Security Vulnerability and Countermeasures of Frequency Offset Correction in 802.11a Systems

Hanif Rahbari, Marwan Krunz, and Loukas Lazos
Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ

***Abstract*—Frequency offset (FO) is an inherent feature of wireless communications. It results from differences in the operating frequency of different radio oscillators. Failure to compensate for the FO may lead to a decoding failure, particularly in OFDM systems. IEEE 802.11a/g systems use a globally known preamble to deal with this issue. In this paper, we demonstrate how an adversary can exploit the structure and publicity of 802.11a's frame preamble to launch a low-power reactive jamming attack against the FO estimation mechanism. In this attack, the adversary will need to quickly detect a PHY frame and subsequently distort the FO estimation mechanism, irrespective of the channel conditions. By employing a fast frame detection technique, and optimizing the energy and structure of the jamming signal, we show the feasibility of such an attack. Furthermore, we propose some mitigation techniques and evaluate one of them through simulations and USRP testbed experimentation.**

I. Introduction

The effectiveness of a jamming attack is usually measured by its energy efficiency and the amount of disruption/damage it inflicts upon the system. These two parameters are often conflicting. For example, constant, deceptive, random, and other reactive jamming models achieve a high level of denial-of-service (DoS), but exhibit poor energy efficiency [8]. On the other hand, energy-efficient attacks such as selective jamming often rely on traffic analysis and protocol semantics [8]. These attacks fail to significantly corrupt ongoing transmissions if randomization, coding, and/or encryption techniques are used to hide the transmission features.

One potential method for achieving energy-efficient and highly disruptive jamming attack is to target the acquisition of important communication parameters, such as transmission timing and frequency offset (FO). Preventing correct estimation of these parameters can seriously jeopardize the reception process. Physical (PHY) layer standards usually employ globally known sequences (known as *preambles*) and signal estimation algorithms to acquire these critical communication parameters. An adversary may exploit the publicity of the preamble to construct a reactive-selective jamming attack against the receiver, with the objective of preventing correct estimation of such parameters. Among other functions, the preamble is used for frame detection, channel estimation, time synchronization with the transmitter, and FO estimation [1]. In this paper, we demonstrate the feasibility of a short-lived, energy-efficient attack against the FO estimation process of IEEE 802.11a devices (the ERP-OFDM mode in 802.11g and the non-HT format of 802.11n also have the same preamble structures as 802.11a). We particularly consider the commonly used method of Schmidl and Cox [10] for FO estimation as a representative scheme. Our study highlights the need to provide a more secure FO estimation mechanism.

This research was supported in part by NSF grant CNS−1016943 and ARO grant W911NF−13 − 1 − 0302. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

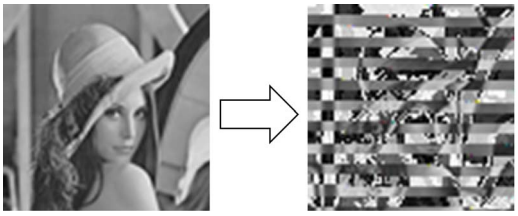


Fig. 1. Effect of uncompensated FO on a bitmap image over a noiseless channel (FO = 0.16% of the subcarrier spacing).

IEEE 802.11a/g systems are based on Orthogonal Frequency Division Multiplexing (OFDM). OFDM systems have recently been the subject of extensive security research (e.g., [2], [4], [7], [8]). La Pan *et al.* [6] demonstrated several jamming attacks against OFDM time synchronization, including barrage attacks, which is transmitting white noise to simply destroy the preamble, false preamble timing, and preamble nulling. In false preamble timing, the jammer forges a preamble to fool the receiver about the actual start time of the frame. A similar technique was used in [4] against an 802.11b receiver to significantly reduce the network throughput. Assuming a known transmitter-receiver channel, the preamble nulling attack tries to wipe out the preamble at the receiver.

More devastating than timing errors in OFDM systems are frequency synchronization errors [5]. Without frequency synchronization, the performance of OFDM systems degrades severely because of subcarriers' orthogonality violation, which creates inter-carrier interference (ICI) [5]. To illustrate the sensitivity of OFDM systems to FO estimation errors, consider two 802.11a radios that are tuned to the same target frequency. Their oscillators cannot be exactly aligned to that frequency due to hardware imperfections. FO is the difference between the actual frequencies of the two transceivers, usually normalized to the inter-subcarrier frequency interval, called *subcarrier spacing*. Figure 1 depicts the effect of a small FO estimation error (0.16%) on the transmission of a bitmap image (left image of Figure 1) using 64 subcarriers at the lowest data rate (6 Mbps). The received image (to the right) exhibits noticeable quality degradation in the form of image block misplacement. In practice, FO can be even larger than the subcarrier spacing.

Recently, La Pan *et al.* [7] presented two simple OFDM FO attacks (phase warping and differential scrambling) against the preamble structure proposed by Schmidl and Cox [10], which is different from the one used in 802.11 OFDM-

based standards. Gummadi *et al.* [4] showed the vulnerability of 802.11a clock (frequency) synchronization to a certain narrow-band jamming pattern that interferes with the entire preamble. Our work is different from [4] and [7] in that it focuses on *energy-efficient* jamming, i.e., only a small portion of the preamble is jammed. Furthermore, our work exposes and efficiently exploits the 802.11a's FO vulnerability for the first time. Our results can be extended to other OFDM-based standards, including 802.11g/n/ac, 802.16e/m (WiMAX), and LTE.

The contributions of this paper are as follows. We design an energy-efficient jamming signal that interferes with a small portion of the 802.11a preamble used for FO estimation. To do that, we tackle several challenges. First, the adversary (Eve) needs to estimate the FO between the transmitter (Alice) and receiver (Bob), and then quickly detect the transmission of a target frame. Second, the jamming sequence should delude Bob into estimating a wrong FO that is sufficiently far from the actual FO so that Bob decodes a corrupted frame. We derive the amount of FO estimation error needed to achieve erroneous decoding. Third, the jamming signal should be independent of the channel parameters (which are unknown to Eve). A *pairing* scheme is proposed to address this challenge. Fourth, the jamming attack should account for some frame detection errors. A *chaining* scheme is designed for this purpose. Subsequent to this attack, not only the channel estimation is corrupted at Bob, but also all the subcarriers are shifted forward or backward. Hence, the receiver will have a shifted version of the bitstream transmitted in an OFDM symbol. Combined with a faulty channel estimation and thus demodulation errors, the bits will be nonrecoverable. We further optimize the power of this jamming attack and experimentally evaluate it on a USRP testbed. The investigated attack is short-lived, lasting for only $2.4\ \mu s$ per PHY frame (12% of the PLCP sublayer duration and less than 0.5% of the MTU, transmitted at the highest data rate). Finally, we propose a few defense strategies and implement one of them (called sequence hopping).

The paper is organized as follows. In Section II, we provide background on frame detection, FO, and channel estimation in 802.11a systems. The system model and assumptions are given in Section III. The proposed attack and some possible remedies are presented and discussed in Sections IV and V, respectively. Section VI demonstrates the effectiveness of the attack through simulations and experiments.

