Interleaving Jamming in Wi-Fi Networks

Triet D. Vo-Huu Tien D. Vo-Huu Guevara Noubir

College of Computer and Information Science
Northeastern University
Boston, MA 02115, USA
{vohuudtr|tienvh|noubir}@ccs.neu.edu

ABSTRACT

The increasing importance of Wi-Fi in today's wireless communication systems, both as a result of Wi-Fi offloading and its integration in IoT devices, makes it an ideal target for malicious attacks. In this paper, we investigate the structure of the combined interleaver/convolutional coding scheme of IEEE 802.11a/g/n. The analysis of the first and second-round permutations of the interleaver, allows us to design deterministic jamming patterns across subcarriers that when de-interleaved results in an interference burst. We show that a short burst across carefully selected sub-carriers exceeds the error correction capability of Wi-Fi. We implemented this attack as a reactive interleaving jammer on the firmware of the low-cost HackRF SDR. Our experimental evaluation shows that this attack can completely block the Wi-Fi transmissions with jamming power less than 1% of the communication (measured at the receiver) and block 95% of the packets with less than 0.1% energy. Furthermore, it is at least $5\,\mathrm{dB}$ and up to $15\,\mathrm{dB}$ more power-efficient than jamming attacks that are unaware of the Wi-Fi interleaving structure.

1. INTRODUCTION

The broadcast nature of the wireless medium makes it vulnerable to two types of major attacks *denial of service*, and *information leakage*. Designing countermeasures to wireless DoS attacks before they become widespread is very important for both military and commercial applications. Due to a series of recent incidents, the FCC has stepped up its education and enforcement effort [11], rolled out a new jammer tip line (1-855-55NOJAM), and issued several fines [12]. At the same time jammers are becoming a commodity and are growing in sophistication and convenience of use and deployment. Beyond degrading a critical communication infrastructure, wireless DoS can also be the prelude to more sophisticated attacks where the adversary deploys rogue infrastructure [6]. Evidence of such attacks in the real world started emerging in the recent years [13, 33].

Within the wireless ecosystem, Wi-Fi (IEEE 802.11) has emerged as the defacto primary technology for connecting devices to the Internet. This manifests itself first in the increasing Wi-Fi offloading of mobile traffic, caused by the limited ability of cellular ISPs to scale to applications demands, and second in the integration of Wi-Fi in a variety of low-cost Internet of Things (IoT) and Machine to Machine (M2M) devices.

In this work, we are interested in investigating the most efficient, yet practical, jammer against IEEE 802.11a/g/n physical layer. We analyzed the structure of the combined interleaver/convolutional coding scheme of 802.11a/g/n, and observed two key properties: (1) the coded bits' deterministic and predictable interleaving pattern common to all frames, and (2) the interleaving is deterministic across OFDM subcarriers. Further analysis, of the first and second round permutations of the interleaver, allows us to design jamming patterns across subcarriers that when de-interleaved results in an interference burst. We show that a short burst across carefully selected subcarriers exceeds the convolutional code error correction capability. The 802.11 OFDM interleaving across subcarriers makes interleaving attacks highly practical. In order to evaluate the efficiency of this attack, we developed an experimental testbed that enables a systematic comparison of various types of IEEE 802.11 jamming attacks. We implemented a reactive jammer that specifically targets 802.11a/g/n frames. This jammer runs as part of the firmware of the low-cost HackRF One and achieves a response time of less than $30\mu s$. Using a tested including our HackRF-based jammer, off-the-shelf Wi-Fi cards transmitter/receivers, and also our own Wi-Fi SDR receiver [39], we investigated several interleaved jamming techniques (including single and multi OFDM symbols) and compared them to whole band jamming and pilot jamming. We show that interleaving jamming is 5-15 dB more efficient than the most efficient known techniques. In the absolute, we show that an adversary can destroy over 95% of the packets with an energy cost less than three orders of magnitude in comparison to the communicating nodes; blocking all communication requires less than two order of magnitude energy in comparison the communication nodes. Interleaving jamming can be combined with attacks on rate adaptation [25, 27], and potentially with other attacks [32], and can be embedded in traditional Access Points firmware [4]. Besides understanding the threat of such attacks against IEEE 802.11a/g/n, interleaving jamming can also be used for spatial access control [4, 18], as well as in other IEEE 802.11 OFDM systems. We summarize our contributions as follows:

