authentication response (Auth. Res.) frames in Fig. 13 do not embed anything in the STFs of their preambles. Since standard STFs correspond to the preamble seed 0, we use seed 1 for preamble customization when encountering seed 0. This prevents PrInS attacks that use standard STFs in post-BIGTK frames. So 255 possible unique preamble seeds can be embedded in an STF. Nonetheless, we can still authenticate these preambles by the content of their SIG fields. [19], [42] show that SIG fields (length, rate, etc.) of pre-authentication frames are fixed for a given network. Therefore, any forged preambles with SIG fields different from the legitimate ones would be considered malicious and can be dropped.

How can an STA authenticate a preamble received from a neighboring BSS? Since the preamble is also used for PD-based carrier sensing, it is crucial that our design ensures all devices within the transmitter's PD range can authenticate the preamble. Therefore, for co-located Basic Service Sets (BSSs), the presence of a shared key is essential. One potential existing key that can be utilized is the BIGTK, which is common among co-located BSSs and employed for protecting beacons. If this key is unavailable, an alternative approach involves having the authenticator server orchestrate the negotiation of a shared symmetric key among physically proximate BSSs. This negotiated key can subsequently be distributed during the authentication process when devices join the network. Moreover, as the beacon timestamp differs across multiple BSSs, it should be mapped to the universally applicable Precision Time Protocol (PTP) [41] timestamp.

For each STF, the 8-bit preamble seed (P_{kjl} or half of P'_{kjn}) is embedded in a way that the first 4 bits (the Q-Seq) are mapped to a 16-DPSK symbol that controls the cyclic time-shift of the STF waveform, while the last 4 bits (the M-Seq) are mapped to a 16-PSK symbol that controls the phase shift of the STF waveform. Please refer to Section 3.2 of [40] for a detailed explanation.

6.2.2 Receive State Machine Enhancement with Preamble Authentication

The customized and randomized preamble allows the receiver to authenticate it before further processing. We propose to enhance the RX state machine as illustrated in Fig. 14. After detecting SIGs and frame format from a received preamble, the receiver should execute a few additional steps for preamble authentication.

The preamble demodulation (P-demod) is the most important step. For P-demod, the receiver first equalizes the

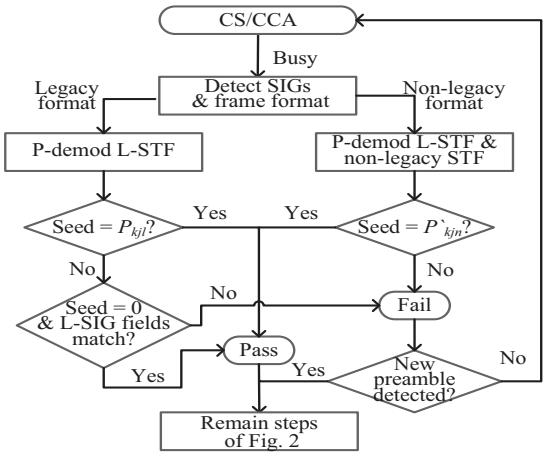


Fig. 14. First part of the PHY-layer RX state machine enhanced by proposed preamble authentication.

received STFs based on the estimated channel, then demodulates STFs, and finally extracts the preamble seeds embedded in them. According to [40], the extraction process essentially estimates the time and phase shifts from the equalized STF, mapping them to 16-DPSK and 16-PSK symbols, and finally demodulating them to bits. However, as 16-PSK is vulnerable to phase errors due to channel impairments, we set thresholds for phase errors during authentication. The default threshold is $\pi/32$, which is often used for 16-PSK. We can also relax this threshold to mitigate P-demod errors, hence reducing the false alarm rate (FAR). Furthermore, the FAR can be reduced by diversity gains achieved from space-time coding and multiple antennas [40], as well as error-correction code [32]. Detailed steps for preamble authentication are dependent on the frame format.

When a legacy preamble is detected, P-demod extra the preamble seed from the received L-STF. Because the update interval for the preamble seed is much smaller than the propagation delay (typically a few nanoseconds), the receiver's local-generated seed is the same as the one embedded in a legitimate frame. Therefore, the receiver first verifies whether the extracted preamble seed is the same as the local-generated one, i.e., P_{kjl}. If they are the same, the authentication passes. If the extracted seed is 0, the received frame could be a legitimate pre-BIGTK frame. Then, the receiver validates the values of the Rate and Length fields in L-SIG. If these values match common values or are within the ranges seen in the network, the authentication passes. Otherwise, the authentication fails.

When a non-legacy preamble is detected, the receiver extracts two halves of the preamble seed from the received L-STF and non-legacy STF, respectively. Then, the receiver concatenates the two halves to get the preamble seed. If it is the same as the local-generated one, i.e., P'_{kjn}, the authentication passes. Otherwise, the authentication fails.

Only the preambles that pass authentication are considered legitimate and can be accepted for further processing. Preambles that fail authentication are labeled as suspicious. To further reduce the FAR, the receiver can take into account the error code (see Table 2) sent to the MAC by the PHY layer. The finally failed preambles are dropped consequently. If no new preamble is detected in the remaining signal, the receiver will return to CS/CCA state immediately

to avoid unnecessary waiting. If the receiver detects a new preamble, it will recap the legitimate frame interfered with the early-arrived injected preamble. As long as the SJR is sufficient high for the data rate, the receiver can still decode the legitimate frame successfully. Even if decoding fails, early reporting of an error caused by the attack can save time for state transition.