II. Frame Detection and FO Correction

In OFDM, a bitstream is divided into several substreams, which are transmitted concurrently using a set of orthogonal frequency channels (subcarriers). For example, 802.11a defines 64 subcarriers with 312.5 kHz subcarrier spacing over 20 MHz of bandwidth. Only 48 of these subcarriers are used as data streams (user payload). ICI in OFDM systems creates significant BER distortion at the receiver [9]. To prevent ICI, the receiver estimates the FO using the PHY-layer preamble and adjusts the subcarriers to their intended frequencies. Another function of the preamble is to detect the beginning of a frame, enabling the receiver to decode the rest of the frame.

In 802.11a, the preamble consists of two parts (see Figure 2). The first part contains ten identical short training

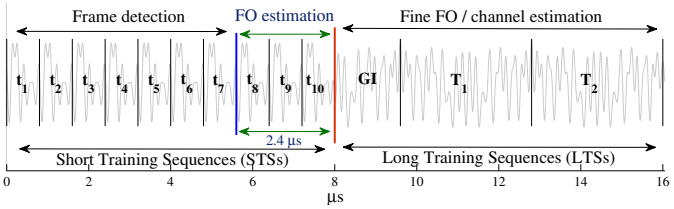


Fig. 2. The 802.11a preamble.

sequences (STSs), which represent ten cycles of a particular periodic function with period $\lambda_{STS} = 0.8\ \mu s$. The second part consists of two long training sequences (LTSs), which also represent two cycles of another known periodic function of period $3.2\ \mu s$ period, plus a $1.6\ \mu s$ cyclic prefix (GI). The STSs are transmitted over one of every four data subcarriers, making the subcarrier spacing of STSs four times the normal subcarrier spacing, and are used for frame detection and coarse FO estimation. LTSs, on the other hand, employ all the data subcarriers and are used for channel estimation and fine-tuning the coarse FO estimation.

A. FO Estimation and Correction

Let Δf be the actual frequency offset between a transmitter and a receiver. This FO translates into a phase offset of $\Delta\varphi(t) = 2\pi\Delta f t$ for the received signal, where t is the time elapsed since the start of the transmission. In addition to ICI, accumulation of the phase offset over time makes the samples rotate on the constellation map, leading to more bit errors.

The *de facto* FO estimation method used in 802.11a devices is the one proposed by Schmidl and Cox [10]. It assumes that the channel does not change during a preamble. Having a sequence with two identical halves is the key idea in this method. Assume that each half of the sequence has L samples with sampling period of t_s. Let r_i be the ith sample of the sequence, $i = 1, \ldots, 2L$. So $r_i = r_{L+i}$. Ignoring the noise, this equality also holds for the corresponding samples at the receiver as long as there is no FO. However, with an FO of Δf, the phase of r_{L+i} is rotated by $\Delta\varphi = 2\pi\Delta f L t_s$ relative to r_i. Now if we multiply the conjugate of r_i by r_{L+i}, we obtain:

$$s_i \stackrel{\text{def}}{=} r_i^* r_{L+i} = |r_i|^2 e^{-j2\pi\Delta f L t_s} = |r_i|^2 e^{-j\Delta\varphi}. \quad (1)$$

Taking into account the channel coefficient $h_i = h_{L+i}$ and the noise terms, n_i and n_{L+i}, the value of s_i at the receiver, denoted by $\widetilde{s_i}$, is:

$$\widetilde{s_i} = |h_i r_i|^2 e^{-j2\pi\Delta f L t_s} + \bar{n_i} \quad (2)$$

where $\bar{n_i} \stackrel{\text{def}}{=} r_i n_{L+i}^* + r_{L+i}^* n_i + n_i n_{L+i}^*$ has zero mean. To average out the $\bar{n_i}$'s, the estimated phase offset, $\widetilde{\Delta\varphi}$, is measured over the summation of all the $\widetilde{s_i}$'s, i.e.,

$$\widetilde{\Delta\varphi} = \measuredangle\Big(\sum_{i=0}^{L-1} \widetilde{s_i}\Big) \quad (3)$$

where the notation $\measuredangle(x)$ indicates the phase of a complex quantity x. Thus, the estimated FO is:

$$\widetilde{\Delta f} = \frac{\widetilde{\Delta\varphi}}{2\pi L t_s}. \quad (4)$$

Figure 3 shows an example of a sequence of length $2L = 8$ samples. The more samples are used to estimate $\widetilde{\Delta\varphi}$, the more accurate is the estimated FO.

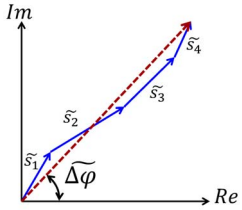


Fig. 3. Example of phase offset averaged over all the $\widetilde{s_i}$ products ($L = 4$).

What the receiver observes regarding the phase of a complex number such as $\widetilde{s_i}$ is a value between $-\pi$ and π. In other words, the receiver cannot distinguish $\Delta\varphi$ from $\Delta\varphi \pm 2k\pi$ in (4), for any integer k. In particular, consider two FOs, Δf_1 and Δf_2, where $|\Delta f_1| \leq \frac{1}{2Lt_s}$ and $|\Delta f_2| = |\Delta f_1| + \frac{1}{Lt_s}$. The corresponding phases are $2\pi|\Delta f_1|Lt_s$ and $2\pi|\Delta f_1|Lt_s + 2\pi$, respectively. Because the phases differ by 2π, there will be an ambiguity in distinguishing between them. The receiver interprets $\Delta f_1 + \frac{1}{Lt_s}$ as Δf_1. In general, the phase is unambiguous and correctable as long as $|\Delta f| < \frac{1}{2Lt_s}$ (half a subcarrier spacing). This implies that, given a fixed t_s, the higher the value of L in a cycle, the smaller is the range of FO that can be corrected unambiguously. Let th_s and th_l be the maximum $|\Delta f|$ values that STSs and LTSs can correct unambiguously, respectively. In the 802.11a preamble, two of the last three STSs are chosen to form a sequence with two identical halves for coarse FO estimation. Since L can be as small as 16 and 64 for an STS and an LTS, the FO estimation is unambiguous as long as $|\Delta f|$ is less than $th_s = 625$ kHz and $th_l = 156.25$ kHz, respectively.

The above discussion reveals a tradeoff between the accuracy and range of the correctable FO. The goal of the STSs is to estimate a large FO value and compensate for it by multiplying the rest of the samples (including those obtained during the LTSs) by $e^{-j(-2\pi\widetilde{\Delta f_s}\, i\, t_s)}$, where $\widetilde{\Delta f_s}$ is the estimated FO in the STSs phase and i is the sample index. Using LTSs as another sequence with identical halves, the receiver then computes $\widetilde{\Delta f_l}$ to fine-tune the coarsely estimated FO. This explains one of the rationales behind the concatenation of a short and a long training sequences in 802.11a systems. Consequently, if the actual FO is larger than 625 kHz, this FO estimation method fails to compensate for it. A device is considered 802.11a-compatible if its FO is confined to this limit. Specifically, only devices with maximum $|\Delta f|$ of $th_d = 212$ kHz [1] comply with the 802.11a standard.[3]

Even after the LTS-based FO correction, a small residual FO may remain due to noise. This error is typically too small to cause ICI, but it starts to gradually rotate the phase of the received symbols on the constellation map and may increase the BER. A predetermined subset of subcarriers (called *pilot subcarrier*) are used to track and compensate for these small phase changes. Theoretically, there is no frequency range limitation for FO estimation in pilot subcarriers [5].