- We analyze IEEE 802.11a/g/n physical layer (modulation, coding, interleaving) for OFDM. We discover that the deterministic combined first/second round permutation when combined with multi-carrier coding in OFDM, enables efficient interleaving jamming attacks.
- We developed a reactive jammer using the low-cost HackRF One SDR platform that can realize the interleaving jamming

WiSec'16 , July 18–20, 2016, Darmstadt, Germany

DOI: http://dx.doi.org/10.1145/2939918.2939935

Table 1: Relations between Rate [Mbps], Modulation and Coding Scheme, Interleaving size m [bits] and Number of bits per subcarrier b [bits]

Rate	MCS	b	m	Rate	MCS	b	m
6	BPSK 1/2	1	48	24	16-QAM 1/2	4	192
9	BPSK 3/4	1	48	36	16-QAM 3/4	4	192
12	QPSK 1/2	2	96	48	64-QAM 2/3	6	288
18	QPSK 3/4	2	96	54	64-QAM 3/4	6	288

attack in real time. This jammer is implemented in the HackRF firmware and has a response time of $30\mu s$ making it practical even for high rates short packets.

- We demonstrate that the interleaving jamming attack can significantly degrade and even block Wi-Fi communications at an energy cost of 2 to 4 orders of magnitude less than the communicating nodes (not even accounting for the additional benefits of being reactive and/or combining with impact on rate adaption). We also show the interleaving jammer is 5-15 dB more efficient than existing Wi-Fi attacks (including pilot jamming). The performance evaluation is carried both on our custom made Wi-Fi receiver and commercial Wi-Fi cards.

2. IEEE 802.11 PHYSICAL LAYER INTERLEAVING

In this section, we briefly overview the interleaving mechanism used at the Physical Layer in OFDM-based IEEE 802.11 networks. Figure 1 illustrates a packet transmission flow carried out at the Physical Layer. The interleaving mechanism is performed by the Interleaver component as part of the Encoding phase. The goal of interleaving is to improve the receiver's capability of correcting bursty errors that might happen due to channel distortions during the signal propagation. The principle of interleaving is to scatter the bursts of errors by separating bits in a small vicinity to larger distances and vice versa. Specifically, the interleaving process defined in the IEEE 802.11 standard first divides the coded bit sequence produced by the convolutional encoder into multiple same-size groups. The number of bits per group, or the *interleaving size*, depends on the Modulation and Coding Scheme (MCS) specified in the RATE field of the Physical Header. More precisely, let b be the number of bits per subcarrier (BPSC), i.e., the number of bits transmitted per constellation point, the interleaving size m is determined by $m = 48b$ (cf. Table 1). Within each group of m bits, the interleaving process is performed in two rounds of permutations.

First-round permutation: The purpose of the first-round permutation is to scatter adjacent coded bits into non-adjacent subcarriers in order to counter interference affecting multiple adjacent subcarriers. This permutation is performed by

$$K' = (K \bmod 16)\frac{m}{16} + \lfloor K/16 \rfloor,$$

where $K, K' = 0, \ldots, m-1$ are the positions of a bit before and after the first-round permutation, respectively. Intuitively, this permutation is thought as if the m-bit input group was arranged in a matrix of 16 rows and $(m/16)$ columns, where bits are stored in column-major order, then the positions of m output bits were read in row-major order. Figure 2 illustrates an example of first-round permutation for BPSK modulation ($m = 48$).

Second-round permutation: The second-round permutation's purpose is to shuffle adjacent coded bits within every subcarrier in order to avoid biased distortion that might occur on the same bit of multiple constellation points. The permutation rule is defined by the following formula

$$K'' = s\lfloor \frac{K'}{s} \rfloor + (K' + m - \lfloor 16\frac{K'}{m} \rfloor) \bmod s,$$

where $s = \max(b/2, 1)$. Note that for BPSK and QPSK modulations, where $s = 1$, the second-round permutation has no effect (i.e., interleaving BPSK and QPSK data is equivalent to first-round permuting only). For 16-QAM and 64-QAM modulations, the second-round permutation is interpreted as cyclically shifting each half of a constellation point. The number of bits shifted is either 0 or 1 for 16-QAM, and either 0, 1, or 2 for 64-QAM, dependently on the subcarrier index. Figure 3 shows all possible permutations for the second round.

We observe from the above Wi-Fi interleaving rule that while the first-round permutation separates adjacent bits into two different subcarriers, the second round permutes bits within the same subcarrier. This can be viewed as an outer permutation followed by an inner permutation.