6.3 Security Analysis

We assume that MAC-layer security is guaranteed, so the root key can be securely distributed to trusted parties. Thus, an external adversary $\mathcal{A}$ does not directly receive this key. Because both the root key and preamble seed are derived from strong pseudo-random functions (HMAC-SHA-256), it would be very difficult for $\mathcal{A}$ to extract them by observing the transmission. The only way for $\mathcal{A}$ to obtain the 8-bit preamble seed in a frame is to randomly guess it, excluding seed 0, which corresponds to the default preamble. Consequently, the success probability of $\mathcal{A}$'s guess is $p_g = \frac{1}{2^8-1} \approx 3.92 \times 10^{-3}$.

According to the analysis in [40], due to the repetition of STS's and the design of *eP-Mod*, the 16-DPSK and 16-PSK modulated symbols enjoy an extra SNR gain of 24.6 dB ($= 288$ in real value). Let σ be the real-valued *effective* SNR of a preamble symbol (including the 288 gain due to *eP-Mod*) and let σ^* be the SNR without the *eP-Mod* gain. Thus, $\sigma = 288\sigma^*$. The authentication procedure is considered successful only if the recovered preamble seed is error-free. Therefore, the probability that $\mathcal{A}$ passes the authentication, denoted as Pr[pass], depends on the error probability of decoding the 8-bit preamble seed under the *eP-Mod* scheme. This error probability can be approximated by the symbol error rates (SERs) for the 16-DPSK and 16-PSK modulation, which are provided in Appendix B. Unfortunately, there is no closed-form expression for the SER under either modulation scheme. Instead, we use approximate SER expressions for both to derive Pr[pass], assuming an AWGN channel. Note that our proposed authentication system is not limited to any specific channel model.

Setting $M = 16$ in (10) and incorporating the *additional* SNR gain of 288 due to *eP-Mod*, we obtain the error rate for the 4-bit M-Seq demodulated by 16-PSK:

$$P_m \simeq 2Q\left(\sqrt{576\sigma^*}\sin\left(\frac{\pi}{16}\right)\right). \tag{1}$$

Similarly, the error rate for the 4-bit Q-Seq demodulated by 16-DPSK can be obtained from (12):

$$P_q \simeq 4Q\left(\sqrt{576\sigma^*}\sin\left(\frac{\pi}{16}\right)\right). \tag{2}$$

Let ρ be the probability of successful preamble demodulation, which is the probability of an error-free preamble seed being demodulated by the receiver. This probability can be obtained as follows:

$$\rho = (1-P_m)(1-P_q) \tag{3a}$$

$$\simeq 1 - 6Q\left(\sqrt{576\sigma^*}\sin\left(\frac{\pi}{16}\right)\right) + 8\left(Q\left(\sqrt{576\sigma^*}\sin\left(\frac{\pi}{16}\right)\right)\right)^2 \tag{3b}$$

$$\simeq 1 - 6Q\left(\sqrt{576\sigma^*}\sin\left(\frac{\pi}{16}\right)\right). \tag{3c}$$

For $\sigma^* \geq 1$, the second term in the RHS of (3c) is much smaller than 10^{-6}, so ρ is very close to 1. This demonstrates the feasibility of our preamble authentication scheme. In the following, we use the metric ρ to analyze the PHY-layer security of our defense mechanism in various scenarios.

Scenario 1: In one scenario, the verifier $\mathcal{V}$ detects a preamble with no payload appended to it. This implies the possibility of a channel silencing attack. According to Fig. 14, $\mathcal{V}$ directly demodulates the whole received L-STF to extract the preamble seed for authentication. There are two cases that $\mathcal{A}$ passes the authentication: (1) $\mathcal{A}$ guesses the preamble seed correctly which is then recovered correctly from $\mathcal{A}$'s forged preamble by $\mathcal{V}$. (2) $\mathcal{A}$'s guess has $k(0 < k \leq 8)$ bits error, and then $\mathcal{V}$ flips exactly these error bits back to the legitimate one due to preamble demodulation errors. The probability of the first case is $p_g\rho$, while the probability of the second case can be approximated by $\sum_{k=1}^{8}\binom{8}{k}\frac{1}{2^8}(\frac{1-\rho}{8})^k(1-\frac{1-\rho}{8})^{8-k}$. Because ρ is close to 1, the second probability is negligible. Then, the probability that an adversary $\mathcal{A}$ passes the authentication is:

$$\Pr[\text{pass}] = p_g\rho = \frac{\rho}{2^8-1}. \tag{4}$$

Obviously, Pr[pass] is primarily determined by p_g. Indeed, our simulations in Section 6.4.2 demonstrate an FAR of $0 \sim 0.1$, which implies $0.9 \leq \rho \leq 1$. And experimental results in [40] have an average $\rho = 0.91$. In summary, both the experimental and simulation results demonstrate that in Scenario 1

$$0.353\% \leq \Pr[\text{pass}] \leq 0.392\%. \tag{5}$$

Scenario 2: In another scenario, $\mathcal{V}$ detects the payload along with the preamble. In this case, it is possible that the injected preamble overlaps with a legitimate one. Even worse, the overlap may be between the L-STF of the injected preamble and a legitimate preamble. According to (3c), ρ is a function of SNR σ^*. When authenticating the injected preamble by P-demod, the legitimate frame is the primary component of the noise. Therefore, we can replace SNR σ^* in (3c) with the inverse of SJR. Recall that γ_{dec} is the threshold on SJR that a preamble can still be successfully decoded under interference. Denote the time offset between the two preambles as Δt and the L-STS duration as T_s. Since we use γ in decibels for SJR throughout the paper, we denote its absolute value as Γ for convenience. We consider three cases for analysis:

(a) $\gamma < -\gamma_{dec}$: In this case, the injected preamble will be successfully demodulated regardless of Δt, even though part or all of it interferes with the weak legitimate preamble. If we replace σ^* with $1/\Gamma$, we will have a larger ρ because of negative γ. Using $\gamma = -\gamma_{dec} = -6$ dB obtained later in Section 6.4.2, we have $\sigma^* = 1/\Gamma = 10^{-\gamma/10} = 4$. Plug this σ^* in (3c), we get a smaller ρ. Therefore, the probability that $\mathcal{A}$ passes the authentication is much smaller than the one in (5).