Channel estimation: Channel estimation is the task of estimating the response of the channel. It is applied to each subcarrier. We briefly explain it here because it is affected by the coarse FO estimation. LTSs are used for this purpose because they are supposed to be almost FO-free after STS-based FO correction. There are two general approaches for channel estimation [5]: Frequency domain and time domain. In both approaches, the known LTS symbols are compared with the received symbols in order to estimate the impulse or frequency response that results in the minimum mean-square-error (MSE). Pilot subcarriers also can be used for channel estimation.

[3]For the 2.4 GHz band (802.11g), this value is about 125 kHz.

B. Frame Detection

For a typical 802.11a receiver, an increase in the received power is a first indication of a new PHY frame. To make sure that this increase is indeed due to an 802.11a frame and then time synchronize with it, the receiver checks for the existence of successive identical sequences with a preset length [10].

Schmidl and Cox's method of frame detection works as follows. Consider two non-overlapping intervals, each of duration $k\lambda_{STS}$ microseconds (equivalently, kL samples, where k is an integer) to represent two identical halves of a frame detection sequence. In the 802.11a $1 \leq k \leq 5$. For example, three STSs with a $t_s = 50\ ns$ sample period (owing to the Nyquist rate of 20 MHz) result in $L = 48$ samples. Initially, the intervals are placed back-to-back at the beginning of the received sequence. The correlation between the samples' conjugate in the first interval (window) and the corresponding samples in the second one is computed. Let $\mathcal{A}(n)$ be the summation of these correlations when the first window starts at the nth sample of the whole sequence:

$$\mathcal{A}(n) = \sum_{i=0}^{L-1} \widetilde{s}^*_{n+i}\widetilde{s}_{n+L+i}. \tag{5}$$

Using $\mathcal{A}(n)$, a normalized timing metric, $\mathcal{M}(n)$, is computed as:

$$\mathcal{M}(n) = \frac{|\mathcal{A}(n)|^2}{\big(\mathcal{E}(n)\big)^2} \tag{6}$$

where $\mathcal{E}(n) \stackrel{\text{def}}{=} \sum_{i=0}^{L-1} |\widetilde{s}_{n+L+i}|^2$ is the received signal energy over the second window. $\mathcal{M}(n)$ is close to 0 if either window does not contain any preamble sample. On the other hand, $\mathcal{M}(n)$ peaks when both windows contain only preamble samples. Ideally, $\mathcal{M}(n)$ should stay constant at the maximum value of 1, as long as both the windows are being moved inside the preamble boundaries. So the first time that it hits the maximum is supposed to be the beginning of the frame. However, because of noise, the maximum may occur later than the actual beginning of the preamble. To account for this perturbation, the algorithm first finds $\hat{\mathcal{M}} = \max_n \mathcal{M}(n)$ and then goes back from the beginning and looks for the first occurrence of an $\mathcal{M}$ value greater than $(1-\epsilon)\hat{\mathcal{M}}$, where $0 < \epsilon < 1$ is a system parameter. That point is regarded as the beginning of the frame.

Figure 4 shows two examples of the smallest and largest possible window sizes in the 802.11a frame detection scheme. When $L = 80$, in (6) the noise is almost averaged out because as many as samples as possible are included. So the maximum point is more reliable. In contrast, when $L = 16$, $\mathcal{M}(n)$ is more fluctuating and we need a larger ϵ to decrease the

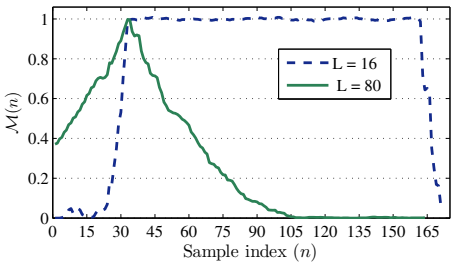


Fig. 4. $\mathcal{M}(n)$ vs. n for two extreme cases of window lengths (noise level: -42 dBm, frame starts at $n = 31$, $t_s = 50$ ns).

chance of missing the actual beginning. Even though the sharp increase of $\mathcal{M}(n)$ allows us to increase ϵ, it is not clear how much we should increase it and thus the selected point is less reliable.

III. Model and Assumptions

We consider a single hop link between Alice (the transmitter) and Bob (the receiver). The adversary (Eve) is in the transmission ranges of both Alice and Bob.

A. Transmitter and Receiver Models

We assume that Alice and Bob employ the Schmidl and Cox method for FO estimation. Bob uses a few of the first STSs for frame detection and chooses two of the last three STSs for FO estimation in conformity with the standard (see Figure 2). Once Bob finishes coarse FO estimation using STSs and compensates for $\widetilde{\Delta f_s}$, he assumes, by default, that the residual FO is less than th_l. According to 802.11a, Bob does not perform any kind of boundary check during the LTSs and pilot subcarriers-based FO estimation processes.

B. Jammer Model

Eve's motivation is to irrecoverably corrupt Alice's frame at Bob using the lowest possible jamming power and duration. Eve knows the PHY-layer protocol and the FO correction mechanism at Bob. Moreover, she makes no assumption about here distance to Alice or Bob, the channel parameters, and Alice's transmission power. Assume all oscillators are either stable or accurate.[4] Eve initially eavesdrops on Alice's and Bob's preamble transmissions for a while (e.g., from data-ACK exchanges). Through averaging, she estimates their FOs relative to Eve, denoted by Δf_a and Δf_b, respectively.[5]

C. Metrics

The parameters of interest are: coarse and final FO estimation at Bob, the MSE of channel estimation, and the BER after demodulation but before decoding. These metrics will be studied with respect to the signal-to-jamming ratio (SJR) and the distance from Eve to Bob (d_{eb}).

[4]The FO of a stable oscillator does not vary much over time, while the FO of a non-stable but accurate one fluctuates randomly with zero mean [12].

[5]In the case of non-stable and inaccurate oscillators, FO estimation is performed along with fast frame detection (see Section IV-A).

IV. Frequency Offset Estimation Attack

Eve fulfills an FO estimation jamming attack in two phases:

1) Eavesdropping on the channel to detect the start of Alice's transmission and acquire its timing information.
2) Jamming the last three STSs of the preamble, which are used for coarse FO estimation.

A. Phase 1: Fast Frame Detection

Eve can pinpoint the last three STSs in time by detecting the start of the preamble, as explained in Section II. However, the detection should be fast enough to allow sufficient time for processing, switching to transmission mode, and preparing for the arrival of last three STSs. Therefore, Eve chooses the minimum possible window size ($L = 16$) and reduces the capture time to $3\lambda_{STS} = 2.4$ μs to make sure that at least two STSs are captured. To account for the reduced detection accuracy, Eve first sets the value of ϵ to $1/16$, the contribution of a preamble sample pair in $\mathcal{M}(n)$. This is also an attempt to exclude the samples located more than one index before the actual frame's start time. Next, she assumes that the actual start time is among the $V = \log_2(L)$ most probable sample indices $i_0, i_1, \ldots, i_{V-1}$ and then finds all of them. The most probable candidate (i_0) is the one provided by the standard frame detection method, and the others are the ones obtained of the same method but after removing the previously considered start times. This would complete the set of parameters needed for triggering the jamming phase, i.e., Δf_a and Δf_b, and the list of probable candidates of the frames's start time.