3. INTERLEAVING JAMMING

In this section, we study the effectiveness of the interleaving structure defined by IEEE 802.11 from the viewpoint of an adversary, and based on that, we propose an efficient jamming strategy. We will later show in Section 5 that it can significantly degrade the performance of Wi-Fi and even block it at very low energy cost.

3.1 Understanding the Interleaving Pattern

First, we illustrate the operation of the interleaver, using as an example rate 54 Mbps, which used 64-QAM modulation and a convolutional code with rate 3/4. According to the interleaving rule described in Section 2, the interleaving table (mapping results after two rounds of permutations) is constructed and partially shown in Figure 4, in which each data subcarrier (DSC) carries 6 bits. Note that for the indexing of DSCs, since the interleaving is performed only on the data subcarriers, we index the DSCs from 0 to 47, skipping the pilot and null subcarriers[1].

We emphasize that while the transmission rate can vary across packets according to the channel state (by means of rate adaptation algorithms), each individual packet is transmitted using one constant rate for all of the OFDM symbols. Therefore, the same interleaving table is used for every group within a packet. For instance, when a 1500-bytes packet is transmitted at 54 Mbps, it contains 58 groups of 288 bits, each of which corresponds to one OFDM symbol. All the OFDM symbols are interleaved in the same manner.

In Figure 4, each square represents a bit after interleaving, while its numeric content indicates the original bit position before interleaving. For instance, the first 6 squares contain bits originally located at positions 0, 16, 32, 48, 64, and 80. By the first-round permutation, any two bits carried by the same subcarrier symbol are originally from those positions whose difference is a multiple of 16. By the second-round permutation, bits of each half of a subcarrier can be rotated, e.g., $(1, 17, 33) \rightarrow (17, 33, 1)$. An important pattern is observed that the coded bits at original positions $0, 1, 2, 3, 4, \ldots$ are interleaved into new positions $0, 20, 37, 54, 74, \ldots$, respectively corresponding to DSC $0, 3, 6, 9, 12, \ldots$, which are separated at distances of multiples of 3. Interestingly, this property does not only hold for rate 54 Mbps, but also holds for all non-BPSK modulations, as stated in Theorem 1.

[1]The 20 MHz of Wi-Fi channel is divided into a total 64 subcarriers, among which 48 subcarriers are used for data transmission, 4 pilot subcarriers are inserted among DSCs for channel estimation, and the remaining 12 null subcarriers are used as guards to avoid inter-channel interference.

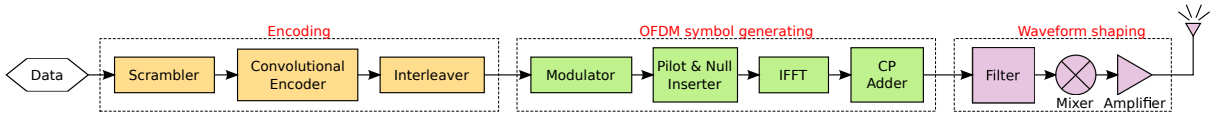


Figure 1: A packet transmission flow consists of three phases: (1) in the Encoding phase, the PSDU (Physical Service Data Unit) received from the MAC Layer is transformed into a sequence of coded bits with a certain amount of redundancy for error correction, then the coded bit sequence is embedded into the Physical Frame with an appropriate Physical Header and payload padding; (2) in the second phase, OFDM (Orthogonal Frequency Division Multiplexing) symbol signals are generated by a series of digital signal processing operations on the Physical Frame; and (3), finally the Wi-Fi signals are upconverted to the channel's carrier frequency and transmitted by the RF front end.

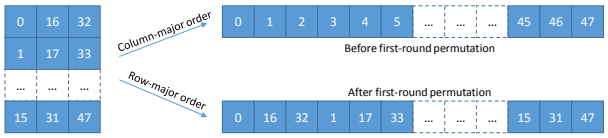


Figure 2: First-round permutation for BPSK modulation ($m = 48$).