(b) $\Delta t < -10T_s$: In this case, the injected preamble arrives first and its L-STF does not overlap with the later-arriving legitimate frame. Therefore, P-demod successfully extracts from the injected preamble a seed, which passes the authentication with the probability in (5).

(c) $0 > \gamma \geq -\gamma_{dec}$ and $-10T_s \leq \Delta t \leq -T_s$: In this case, the injected preamble arrives first, and at least one out

of its ten L-STSs does not overlap with the legitimate frame. When $|\gamma| < \gamma_{dec}$, it is infeasible to decode the preamble seed from the whole L-STF. Nonetheless, we can still extract the embedded preamble seed from the clean L-STSs. However, the extra SNR gain in 16-DPSK and 16-PSK modulated symbols is proportional to the number of L-STSs used for P-demod. In (3c), we replace σ^* with $1/\Gamma$ and apply the SNR gain decrease of $\min\{\lceil\frac{|\Delta t|}{T_s}\rceil, 10\}/10$. As a result, we get the rate of successful preamble demodulation in this scenario as

$$\rho' = 1 - 6Q\left(\sqrt{576\min\{\lceil\frac{|\Delta t|}{T_s}\rceil, 10\}/10\Gamma\sin\left(\frac{\pi}{16}\right)}\right). \quad (6)$$

Using 1 for the minimum term in (6) and $\gamma = 0$ dB, we get the lower boundary of ρ', which is 0.5839. Then, Pr[pass] can be approximated by

$$p_g\rho' + \sum_{k=1}^{8}\binom{8}{k}\frac{1}{2^8}(\frac{1-\rho'}{8})^k(1-\frac{1-\rho'}{8})^{8-k} \approx 0.37\%. \quad (7)$$

Using 10 for the minimum term in (6) and $\gamma = -\gamma_{dec} = -6$ dB, we get the upper bound of ρ', which is 1. Consequently, the probability that the adversary $\mathcal{A}$ passes the authentication is 0.

(d) In all other attack scenarios that involve collisions between legitimate and injected preambles, the adversary has no chance to pass the authentication due to interference from the legitimate frame.

In conclusion, under our proposed defense scheme, the adversary has a very low possibility ($< 0.4\%$) to spoof legitimate users.

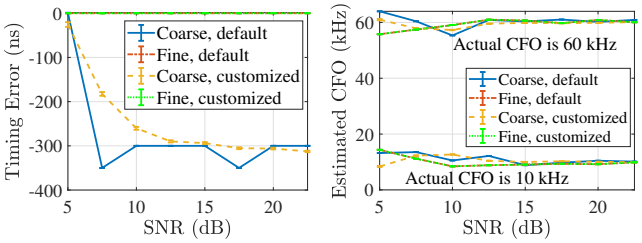
6.4 Prototype and Evaluation

6.4.1 Prototyping

Since modifying the lower-layer FPGA of the Application Framework is not trivial to prototype the defense mechanism, we implement it in Matlab. The IEEE 802.11a/ac SISO Wi-Fi links with 20 MHz bandwidth are built upon the WLAN Toolbox. WLAN channel model B (a typical indoor Rayleigh multipath channel) is used to emulate the real-world channel. As our defense uses existing keys and timestamps in any general Wi-Fi system, we do not implement such mechanisms in our prototype. Instead, we mainly focus on PHY-layer implementation, including the 8-bit *eP-Mod* for customizing STFs and the preamble authentication protocol adapted from [39] to incorporate threshold on the phase error.

6.4.2 Evaluation

We first study the impact of the proposed defense scheme under normal operation. In particular, we explore whether the customized preamble impacts the primary functions of the preamble, and consequently, the reception of the payload. Both theoretical analysis and experimental results in [39], [40] demonstrated that *eP-Mod* do not affect the primary functions of the preamble and therefore do not degrade the system performance. Note that the experimental results in [40] were obtained for a 10-bit *eP-Mod* scheme with $Q = 64$ and $M = 16$ ($\log_2 Q + \log_2 M = 10$). In contrast, a more robust 8-bit *eP-Mod* scheme is used in our defense



(a) Frame start timing accuracy. (b) CFO estimation accuracy.

Fig. 15. Average and 95% confidence interval of coarse and fine estimation of the frame start time and CFO at the receiver for both customized and default preambles (in most cases, the confidence interval is extremely tight).

mechanism. To demonstrate the impact of our defense, we simulate an attack-free Wi-Fi link with and without our defense scheme under WLAN Channel Model, we vary the SNR from 5 to 22.5 dB and adapt the MCS and length of the payload accordingly to optimize the data rate and satisfy a frame error rate (FER) requirement of 10%. We set the center frequency to 5.2 GHz and randomize the CFO in the range $[-80, 80]$ kHz (which is within the regulation of ± 20 ppm). At each SNR value, we transmit 10000 frames with payload size that varies between 300 and 1500 bytes. We evaluate the impact of the customized preamble on frame start time estimation, CFO estimation (at receiver), and FER. For frame start time and CFO, we consider both "coarse estimation" based on the STF as well as "fine estimation," which also uses the LTF. We compare these performance metrics to their counterparts under the standard (default) preamble.