B. Phase 2: Preamble Jamming

Based on i_0, Eve computes the expected time of the last 3 STSs in the preamble. She then generates her jamming sequence. An energy-efficient jamming sequence should be able to beat both the STSs and LTSs-based FO corrections without jamming LTSs. Further, it has to account for unknown channel parameters and frame-detection timing errors. More specifically, the jamming sequence must satisfy the following:

1) Force Bob to make a destructive error: By default, Bob assumes that the remaining FO during LTSs is less than th_l. If Eve deceives Bob to erroneously push the FO beyond th_l after STSs instead of reducing it, then she achieves her goal without needing to jam the LTSs.

Assume that Eve correctly detects the start time of the frame (we will relax this assumption later) and then aligns her jamming signal with the FO estimation part of the STSs of Alice's signal. Let $\Delta f_{eb} = -\Delta f_b$ and $\Delta f_{ab} = \Delta f_a - \Delta f_b$ represent the FO between Eve and Bob and Alice and Bob, respectively. Next, let $\Delta\varphi_{ab}$, $\Delta\varphi_{eb}$, $\Delta\varphi_l = \pi/4$, and $\Delta\varphi_d = 0.3392\pi$ be corresponding phase offsets of Δf_{ab}, Δf_{eb}, th_l, and th_d, respectively, after a single STS (0.8 μs). To cause a wrong FO estimation ($\widetilde{\Delta f_s}$) such that the updated FO after STSs ($\Delta f_{ab} - \widetilde{\Delta f_s}$) is higher than th_l, the following inequality should hold:

$$|\Delta\varphi_{ab} - \widetilde{\Delta\varphi}| > \Delta\varphi_l. \quad (7)$$

Figure 5 shows an example when this inequality holds.

Let g be the Eve-to-Bob channel coefficient and u be a discrete complex random process representing Eve's signal.

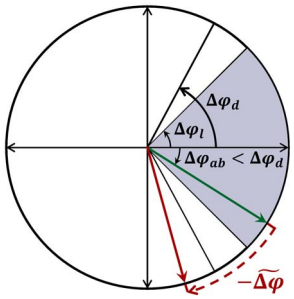


Fig. 5. Phase offsets as observed during the STSs. The shaded area is the LTSs' correctable range. A wrong phase estimation can move $\Delta\varphi_{ab}$ out of the correctable range.

Similar to Alice-to-Bob channel, assume that g is the same for all the jamming samples $u_i, i = 1, \ldots, 2L$. Also, let $\tilde{r} = hr$ and $\tilde{u} = gu$. Now, we consider two different approaches for generating the jamming sequence:

Random noise: The simplest idea to corrupt the FO estimation at Bob is to jam the last three STSs with a random signal. Recalculating the autocorrelation $\mathcal{A}$ at Bob and ignoring the noise term in (2), we have:

$$\begin{aligned}\mathcal{A}_{\text{random}} &\stackrel{\text{def}}{=} \sum_{i=0}^{L-1} \widetilde{s_i} = \sum_{i=0}^{L-1} (\tilde{r}_i + \tilde{u}_i)^* (\tilde{r}_i e^{-j\Delta\varphi_{ab}} + \tilde{u}_{L+i}) \\ &= \sum_{i=0}^{L-1} |\tilde{r}_i|^2 e^{-j\Delta\varphi_{ab}} + \sum_{i=0}^{L-1} \tilde{r}_i{}^* \tilde{u}_{L+i} \\ &\quad + \sum_{i=0}^{L-1} \tilde{u}_i^* (\tilde{r}_i e^{-j\Delta\varphi_{ab}} + \tilde{u}_{L+i}).\end{aligned} \tag{8}$$

The phase and amplitude of the 2nd and 3rd terms in (8) (and hence $\widetilde{\Delta\varphi} \stackrel{\text{def}}{=} \measuredangle\mathcal{A}_{\text{random}}$) are unknown because not only they include random complex numbers $\tilde{u_i}$, but also the phase and amplitude of $\tilde{r}_i$ are unknown after traversing the Alice-to-Bob channel. Even though $\widetilde{\Delta\varphi}$ may satisfy (7), it does not provide any distortion guarantee.

Fake preamble: A smarter jamming approach that exploits both knowledge of FO estimation algorithm and Δf_{ab} is to construct a fake preamble, with the "identical halves" property. For now, assume that the jamming signal time samples can take any arbitrary value as long as the signal conforms to the 802.11a bandwidth requirement (20 kHz). Note that this construction is different from phase warping attack [7], where the jamming signal is a random frequency shift of the preamble. The advantage of having identical halves is that we can control and carefully calculate the FO of u based on how Bob estimates the FO. Here we also note that the channel response between Eve and Bob does not change the FO. Before we explain how a fake FO is determined, consider the superposition of the two signals at Bob. Dropping the index i from (2) and ignoring the noise term, we have:

$$\tilde{s} = (\tilde{r} + \tilde{u})^* (\tilde{r} e^{-j\Delta\varphi_{ab}} + \tilde{u} e^{-j\Delta\varphi_{eb}}) = e^{-j\Delta\varphi_{ab}} \times \underbrace{\left[|\tilde{r}|^2 + |\tilde{u}|^2 e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} + \tilde{r}^* \tilde{u} e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} + \tilde{u}^* \tilde{r} \right]}_{\mathcal{B}}. \tag{9}$$

Thus, the estimated FO at Bob is:

$$\widetilde{\Delta\varphi} = \measuredangle s + \bar{n} = \Delta\varphi_{ab} + \underbrace{\measuredangle\mathcal{B}}_{\varphi_e} + \bar{n}. \tag{10}$$

Note that φ_e is a function of $\Delta\varphi_{eb}$ and if $\varphi_e = 0$, then jamming will have no effect.

Upon calculating $\widetilde{\Delta\varphi}$, Bob changes the phase for the rest of the frame to $\Delta\varphi_{ab} - \widetilde{\Delta\varphi}$. According to (7), Eve is successful if the following inequality holds (assuming $\bar{n} = 0$):

$$|\Delta\varphi_{ab} - \widetilde{\Delta\varphi}| > \Delta\varphi_l = \frac{\pi}{4} \Rightarrow |\varphi_e| > \frac{\pi}{4}. \tag{11}$$

Even if Eve knows $\Delta\varphi_{ab}$ and $\tilde{u}$ and can also control $\Delta\varphi_{eb}$, she has no control over other channel-dependent parameters in $\mathcal{B}$. Specifically, the phase and amplitude of $\tilde{r}$ are channel-dependent and Eve cannot estimate the Alice-to-Bob channel coefficient h as long as she is half a wavelength (a few centimeters) away from Bob [3]. That means that she is still unable to guarantee a successful attack. This is also the case in the preamble phase warping attack.