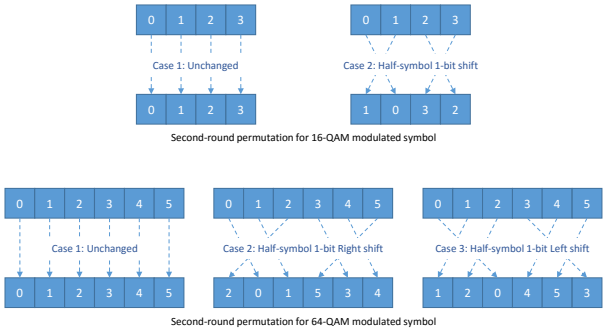


Figure 3: Second-round permutation for 16-QAM and 64-QAM modulations. Depending on the subcarrier index, this permutation can rotate every half-symbol by 0, 1, or 2 bits.

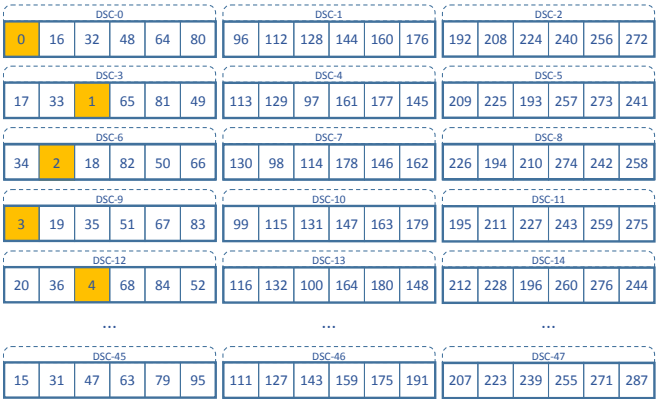


Figure 4: Interleaving table for a 64-QAM transmission. A data packet is divided into multiple same-size groups. In each group of 288 bits, every 6 bits are embedded into one data subcarrier (DSC). The shaded squares indicate the adjacent bits in the original data packet are now interleaved into DSCs separated by distances of multiples of 3.

THEOREM 1. *For IEEE 802.11 non-BPSK transmissions, adjacent bits at positions K and $K+1$ in the original coded data sequence, where $K \notin \{\frac{1}{3}m-1, \frac{2}{3}m-1\}$, are interleaved into separate data subcarriers, whose distance (in the spectrum) is a multiple of 3.*

PROOF. We adopt the notations introduced in Section 2, where K, K', K'' are respectively bit positions prior to interleaving (K), after the first-round permutation (K'), and after the second-round permutation (K''). Also let $M(x) = \lfloor \frac{x}{b} \rfloor$ denote the index of the DSC carrying the bit at position x.

We consider two adjacent bits at positions K and $K+1$ in the coded data sequence (output of the convolutional encoder) prior to interleaving. After interleaving, these bits will be transmitted by subcarriers $M(K'')$ and $M((K+1)'')$. In the following, we investigate the relation between $M(K')$ and $M((K+1)')$ after the first permutation. Recall that $m = 48b$, we consider two cases:

Case 1: If $K+1 \neq 0 \bmod 16$, then $\lfloor (K+1)/16 \rfloor = \lfloor K/16 \rfloor$, and we obtain

$$\begin{aligned}(K+1)' &= ((K+1) \bmod 16)\frac{m}{16} + \lfloor (K+1)/16 \rfloor \\ &= K' + \frac{m}{16} = K' + 3b.\end{aligned}$$

The subcarrier index is then derived as $M((K+1)') = \lfloor (K' + 3b)/b \rfloor = M(K') + 3$, i.e., the adjacent bits at positions K and $K+1$ are permuted into DSCs of distance 3.

Case 2: If $K+1 = 0 \bmod 16$, then $\lfloor (K+1)/16 \rfloor = \lfloor K/16 \rfloor + 1$, $K = 15 \bmod 16$, and

$$\begin{aligned}(K+1)' &= ((K+1) \bmod 16)\frac{m}{16} + \lfloor (K+1)/16 \rfloor \\ &= \lfloor K/16 \rfloor + 1 = K' - (K \bmod 16)\frac{m}{16} + 1 \\ &= K' - 45b + 1.\end{aligned}$$

The subcarrier carrying the bit originally at position $K+1$ is $M((K+1)') = \left\lfloor \frac{K'+1}{b} \right\rfloor - 45$. We see that if $K' \neq -1 \bmod b$, then $\lfloor (K'+1)/b \rfloor = \lfloor K'/b \rfloor$, and $M((K+1)') = M(K') - 45$, in which case adjacent bits K and $K+1$ are permuted into subcarriers of distance 45. For example, Figure 4 shows that two adjacent bits originally at positions $K = 15$ and $K+1 = 16$ are now located in DSC $M(K') = 45$ and $M((K+1)') = 0$. Similar patterns are observed for bits at original positions $K = 31, 47, 63, \ldots$, except when $K = 95$ and $K = 191$, the distance becomes 44.