As shown in Fig. 15(a), fine estimation of the frame start time is always accurate with or without our defense scheme. Our customized preamble outperforms the default preamble in the coarse estimation of the frame start time. Fig. 15(b) depicts the average and 95% confidence interval of coarse and fine estimated CFO. For better visualization, we only show the results when the actual CFO is 10 kHz and 60 kHz. The default and customized preambles have comparable performance in terms of fine CFO estimation. As for coarse CFO estimation, the customized preamble exhibits less deviation from the actual CFO. Since the default preamble has a fixed STF in each frame, it may not perform well for "coarse estimation" in different channel conditions. However, our customized preamble uses one of the 256 STF variants in a frame, so the estimation error is averaged out over various channels. In any case, the default LTF is kept intact, which fine-tunes coarse estimations of the frame start time and CFO. Therefore, there is little difference in the "fine estimations" with or without our defense scheme. As a result, at various MCS and SNR values, the FER is barely impacted by our defense scheme, as summarized in Table 7.

The second issue that we explore is the probability that a legitimate preamble fails authentication when no attack is present. We first consider the scenario where no collision occurs. As shown in Fig. 16(a), there is a small probability of false alarms when the SNR is below 10 dB. For instance, at an SNR of 5 dB, the FAR is only 0.1. The FAR is around 0.01 at SNR of 7.5 and 10 dB. To bring down the FAR, we also evaluate the performance at a relaxed threshold of $3\pi/80$. As a result, false alarms only occur with a rate of around

TABLE 7
Frame error rate with and without the proposed defense scheme when no attack is present.

SNR (dB)	5	7.5	10	12.5	15	20	22.5
Modulation	BPSK	QPSK		16-QAM		64-QAM	
Code rate	3/4	1/2	3/4	1/2	3/4	2/3	3/4
Payload length (bytes)	300	500	500	800	800	1500	1500
FER w/o. defense	1.4%	8.1%	0.7%	1.0%	10.6%	6.1%	1.1%
FER w. defense	1.2%	8.2%	1.2%	0.7%	10.1%	5%	1%

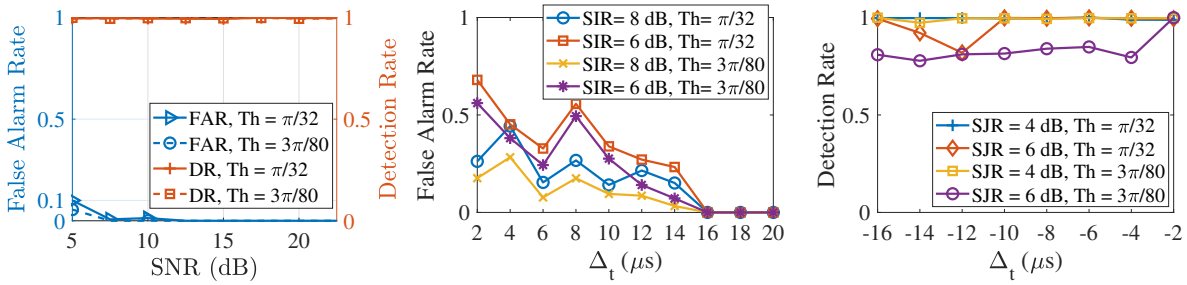


Fig. 16. Performance of the proposed defense scheme: (a) false alarm rate and detection rate vs. SNR when no collision occurs, (b) false alarm rate vs. Δ_t when two legitimate frames collide, (c) detection rate vs. Δ_t under the frame detection attack.

0.06 when the SNR is 5 dB.

Next, we consider more challenging scenarios where two legitimate frames collide. Based on our preliminary simulations, even for the most robust MCS, one of the two collided Wi-Fi frames can be successfully decoded only if the SIR is greater than 6 dB. So in Fig. 16(b), we evaluate the FAR at 6 dB and 8 dB with various time offsets Δ_t. Even with a relaxed threshold on the phase error, FAR still fluctuates around 0.25 when Δ_t is smaller than 16 µs. The fluctuation is mainly attributed to the periodicity of STF and LTF. If a non-integer number of STS repetitions (for the STF) or LTS repetitions (for the LTF) are interfered with due to collisions, channel estimation from the LTF and P-demod from the STF will both be affected unevenly on different subcarriers. Therefore, the performance of preamble authentication varies. However, as long as Δ_t is longer than the duration of the STF plus LTF, then the FAR goes to around 0.The corresponding FER is still close to 0 in all the scenarios of Fig. 16(b).

Next, we evaluate the detection rate (DR) of the proposed defense scheme. Considering the channel silencing attack, we vary the SNR of the forged preamble and plot the DR in Fig. 16(a) (see y-axis on the right). Our defense scheme can almost always detect a channel silencing attack. In the case of a frame detection attack, as seen from Fig. 16(c), DR is mostly close to 1 regardless of Δ_t even at SJR of 6 dB. However, with the relaxed threshold on phase error, DR is around 0.8 when SJR is 6 dB. Since the data falsification attack involves the capture effect, implying SJR < -7 dB, in our experiments, all the data falsification attacks can be successfully detected, i.e., DR = 1.

7 Related Work

DoS by Intelligent Jamming: Knowing the underlying Wi-Fi protocols, a sophisticated adversary can effectively launch intelligent jamming with little effort and low energy. For instance, in [43] an adversary can jam ACK frames or inject fake ACK frames after detecting preceding Data frames. However, jamming MAC frames requires decoding the MAC frame header to determine the jamming timing. In contrast, jamming PHY-layer signals, particularly in OFDM systems, is more straightforward. Legitimate transmissions can be disrupted by jamming the cyclic prefix [44] to cause inter-symbol-interference (ISI), or pilot tones [45] to distort channel estimation. Zhao *et al.* [7] introduced jamming tones with offsets to compromise the orthogonality of OFDM-based Wi-Fi systems. Nevertheless, these jamming signals deviate from standard compliance, often exhibiting high power concentration in time or frequency, making them readily detectable. In comparison, our PrInS attacks transmit standard-compliant signals with low power and duty cycle. Though interleaving jamming [46] was declared to be energy-efficient by jamming every three subcarriers for more than 2 OFDM symbols, its effectiveness is highly dependent on the MCS of the payload. Our PrInS attacks are effective irrespective of the payload properties. Besides, the above attacks cannot prolong the DoS duration beyond the impacted frame's duration, a notable capability exhibited by our PrInS attacks.