2) Design a channel-independent jamming signal: To address the aforementioned challenge, Eve takes advantage of the product sum in (3) and eliminates the terms with unknown phases using the known preamble. Without loss of generality, let (u_1, u_2) be the first pair in the jamming sequence samples. By knowing the preamble sample values at Alice, u_2 can be designed such that when Bob sums up $\tilde{s}_1$ and $\tilde{s}_2$, all the terms that depend on $\tilde{r}$ (excluding $|\tilde{r}|$) in the term $\mathcal{B}$ in (9) are eliminated. Thus,

$$u_2 = -\frac{r_1^*}{r_2^*} u_1 \tag{12}$$

which implies that

$$\begin{aligned}&\widetilde{s_1} + \widetilde{s_2} = e^{-j\Delta\varphi_{ab}} \times \\ &\left[|\tilde{r}_1|^2 + |\tilde{r}_2|^2 + (|\tilde{u}_1|^2 + |\tilde{u}_2|^2) \, e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} \right].\end{aligned} \tag{13}$$

The same requirement in (12) is imposed on the rest of the even samples. We refer to this requirement as the *pairing rule*. Accordingly, the autocorrelation function $\mathcal{A}$ for this scheme becomes:

$$\mathcal{A}_{\text{fake}} = \sum_{i=0}^{L-1} \widetilde{s_i} = e^{-j\Delta\varphi_{ab}} \underbrace{\left[\sum_{i=0}^{L-1} |\tilde{r}_i|^2 + \sum_{i=0}^{L-1} |\tilde{u}_i|^2 \, e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} \right]}_{\mathcal{C}(\Delta\varphi_{eb} - \Delta\varphi_{ab})}. \tag{14}$$

$\mathcal{A}_{\text{fake}}$ is a function of the difference between $\Delta\varphi_{ab}$ and $\Delta\varphi_{eb}$ only. Now Eve is able to determine and augment the value of $\Delta\varphi_{eb}$ in a way that makes $|\measuredangle\mathcal{C}(\Delta\varphi_{eb} - \Delta\varphi_{ab})| > \pi/4$, which satisfies (11).

3) Optimizing the jamming power: The jamming sequence can be optimized to minimize $\sum_{i=0}^{L-1} |\tilde{u}_i|^2$, subject to the constraint $|\measuredangle\mathcal{C}(\Delta\varphi_{eb} - \Delta\varphi_{ab})| > \pi/4$. Figure 6 shows $\mathcal{C}(\Delta\varphi_{eb} - \Delta\varphi_{ab}) = |\tilde{r}|^2 + |\tilde{u}|^2 e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})}$ in polar coordinates. The shaded area shows the feasible region.

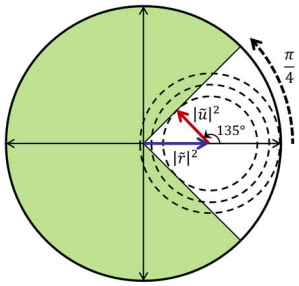


Fig. 6. The optimal phase offset and $|\tilde{u}|^2$ value occur when the vector $|\tilde{u}|^2$ is perpendicular to one of the feasible region edges. The parts of a contour that are inside the shaded area show the feasible phase offsets of a given $|\tilde{u}|^2$.

According to the polar representation, we conclude that:

1) This optimization problem is feasible as long as

$$\text{SJR} = \frac{\sum_{i=0}^{L-1} |\tilde{r}_i|^2}{\sum_{i=0}^{L-1} |\tilde{u}_i|^2} \leq \sqrt{2} \approx 1.5 \text{ dB}. \tag{15}$$

2) The optimal solution is achieved when

$$|\Delta\varphi_{eb} - \Delta\varphi_{ab}| = 3\pi/4, \tag{16}$$

or equivalently, $|\Delta f_{eb} - \Delta f_{ab}| = 468.75$ kHz.

Equation (16) means that the phase offset of Eve's signal as perceived by Bob should have phase difference of $|3\pi/4|$ relative to Alice's signal. Even if $\Delta\varphi_{eb}$ does not satisfy (16), Eve can augment Δf_{eb} by imposing a fake FO of Δf_n on the jamming sequence before having the oscillator transmit it. This is achieved by multiplying the samples of the jamming sequence by $e^{-j2\pi\Delta f_n\, i\, t_s}$. Δf_n is given by:

$$\Delta f_n = |468.75| - \Delta f_{eb} + \Delta f_{ab}. \tag{17}$$

4) Robustness to frame detection error: We now relax the assumption of precise frame detection at Eve and consider a scenario in which Eve compiles a short list of possible frame start times other than i_0, as explained in Section IV-A. Thus far, we have required the jamming sequence to have identical halves with a FO of Δf_n and the even samples to be a function of odd samples (pairing rule). So half of the samples are still unassigned and can take any arbitrary value. To account for detection error, we define the following *chaining rule* for the odd samples.

All but one of the unassigned samples can be defined as a function of another unassigned sample such that the summation of the samples accounts for the other $V-1$ possible frame start times $i_j, j \neq 0$. We explain this for the case of $j=1$ and remaining $L/2^j$ unassigned jamming sequence samples. Let $m_1 = i_0 - i_1 \neq 0$. So the first jamming sample is m_1 samples away from the first sample of Alice's sequence, and (12) cannot eliminate the last two terms within $\mathcal{B}$ in (9). Without loss of generality, we expand the summation of the last term and assume $m_1 < 0$:

$$\begin{aligned}&\sum_{i=1}^{i=8} \tilde{u}^*_{2i-1}\, \tilde{r}_{2i-1-m_1} + \tilde{u}^*_{2i}\, \tilde{r}_{2i-m_1} \\ &= \sum_{i=1}^{i=8} \tilde{u}^*_{2i-1} \left(\tilde{r}_{2i-1-m_1} - \frac{\tilde{r}_{2i-1}\, \tilde{r}_{2i-m_1}}{\tilde{r}_{2i}}\right)\end{aligned} \tag{18}$$

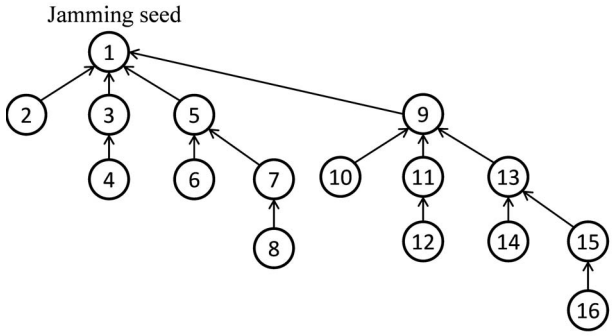


Fig. 7. Cascaded chaining and pairing of the samples towards the jamming seed to account for channel parameters and frame detection errors.

where the equality is due to (12).

Recall that u_{2i-1}'s have not been assigned yet. To eliminate the above summation, Eve takes any two u_{2i-1}'s and defines one of them based on the other such that their summation is 0. For example, if she picks the pairs sequentially, then

$$u^*_{2i+1} = -\frac{r_{2i+2}\left(r_{2i} r_{2i-1-m_1} - r_{2i-1} r_{2i-m_1}\right)}{r_{2i}\left(r_{2i+1-m_1} r_{2i+2} - r_{2i+1} r_{2i+2-m_1}\right)}\, u^*_{2i-1}, \tag{19}$$

where $i = 1, 3, 5, 7$.