PHY-layer Spoofing: Unlike conventional identity spoofing on MAC and IP addresses, or information spoofing on broadcast frames such as beacon [5], PHY-layer spoofing manipulates the PHY-layer waveform. As we demonstrated in the introduction, attacks that manipulate Training fields [11], [12] are infeasible due to high energy consumption and/or stringent timing requirements. The `SigOver` attack [47] crafts messages that overshadow the legitimate broadcast LTE subframes to incur DoS and network downgrading. In addition to high power concerns, the crafted signal is lengthy and nontrivial to construct because it contains control information, reference signal, and data. The authors in [48] launched a similar attack to our channel silencing attack on several Wi-Fi chipsets (e.g., Intel AX200NGW, Atheros AR9271, Realtek RTL8192EU, etc.) that are often embedded in IoT devices. They found that most Wi-Fi

devices are significantly impacted by the attack, resulting in a packet loss ratio as high as 80% even at an SJR of 30 dB. Nevertheless, their attack only succeeded under light traffic conditions, which is a reasonable assumption for IoT networks but not typical Wi-Fi networks. In contrast, PrInS attacks remain effective even under heavy Wi-Fi traffic conditions since they can be executed during legitimate transmissions.

Detection and Mitigation: Although conventional DoS detection methods can raise alarms for potential PrInS attacks based on error rates, throughput [48], and delay analysis, several other attacks can inflict similar damage. Similarly, detecting preamble injection based on the ratio between correctly decoded frames relative to detected preamble counts, as shown in [48], is also unreliable as frame errors can also result from poor channel quality or other types of attacks. A legitimate user could not take the correct action to mitigate PrInS attacks. Crucially, because the above methods rely on long-term statistics, they are incapable of identifying a PrInS attack in real-time and mitigating its impact. In [48], the authors explored the timing and energy of RF signals to detect preamble injection. However, this approach is ineffective against our channel silencing and frame detection attacks where the power can be extremely low. Ramsey *et al.* [49] illustrated preamble manipulation as a means of fingerprinting and intrusion detection for Zigbee devices. Their approach was further expanded to Z-Wave devices by Hall *et al.* [50]. However, these methods are primarily suitable for authenticating known devices. They may lead to false alarms when dealing with new devices seeking to join the network or unknown devices in neighboring networks.

8 Conclusion and Future Work

In this paper, we demonstrated the susceptibility of Wi-Fi networks to PrInS attacks. which are based on forging and injecting preambles. PrInS attacks lead to channel silencing, frame mis-detection, data falsification, and battery depletion. To demonstrate the practicality and impact of such attacks, we conducted extensive analyses, SDR-based experiments, as well as link-level and system-level simulations. Our results show that legitimate users suffer significant reductions in the throughput and increments in the average packet latency and packet loss ratio even when only a fraction of frames are targeted or at a high SJR of 30 dB. We further proposed a backward-compatible defense scheme that customizes, randomizes, and then authenticates the preamble. We theoretically analyzed the security of our defense scheme, which proved in simulations nearly 100% accuracy in detecting PrInS attacks in most scenarios. Meanwhile, our defense scheme does not impact the synchronization and frame error rate of Wi-Fi systems.

Future Work. We will study PrInS attacks that exploit system-level information (e.g., resource allocation, spatial reuse parameters) conveyed in the SIG fields of 802.11ax/be preambles. The impact of such attacks can go beyond DoS and the PD range of the adversary. Accordingly, we will extend the defense scheme to take into account system-level information from higher layers.

Acknowledgment

This research was supported in part by NSF (grants CNS-1563655, CNS-1731164, and IIP-1822071) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

References

[1] Z. Zhang and M. Krunz, "Preamble injection and spoofing attacks in Wi-Fi networks," in *Proc. of the IEEE Global Comm. Conf.*, Madrid, Spain, Dec. 2021, pp. 1–6.

[2] T. Alsop. (2022, July) WLAN connected devices worldwide. [Online]. Available: https://www.statista.com/statistics/802706

[3] E. Khorov *et al.*, "Testbed to study the capture effect: Can we rely on this effect in modern Wi-Fi networks," in *Proc. of the IEEE Int. Black Sea Conf. on Commun. and Netw.*, Batumi, Georgia, June 2018, pp. 1–5.

[4] M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the dragonfly handshake of WPA3 and EAP-pwd," in *Proc. of the IEEE Symp. on Secur. and Privacy*, San Francisco, CA, USA, May 2020, pp. 517–533.

[5] M. Vanhoef, P. Adhikari, and C. Pöpper, "Protecting Wi-Fi beacons from outsider forgeries," in *Proc. of the ACM Conf. on Secur. and Privacy in Wireless and Mob. Netw.*, Linz, Austria, June 2020, p. 155–160.

[6] B. Bloessl, C. Sommer, F. Dressler, and D. Eckhoff, "The scrambler attack: A robust physical layer attack on location privacy in vehicular networks," in *Proc. of the Int. Conf. on Comput., Netw. and Commun.*, Anaheim, California, USA, Feb. 2015, pp. 395–400.

[7] S. Zhao, Z. Lu, Z. Luo, and Y. Liu, "Orthogonality-sabotaging attacks against OFDMA-based wireless networks," in *Proc. of the IEEE Conf. on Comput. Commun.*, Paris, France, May. 2019, pp. 1603–1611.

[8] E. Qi *et al.*, "Beacon protection," IEEE, Report doc.: IEEE 802.11-19/0314r2, Mar. 2019.