With this, half of the previously unassigned samples are assigned. Eve then picks the next possible frame's start time (i_2), rewrites (18) but for the left samples, and defines half of the remaining unassigned samples based on the other half. She continues this process until only one unassigned sample remains. We call this sample the *jamming seed* to which all the other samples are chained either directly or recursively. So this chaining rule can account for up to $V-1$ possible errors. An example of pairing and chaining of the samples based on the first sample is shown in Figure 7.

C. Effect of Long Training Sequences

The LTSs at Bob may still be able to partially correct the manipulated FO and line up the subcarriers, though out-of-order. From these sequences' perspective, the phase offset is between $-\pi$ and π. So as explained in Section II, LTSs will correct up to (Δf_l mod th_l), which is at most half of the subcarrier spacing. Any remaining phase offset must be an integer multiple of 2π, which corresponds to $2kth_l$, or k subcarrier spacing. Consequently, all the subcarrier bins would be shifted forward or backward, and would take their neighboring subcarrier bins. Bob eventually demodulates the bits, but he is unaware that they are shuffled and out-of-order.

Moreover, large FO errors affect the channel estimation performance, which is applied across the LTSs. To elaborate, the phase offset accumulates over time, causing different samples to have different phase offsets. However, Bob complacently tries to interpret this time-varying phase offset as a fixed-value channel phaser. So, his attempt of modeling the FO as if it is a channel parameter causes channel estimation errors and introduces an additional FO after equalization.

V. Defenses

In this section, we propose some possible defenses against the previously presented FO estimation attack.

A. Randomizing Target Sequences (Sequence Hopping)

Because of the redundancy in the STSs, Bob can choose for FO estimation any other pair of consecutive STSs in addition to the ones in last 3 sequences. Furthermore, due to the maximum FO requirement for 802.11a devices ($th_d = 212$ kHz), the two autocorrelation windows do not necessarily need to be contiguous. In fact, the maximum time difference between two identical samples required to unambiguously detect th_d is $2.358\ \mu s$, which is almost three times λ_{STS} ($0.8\ \mu s$). Assuming that the length of a window is one STS, if the two windows are one STS apart from each other (i.e., a sample is two STSs away from its dual), Bob can still estimate the FO correctly. Moreover, provided that $|\text{FO}| < 208.5$ kHz (corresponding to $2.358\ \mu s$), the time difference between the windows can include one more STS.

The above discussion reveals that Bob has the flexibility to randomly hop to any pair of STSs for FO estimation given that they are not more than two STSs away from each other. Even if Bob selects a jamming-free sequence together with a jammed one, he is still able to estimate the same FO as in the case of selecting two jamming-free sequences. We show this by recalculating $\mathcal{A}$. Formally, if the first sequence is a jammed one:

$$\begin{aligned}\mathcal{A}_{\text{hopping}} &= \sum_{i=0}^{L-1}(\tilde{r}_i + \tilde{u}_i)^* \tilde{r}_i e^{-j\Delta\varphi_{ab}} \\ &= \sum_{i=0}^{L-1} e^{-j\Delta\varphi_{ab}}(|\tilde{r}_i|^2 + \tilde{u}_i^* r_i) \overset{(12)}{=} e^{-j\Delta\varphi_{ab}} \sum_{i=0}^{L-1}(|\tilde{r}_i|^2).\end{aligned} \tag{20}$$

If the second sequence is jammed, then:

$$\begin{aligned}\mathcal{A}_{\text{hopping}} &= \sum_{i=0}^{L-1} \tilde{r}_i^*(\tilde{r}_i e^{-j\Delta\varphi_{ab}} + \tilde{u}_{L+i}) \\ &= \sum_{i=0}^{L-1}\left[|\tilde{r}_i|^2 e^{-j\Delta\varphi_{ab}} + \tilde{r}_i^* \tilde{u}_{L+i}\right] \overset{(12)}{=} e^{-j\Delta\varphi_{ab}} \sum_{i=0}^{L-1} |\tilde{r}_i|^2.\end{aligned} \tag{21}$$

To take advantage of all STSs, Bob may first record the received signal while he is in the process of detecting the start time of the frame. Once the frame has been detected and the STSs recorded, he goes back to the beginning and chooses two random numbers between 1 and 10 with the maximum two STSs distance constraint. This counterattack scheme is effective even if Eve decides to jam three other consecutive or noncontiguous sequences instead of the last three STSs.

B. Preamble Obfuscation

Preamble obfuscation aims of making the timing or FO features hard to extract for Eve. It assumes that there is an already exchanged symmetric secret key between Alice and Bob. We provide one simple example for each of the timing and FO extractions.

To have Alice's preamble exhibit misleading timing characteristics, Alice can obfuscate the preamble by adding artificial noise that is generated based on the secret key before transmitting it. Bob, on the other hand, generates the same secret signal and subtracts it from a certain section of the received combo. For example, a signal identical to the first half of an STS may be added to the first half of the second STS in the preamble. So, the increased power would decrease the value of $\mathcal{M}(n)$ through (6) for half of the samples (first $L/2$ samples) while $\mathcal{M}(n)$ stays as before during the rest of the second STS. Hence, Eve cannot include the actual start time sample in any of the V candidates.

Another way of thwarting Eve's attack is by altering the FO of a preamble used to estimate Δf_{ab}. Alice modifies the preamble of only a subset of frames that are suspected to be used by Eve to estimate Δf_a. This subset is a secret between Alice and Bob so Bob can identify the modified preambles. Recall that if $\varphi_e = 0$ (i.e., $\Delta\varphi_{ab} = \Delta\varphi_{eb}$), the attack is unsuccessful. For example, Alice may try to obfuscate the preamble by making Eve over/underestimating Δf_{ab} by 468.75 kHz (or equivalently, over/underestimating $\Delta\varphi_{ab}$ by $3\pi/4$), hoping that it cancels out the $\pm 3\pi/4$ phase difference created by Eve between estimated $\Delta\varphi_{ab}$ and $\Delta\varphi_{eb}$. Without loss of generality, assume that Eve underestimates $\Delta\varphi_{ab}$ by $3\pi/4$. Then, depending on whether Eve adds or subtracts $3\pi/4$, one of the following cases may occur:

1) If Eve adds $3\pi/4$, this makes $\Delta\varphi_{ab} = \Delta\varphi_{eb}$ and so the attack fails.
2) If Eve instead decrements $3\pi/4$ from her estimated $\Delta\varphi_{ab}$ while generating the jamming signal, then $|\Delta\varphi_{ab} - \Delta\varphi_{eb}| = -3\pi/2$. So $\angle\mathcal{C}(-3\pi/2) < \pi/2$ and the attack fails unless Eve boosts her jamming power and makes SJR ≤ 0 dB to satisfy (11) (see Figure 6).

C. FO Rendezvous

Bob can simply disable his FO estimation mechanism during the STSs period to avoid the attack. However, this will bring back the initial fundamental problem: How to compensate for FO?