[9] D. S. Berger *et al.*, "Gaining insight on friendly jamming in a real-world IEEE 802.11 network," in *Proc. of the ACM Conf. on Secur. and Privacy in Wireless and Mob. Netw.*, Oxford, UK, July 2014, pp. 105–116.

[10] H. Rahbari and M. Krunz, "Secrecy beyond encryption: obfuscating transmission signatures in wireless communications," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 54–60, Dec. 2015.

[11] M. J. L. Pan, T. C. Clancy, and R. W. McGwier, "Jamming attacks against OFDM timing synchronization and signal acquisition," in *Proc. IEEE Mil. Commun. Conf.*, Orlando, FL, USA, Oct 2012, pp. 1–7.

[12] H. Rahbari, M. Krunz, and L. Lazos, "Swift jamming attack on frequency offset estimation: The achilles' heel of OFDM systems," *IEEE Trans. on Mob. Comput.*, vol. 15, no. 5, pp. 1264–1278, 2016.

[13] Wi-Fi connects providers with patients across a variety of environments. Wi-Fi Alliance. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/healthcare

[14] R. Smith, "5 things to know about DDoS attacks in healthcare," *Health Tech Magazine*, Sep 2021.

[15] Wi-Fi mesh for public safety. Strix Systems. [Online]. Available: http://www.strixsystems.com/cswifimeshforpublicsefety.aspx

[16] Wi-Fi 6/6E for industrial IoT. Wireless Broadband Alliance. [Online]. Available: https://wballiance.com/wi-fi-6-6e-for-industrial-iot/

[17] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, "Cyber threats to industrial IoT: A survey on attacks and countermeasures," *IoT*, vol. 2, no. 1, pp. 163–186, 2021.

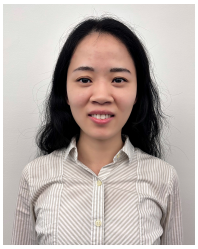
[18] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 2020.

[19] Z. Zhang and M. Krunz, "SIGTAM: A tampering attack on wi-fi preamble signaling and countermeasures," in *Proc. of the IEEE Conf. on Commun. and Netw. Security (CNS)*, Austin, TX, USA, Oct. 2022, pp. 1–9.

[20] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 1: Enhancements for High Efficiency WLAN*, IEEE Std. IEEE 802.11ax, 2021.

[21] "IEEE802.11ac: The next evolution of Wi-Fi standards," White Paper, Qualcomm, 2012.

[22] H. Zhang, S. Sirivasa, R. Banerjea *et al.*, "TGac preamble auto-detection comparisons," IEEE, Report doc.: IEEE 802.11-10/0549r2, May 2010.
[23] S. Moon, D. Lee, and M. Cheong, "Preamble auto-detection in 802.11ax," IEEE, Report doc.: IEEE 802.11-15/0360r1, Mar. 2015.
[24] WINLAB, "Physical layer: Frame capture effect implementation," Rutgers, The State University of New Jersey, Tech. Rep., 2016.
[25] J. Lee *et al.*, "An experimental study on the capture effect in 802.11a networks," in *Proc. of the ACM Int. Workshop on Wireless Netw. Testbeds, Exp. Evaluation and Characterization*, Sep. 2007, pp. 19–26.
[26] E. Endovitskiy, E. Khorov, A. Kureev, and I. Levitsky, "Demo: Experimental study of capture effect in smartphones and wi-fi access points," in *Proc. of the IEEE Wireless Commun. and Netw. Conf. Workshops (WCNCW)*, May 2020, pp. 1–2.
[27] *LabVIEW Communications 802.11 Application Framework 2.1*, National Instrument, 2018.
[28] Mathworks. (2021) Matlab WLAN toolbox. [Online]. Available: https://www.mathworks.com/help/wlan/
[29] MathWorks, "802.11ax multinode system-level simulation of residential scenario using MATLAB," Sep. 2021.
[30] S. Merlin, G. Barriac, H. Sampath *et al.*, "TGax Simulation Scenarios," IEEE, Report Doc. IEEE 802.11-14/0980r1, July 2015.
[31] F. Alriksson, D. H. Kang, C. Phillips *et al.*, "XR and 5G: Extended reality at scale with time-critical communication," *Ericsson Technology Review*, vol. Core RAN, Aug. 2021.
[32] N. Hoque and H. Rahbari, "Countering relay and spoofing attacks in the connection establishment phase of Wi-Fi systems," in *Proc. of the ACM Conf. on Security and Privacy in Wireless and Mob. Netw. (WiSec)*, Guildford, United Kingdom, May 2023, pp. 275–285.
[33] G. Li, J. Yu, Y. Xing, and A. Hu, "Location-invariant physical layer identification approach for Wi-Fi devices," *IEEE Access*, vol. 7, pp. 106 974–106 986, 2019.
[34] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust wireless fingerprinting via complex-valued neural networks," in *Proc. of the IEEE Global Commun. Conf.*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[35] P. Liu *et al.*, "Real-time identification of rogue Wi-Fi connections using environment-independent physical features," in *Proc. of the IEEE Conf. on Comput. Commun.*, Paris, France, April 2019, pp. 190–198.
[36] J. Hua *et al.*, "Accurate and efficient wireless device fingerprinting using channel state information," in *Proc. of the IEEE Conf. on Comput. Commun.*, Honolulu, HI, USA, April 2018, pp. 1700–1708.
[37] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surv. & Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.
[38] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the reliability of wireless fingerprinting using clock skews," in *Pro. of the ACM Conf. on Wireless Netw. Security*, Hoboken, NJ, USA, Mar. 2010, p. 169–174.
[39] Z. Zhang, H. Rahbari, and M. Krunz, "Expanding the role of preambles to support user-defined functionality in MIMO-based WLANs," in *Proc. IEEE Conf. on Comput. Commun.*, July 2020, pp. 1191–1200.
[40] Z. Zhang, H. Rahbari, and M. Krunz, "Adaptive preamble embedding with MIMO to support user-defined functionalities in WLANs," *IEEE Trans. on Mob. Comput. (TMC)*, vol. 22, no. 2, pp. 1–17, Feb. 2023.
[41] P. Chen and Z. Yang, "Understanding precision time protocol in today's Wi-Fi networks: A measurement study," in *USENIX Annual Tech. Conf.*, Jul. 2021, pp. 597–610.
[42] J. Sharp. (2023, Feb.) 802.11 frame types and formats. [Online]. Available: https://howiwifi.com/2020/07/13/802-11-frame-types-and-formats/
[43] J. Heo, J.-J. Kim, S. Bahk, and J. Paek, "Dodge-jam: Anti-jamming technique for low-power and lossy wireless networks," in *Proc. of the Annual IEEE Intl. Conf. on Sens., Commun., and Netw. (SECON)*, San Diego, USA, June 2017, pp. 1–9.
[44] J. A. Mahal, "Analysis of jamming-vulnerabilities of modern multi-carrier communication systems," Ph.D. dissertation, Virginia Tech, 2018.
[45] L. Zhang, F. Restuccia, T. Melodia, and S. M. Pudlewski, "Jam sessions: Analysis and experimental evaluation of advanced jamming attacks in MIMO networks," in *Proc. of the ACM IntL. Symp. on Mob. Ad Hoc Netw. and Comp.*, 2019, pp. 61–70.
[46] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Interleaving jamming in Wi-Fi networks," in *Proc. of the ACM Conf. on Secur.& Privacy in Wireless and Mob. Netw.*, Darmstadt, Germany, July 2016, pp. 31–42.
[47] H. Yang *et al.*, "Hiding in plain signal: Physical signal overshadowing attack on LTE," in *Proc. of the USENIX Security Symp.*, Santa Clara, CA, Aug. 2019, pp. 55–72.
[48] S. Gvozdenovic, J. K. Becker, J. Mikulskis, and D. Starobinski, "Truncate after preamble: PHY-based starvation attacks on IoT networks," in *Proc. of the ACM Conf. on Secur. and Privacy in Wireless and Mob. Netw. (WiSec)*, Linz, Austria, July 2020, pp. 89–98.
[49] B. W. Ramsey, B. E. Mullins, M. A. Temple, and M. R. Grimaila, "Wireless intrusion detection and device fingerprinting through preamble manipulation," *IEEE Trans. on Dependable and Secure Comput.*, vol. 12, no. 5, pp. 585–596, 2015.
[50] J. Hall, B. Ramsey, M. Rice, and T. Lacey, "Z-wave network reconnaissance and transceiver fingerprinting using software-defined radios," pp. 163–X, 2016.
[51] J. Lu, K. Letaief, J.-I. Chuang, and M. Liou, "M-PSK and M-QAM BER computation using signal-space concepts," *IEEE Trans. on Commun.*, vol. 47, no. 2, pp. 181–184, 1999.
[52] M. K. Simon and M.-S. Alouini, "Digital communications over fading channels," *IEEE Trans. on Info. Theory*, vol. 54, no. 7, pp. 3369–3370, 2008.