Recall that LTSs can correct up to $th_l < th_d$. Under BPSK modulation, which is used for transmitting the PHY header, Bob can tolerate channel estimation errors due to an FO estimation error of 15 kHz (5% of the subcarrier spacing [5]). Hence, Bob divides the range of $[-th_s, th_s]$ into $th_d/15$ equal-size frequency regions, and waits for Alice to transmit a known *hello* frame. Since the actual FO falls within one of these regions, Bob can try each of the possible regions and compensate for the amount of the FO of the center frequency of that region before he estimates the FO using LTSs. Eventually Alice and Bob rendezvous and Bob successfully decodes the header. From that point and on, he will keep using his successfully guessed for FO correction for other frames instead of using STSs.

VI. Performance Evaluation

In this section, we evaluate the efficiency and effectiveness of the FO estimation attack and one of the mitigation techniques, namely sequence hopping (SH), through simulations and USRP experiments. We implemented the 802.11a preamble (including both the short and long training sequences) by extending the PHY-layer library functions of LabVIEW. The transmitter appends 500 QPSK-modulated random bits to the preamble. Pilot subcarriers and the coding were not implemented to concentrate on the specific effects of the FO

attack on received raw bits. We discuss the impact of coding or pilot subcarriers in Section VI-C.

Without loss of generality, we assume that the receiver uses the STSs t_9 and t_{10}, as defined in Figure 2, for coarse FO estimation, followed by fine FO estimation using LTSs. Also, the channel estimation is performed over the LTSs using the time domain method [5]. We evaluate the schemes under AWGN channel model with precise frame detection and then in a real environment. We vary the SJR (equivalently, relative distance of Eve to Bob) and the $\Delta f_{eb} - \Delta f_{ab}$ values.

A. Simulation: AWGN Channel with Perfect Frame Detection

First, we consider an AWGN channel with zero delay spread. We assume for now that Eve can correctly detect the start of a frame, and we study the effects of the identical-halves property, pairing rule, and the amount of $\Delta f_{eb} - \Delta f_{ab}$.

Figure 8 shows the coarse FO estimation error ($|\widetilde{\Delta f_s} - \Delta f_{ab}|$) as a function of $\Delta f_{eb} - \Delta f_{ab}$. If $\widetilde{\Delta f_s}$ exceeds 156.25 kHz, then the LTSs cannot correct it, resulting in 312.5 kHz of FO error; a shift in subcarriers' indices. As was shown in Figure 6, the coarse estimation error is always increasing with $\Delta f_{eb} - \Delta f_{ab}$, and hence $\Delta\varphi_{eb} - \Delta\varphi_{ab}$, as long as $0 < \Delta\varphi_{eb} - \Delta\varphi_{ab} < 3\pi/4$. The contours and φ_e in Figure 6 explains this behavior. The error becomes zero when $|\Delta\varphi_{eb} - \Delta\varphi_{ab}| \simeq \pi$. Figure 8 also shows that as long as SJR ≤ 1.46 dB, there is a point where the FO estimation error can exceed th_l. This verifies (15). (Note that the same SJR value in [7] creates only ~ 0.6 frequency error rate.) However, when the SJR is optimal (1.46 dB), the optimal $\Delta f_{eb} - \Delta f_{ab}$ occurs before 468.75 kHz. The noise and modification of Δf_n due to pulse shaping are possible reasons of this variation from the theoretical analysis.

But this is not the only gain of the attack. Even when the spoofed estimation is less than th_l, the channel estimation error may cause a fixed additional phase offset and hence a rotation in the constellation points, as explained in Section IV-C. Figure 9 shows that when $\Delta f_{ab} = 0$, the MSE is increasing with the increase of Δf_{eb} and decreasing with the increase of the SJR. Depending on the amount of the channel and FO estimation error, the attack is in one of the following states: no-impact, rotation, or randomization. Figure 12 shows the constellation diagrams of the transmitted I/Q points and the three different states.

1) No-impact/Rotation: If the rotation keeps the symbol phases within the original constellation regions, we say it has *no impact* (Figure 12-b). As the phases are further increase, the points move to first one of their neighboring regions which makes half the bits flip and then to opposite regions where all the bits flip (Figure 12-c). We call this case *rotation*.

2) Randomization: When the FO estimation error is larger than th_l, the subcarriers are shifted and the data bits are shuffled while decoded, in addition to the channel estimation error. The shift will make the received bits appear as a random sequence relative to the original bitstream. So we call this state as the *randomization*. For example, if the shift is one subcarrier spacing forward, the 111000 bitstream is changed to $xx1110$.

The BER performance depicted in Figure 10 summarizes these cases (again, $\Delta f_{ab} = 0$). Constellation points rotations cause 0, 0.5, and 1 BER periodically while the period is a function of the SJR. But as long as the estimated FO is above the threshold, the BER stays constant at 0.5. Bob can avoid a BER of 1 and do better if he notices there is an attack. Moreover, since $|\tilde{r}|^2$ is unknown to Eve, she cannot predict the Δf_{eb} range in which BER is not zero without passing the threshold. However, having purely random bits, i.e., BER $= 0.5$, in the randomization state precludes any recovery mechanism even if Bob detects the existence of an attack. Further, Eve can estimate the Δf_{eb} range of a successful attack. The effectiveness of the sequence hopping in preserving the BER is also plotted in Figure 10, averaged over more than 300 runs. Assuming Eve randomly jams 3 STSs, Bob does not know in advance which STSs are under attack. So he cannot avoid picking two of the jammed sequences in $3/17$ of cases, which explains the non-zero BER.

B. USRP Experiments: Real Environment

We also implemented the FO estimation attack and the SH mitigation technique on an NI USRP-2921 testbed to evaluate the entire scheme in a real environment; ISI channel. The experiment configuration consisted of 3 USRPs acting as Alice, Bob, and Eve. We note that the reaction time, which consists of the communication delay between a USRP and its host PC through an Ethernet cable, host's processing delay, and the time to initialize for transmitting the received jamming sequence, exceeds several milliseconds. So Eve will miss the rest of the frame before the corresponding USRP starts jamming.[6]

To overcome this performance bottleneck, we made the following modification in the implementation. Alice keeps sending many frames periodically with a known period of $\bar{t}$ ms. Upon detecting a received power increase, Eve captures 2.4 μs worth of the sequence. If a frame is detected, she assumes that the next frame starts exactly $\bar{t}$ after the start of the previous one. The host then constructs a jamming signal based on the timing information of the first frame detection and sends it to the USRP. After initializing, the USRP's onboard timer, which has nanoseconds accuracy, waits for the remaining time before the next frame arrival. Once the timer expires, the device starts jamming the preamble of the new frame.

In this experiment, we varied the Eve-to-Bob distance normalized with the Alice-to-Bob distance, denoted by d_{eb}, and set Eve's jamming power equal to Alice's transmission power. Figure 11 shows the final estimated FO error for different values of $\Delta f_{eb} - \Delta f_{ab}$. Because our USRPs do not have stable oscillators, each time we change one of the parameters and restart the USRPs, Δf_{ab} is measured again. So the x-axis represents the difference between Δf_{eb} and the latest measured Δf_{ab}. The average Δf_{ab} was 146.95 kHz with a standard deviation of 14.9 kHz. Each value of the estimated FO is obtained by averaging over 35 runs.