Zhengguang Zhang received the B.S. and M.S. degrees in Communication and Information Engineering from the University of Electronic Science and Technology of China, in 2014 and in 2017, respectively. She is currently a Ph.D. student in the ECE Department at the University of Arizona. Her research interests include PHY-MAC cross-layer design of WLAN, wireless and spectrum-sharing security, and artificial intelligence in wireless networking.

Marwan Krunz [S'93-M'95-SM'04-F'10] is a Regents Professor of electrical and computer engineering at the University of Arizona. He also holds a joint appointment as a professor of computer science. From 2015 to 2023, he was the Kenneth VonBehren Endowed Professor in ECE. Currently, he directs the Broadband Wireless Access and Applications Center (BWAC), a multi-university NSF/industry center that focuses on next-generation wireless technologies. He is also an Affiliated Faculty of the UA Cancer Center. Previously, he served as the Site Director for the Connection One Center. He served as the chief scientist for two startup companies that focus on 5G and beyond systems and machine learning for wireless communications. He has published more than 330 journal articles and peer-reviewed conference papers and is a named inventor on ten patents. His latest H-index is 62. His research interests include wireless communications and protocols, network security, and machine learning. He was an Arizona Engineering Faculty Fellow and an IEEE Communications Society Distinguished Lecturer. He received the NSF CAREER Award. He was the TPC Chair for several conferences and symposia, including INFOCOM'04, SECON'05, WoWMoM'06, and Hot Interconnects 9. He was a general chair for WiOpt'23, vice-chair for WiOpt'16, and the general co-chair for WiSec'12. He served as the Editor-in-Chief for the IEEE Transactions on Mobile Computing. He served as an editor for numerous IEEE journals.

APPENDIX A
NORMAL WI-FI OPERATION

To demonstrate the significance of PrInS attacks, we show in Fig. 17 the typical behavior of legitimate devices in the absence of an adversary. When the reception fails due to a collision, the channel is only reserved until the frame duration plus EIFS has elapsed. In contrast, as seen in Fig. 4, one forged preamble in a PrInS attack can reserve the channel for an extremely long announced frame duration plus EIFS.

APPENDIX B
SER OF M-PSK AND M-DPSK

According to Lu et al. [51], the bit error rate (BER) under M-PSK ($M > 16$) for a transmission over an AWGN channel with SNR of σ per symbol is:

$$P_b(\text{MPSK}) \simeq \frac{2}{\log_2 M} Q\left(\sqrt{2\sigma}\sin\left(\frac{\pi}{M}\right)\right) \tag{8}$$

where $Q(\cdot)$ is the complementary distribution of a standard normal random variable. From (8), it is straightforward to express the symbol error rate (SER) for M-PSK, denoted as $P_s(\text{MPSK})$:

$$P_s(\text{MPSK}) = 1 - (1 - P_b\,(\text{MPSK}))^{\log_2 M}. \tag{9}$$

At large σ and for $M > 4$, $P_b(\text{MPSK}) \ll 1$, as can be deduced from (8). Thus,

$$\begin{aligned} P_s(\text{MPSK}) &\simeq (\log_2 M) P_b(\text{MPSK}) \\ &= 2Q\left(\sqrt{2\sigma}\sin\left(\frac{\pi}{M}\right)\right) \end{aligned} \tag{10}$$

where in the second equality of (10), we substituted for the expression of $P_b(\text{MPSK})$ in (8).

Next, we consider M-DPSK. Similar to M-PSK, there is no closed-form expression for the SER. However, the SER under M-DPSK is approximately related to the SER under M-PSK as follows [52, Eq. 8.35]:

$$P_s(\text{M-DPSK}) \simeq 2P_s(\text{MPSK}) - P_s^2(\text{MPSK}). \tag{11}$$

Thus, the SER of M-DPSK symbol at large SNR and $M > 4$ can be approximated as:

$$\begin{aligned} P_s(\text{M-DPSK}) \simeq& 4Q\left(\sqrt{2\sigma}\sin\left(\frac{\pi}{M}\right)\right) - 4\left(Q\left(\sqrt{2\sigma}\sin\left(\frac{\pi}{M}\right)\right)\right)^2 \\ \simeq& 4Q\left(\sqrt{2\sigma}\sin\left(\frac{\pi}{M}\right)\right). \end{aligned} \tag{12}$$

APPENDIX C
TRAFFIC LOAD AND APP-LAYER DATA RATE

We define the traffic load as the actual channel time allocated to a packet (including the PHY frame duration, contention time, SIFS, and ACK frame duration) divided by the time interval between the generation of two subsequent APP packets. For an MCS index of 7, each OFDM symbol of 16 µs conveys 1210 bits. A MAC frame with a MAC header of 36 bytes has $8 \times (1500 + 36) = 12288$ bits, which are carried by $\lceil 12288/1210 \rceil = 11$ OFDM symbols spanning over $11 \times 16 = 176$ µs. Adding the 802.11ax PHY preamble of 56 µs, SIFS of 16 µs, ACK frames of 28 µs, average channel contention time of $0.5CW_{\min} = 0.5{\times}15{\times}9 = 68$ µs, one APP packet typically takes 344 µs to be transmitted. If the 1500-byte packet is generated at a data rate of r bps, the inter-packet interval would be $8 \times 1500/r$. Therefore, the traffic load of a STA is $344r/120000$. Then, it is straightforward to compute the data rates for each STA for various traffic loads in both homogeneous and heterogeneous cases. Table 8 and 9 are the configurations we used for our system-level simulations in Section 5.3.3.

TABLE 8
UL APP-layer data rate (in Mbps) of 4 STAs in the homogeneous case (MAC payload size $= 1500$ bytes, MCS index $= 7$)

Total load	0.2	0.4	0.6	0.8	1.0	1.2
per-STA load	0.05	0.1	0.15	0.2	0.25	0.3
data rate	1.72	3.45	5.17	6.90	8.62	10.34

TABLE 9
UL APP-layer data rate (in Mbps) of 4 STAs in the heterogeneous case (MAC payload size $= 1500$ bytes, MCS index $= 7$)

Total Load	Per-STA data rate (load)			
	STA_1	STA_2	STA_3	STA_4
0.2	0.69 (0.02)	1.38 (0.04)	2.07 (0.06)	2.76 (0.08)
0.4	1.38 (0.04)	2.76 (0.08)	4.14 (0.12)	5.52 (0.16)
0.6	2.07 (0.06)	4.14 (0.12)	6.21 (0.18)	8.28 (0.24)
0.8	2.76 (0.08)	5.52 (0.16)	8.27 (0.24)	11.03 (0.32)
1.0	3.45 (0.10)	6.90 (0.20)	10.34 (0.30)	13.79 (0.40)
1.2	4.14 (0.12)	8.28 (0.24)	12.41 (0.36)	16.55 (0.48)

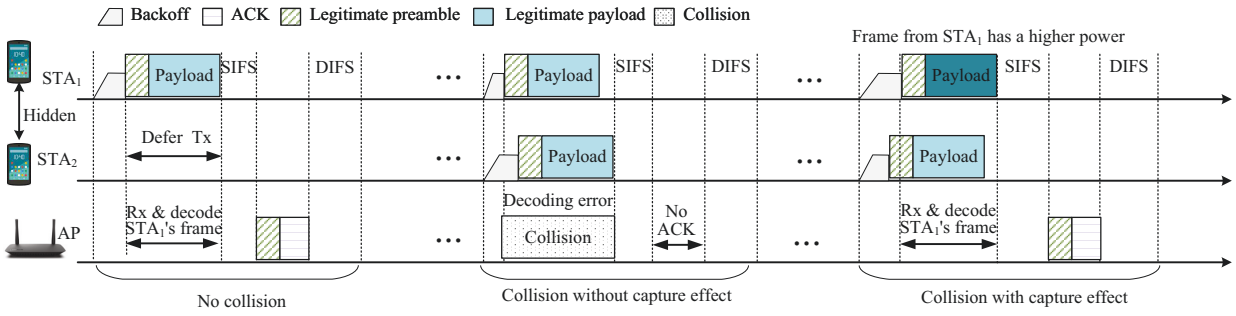


Fig. 17. Behaviors of legitimate devices in the absence of PrInS attacks.