When Eve and Alice are almost equidistance from Bob ($d_{eb} = 0.98$), Eve can effectively jam Bob and stably force him to estimate an incorrect FO that is almost a subcarrier spacing away from the actual one (312.5 kHz) for most of $\Delta f_{eb} - \Delta f_{ab}$ values. With the increase of $\Delta f_{eb} - \Delta f_{ab}$, the corresponding $\Delta\varphi_{eb} - \Delta\varphi_{ab}$ exceeds π and causes a negative-value error

[6]This is not the case for an off-the-shelf reactive jammer because it usually has an onboard processor and fast dedicated hardware. Implementing a reactive jammer on the USRP's FPGA can achieve a reaction time of 15 μs [11].

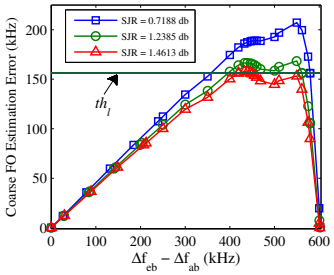


Fig. 8. Estimated FO error after the short training sequences.

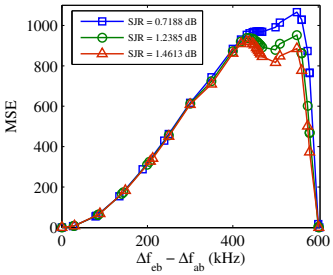


Fig. 9. Mean-Square-Error of channel estimation ($\Delta f_{ab} = 0$).

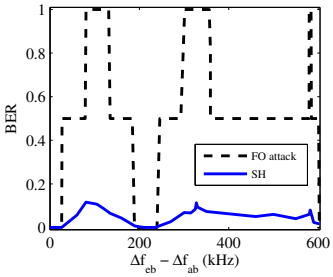


Fig. 10. BER performance of the FO attack and the SH mitigation technique (SJR = 0.7188 dB and $\Delta f_{ab} = 0$).

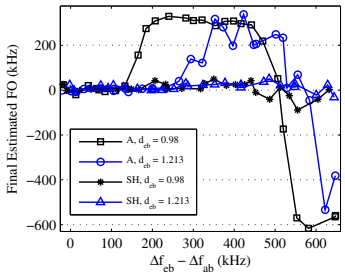
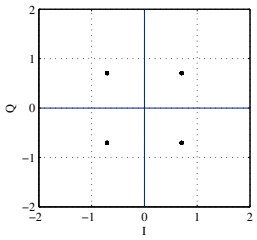
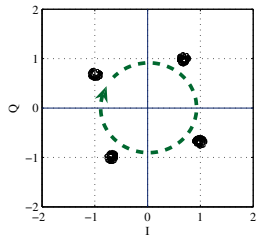


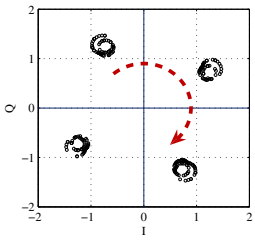
Fig. 11. USRP results: Final FO estimation error of the attack (A) and the sequence hopping scheme (SH).



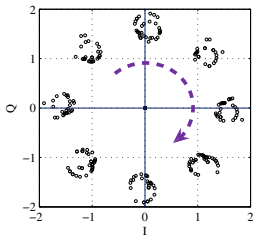
(a) $\Delta f_{eb} - \Delta f_{ab} = 0$ kHz



(b) $\Delta f_{eb} - \Delta f_{ab} = 220$ kHz



(c) $\Delta f_{eb} - \Delta f_{ab} = 320$ kHz



(d) $\Delta f_{eb} - \Delta f_{ab} = 360$ kHz

Fig. 12. Different states of the attack: (a) transmitted constellation, (b) no-impact state, (c) rotation state, and (d) randomization state (SJR = 1.2385 dB).

(backward shift) equal to twice the subcarrier spacing. But if Eve moves further away from Alice ($d_{eb} = 1.213$), the range of effective $\Delta f_{eb} - \Delta f_{ab}$ values shrinks and the attack is no longer stable. This frequency range is inline with the results of the simulation when SJR= 1.46 dB. Also, the distance ratio of $d_{eb} = 1.213$ is approximately equivalent to the SJR in (15). For larger values of d_{eb}, the attack does not show a stable behavior and is not successful. The mitigation technique, on the other hand, performs well in protecting Bob as it significantly reduces the average FO estimation error.

C. Discussion

802.11a systems use coding to increase the resiliency against bit errors. Even though we have not integrated coding techniques in our experiment, the random bits at the receiver with maximum possible BER (0.5) is sufficient to claim that the mutual information is zero and coding cannot rescue the frame. Moreover, one might say that because pilot subcarriers are supposed to be at certain frequencies, the receiver can compare the known symbols of pilot subcarriers to the received symbols over different subcarriers to find out if there is a shift. However, because the channel estimation is distorted, locating the corrupted pilot subcarriers is challenging. Also, corrupted pilot subcarriers cannot be easily used for channel estimation.

VII. Conclusion

We studied the vulnerability of the FO estimation mechanism of 802.11a/g/n OFDM systems against energy-efficient and highly disruptive DoS attacks. We designed an FO estimation attack that is channel-independent and also robust to limited frame detection errors. This short-lived attack is able to impose the maximum BER even if the jammer is located further away from the receiver compared with the legitimate transmitter. We also proposed several mitigation techniques that can provide a protection for the 802.11a systems against FO estimation attacks.

References

[1] *Supplement to IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, IEEE Std 802.11a-1999 1999.

[2] T. C. Clancy. Efficient OFDM denial: Pilot jamming and pilot nulling. In *Proc. IEEE Int. Conf. Commun. (ICC'11)*, June 2011.

[3] A. Goldsmith. *Wireless communications*. Cambridge University Press, 2005.

[4] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of RF interference on 802.11 networks. In *Proc. ACM SIGCOMM'07 Conf.*, pages 385–396, New York, NY, USA, 2007.

[5] J. Heiskala and J. Terry. *OFDM Wireless LANs: A Theoretical and Practical Guide*. SAMS Publishing Indianapolis, 2002.

[6] M. J. L. Pan, T. C. Clancy, and R. W. McGwier. Jamming attacks against OFDM timing synchronization and signal acquisition. In *Proc. Military Commun. Conf. (MILCOM'12)*, Orlando, FL, Nov. 2012.

[7] M. J. L. Pan, T. C. Clancy, and R. W. McGwier. Phase warping and differential scrambling attacks against OFDM frequency synchronization. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'13)*, pages 2886–2890, Vancouver, BC, Canada, May 2013.

[8] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *IEEE Commun. Surveys Tutorials*, 13(2):245–257, 2011.

[9] T. Pollet, M. Van Bladel, and M. Moeneclaey. BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise. *IEEE Trans. Commun.*, 43(234):191–193, 1995.

[10] T. M. Schmidl and D. C. Cox. Robust Frequency and Timing Synchronization for OFDM. *IEEE Trans. Commun.*, 45(12):1613–1621, 1997.

[11] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. Short paper: Reactive jamming in wireless networks—How realistic is the threat? In *Proc. ACM WiSec'11 Conf.*, pages 47–52, Hamburg, Germany, 2011.

[12] H. Zhou, C. Nicholls, T. Kunz, and H. Schwartz. Frequency accuracy & stability dependencies of crystal oscillators. Technical Report SCE-08-12, Carleton University, Sys. and Comput. Eng., Nov. 2008.