In fact, we show that there are only two values of K that result in adjacent bits K and $K+1$ being moved to two DSCs with distance of 44. This subcase happens when both conditions $K+1 = 0 \bmod 16$ and $K' = -1 \bmod b$ hold. First, as $K+1 = 0 \bmod 16$, we write $K = 16k + 15$ for some integer k, then express $K' = (K \bmod 16)\frac{m}{16} + \lfloor K/16 \rfloor = 45b + \lfloor K/16 \rfloor = 45b + k$. Next, due to the second condition $K' = -1 \bmod b$, i.e., $45b + k =$

$-1 \bmod b$, it is required that $k = qb-1$ for some integer q. Finally, combining the above requirements, we obtain $K = 16(qb-1) + 15 = qm/3 - 1$. Given the constraint $0 \leq K \leq m-1$, these conditions are satisfied by only two values of K: $K = m/3-1$ or $K = 2m/3-1$.

Since the second-round permutation shuffles the bits only within each data subcarrier, $M(K'') = M(K')$ and $M((K+1)'') = M((K+1)')$, i.e., the mapping between DSCs and bits after the first-round permutation is not altered by the second-round permutation. Therefore, we only need to investigate the bit-subcarrier mapping after the first permutation.

In summary, except for $K = m/3-1$ and $K = 2m/3-1$, adjacent bits at positions K and $K+1$ will be interleaved into data subcarriers separated by a distance of multiple of 3. □

Using Theorem 1, one can derive which subcarriers will carry the adjacent bits, and how far they are after the interleaving process. From the point of view of the adversary, however, the reverse mapping is desired, that determines in case of some particular subcarriers being jammed, which bits are destroyed and whether they are adjacent to each other. This reverse mapping is provided by Theorem 2 as follows.

THEOREM 2. *For IEEE 802.11 non-BPSK transmissions, any two data subcarriers, whose distance is either* 3 *or* 45*, always consist at least two bits originally located adjacently in the coded data sequence.*

PROOF. The proof for this theorem is derived directly from Theorem 1. Let K and L be the positions of two bits before interleaving. Under the theorem's assumption, either $M(L'') = M(K'') + 3$ or $M(L'') = M(K'') - 45$ holds.

If $M(K'') \neq M((m/3-1)'')$ and $M(K'') \neq M((2m/3-1)'')$, then by Theorem 1, we have $K \neq m/3-1$ and $K \neq 2m/3-1$. Now, due to the assumption of distance 3 or 45, we conclude $L = K+1$. In other words, the data subcarriers $M(K'')$ and $M(L'')$ consist of bits originally at positions K and $K+1$.

If $M(K'') = M((m/3-1)'')$ or $M(K'') = M((2m/3-1)'')$, then letting $\hat{K} = K-1$ and $\hat{L} = L-1$, we have $M(\hat{K}'') = M(K'')$ and $M(\hat{L}'') = M(L'')$. By similar arguments above, we have $\hat{L} = \hat{K}+1$. We conclude that the data subcarriers $M(\hat{K}'')$ and $M(\hat{L}'')$ carry bits originally at positions $\hat{K}$ and $\hat{K}+1$, or equivalently $M(K'')$ and $M(L'')$ consist positions $K-1$ and K.

Consequently, the two data subcarriers with distance 3 or 45 always contain at least two bits originally adjacent to each other. □

3.2 The Interleaving Jamming Strategy

Theorem 2 imply that if two data subcarriers separated by distance 3 (or 45) are not correctly decoded at the receiver, there will be two adjacent bit errors in the bit sequence fed to the convolutional decoder. In general, a sequence of n consecutive bit errors is created when interference is caused to a group of n DSCs separated by distances that are multiples of 3. As the design of IEEE 802.11 standard has mainly focused on protecting the communications against non-malicious interference, the specified interleaving structure is only sufficient for dealing with typically random noise in the environment, where only a few of subcarriers at random positions are defective at a given time, and is adequate for a repeating worst-case scenario. Against multi-carrier malicious interference, according to the pattern we identified, the Wi-Fi interleaving process is unable to prevent bursts of bit errors, making the convolutional codes ineffective. Exploiting this property, we devise an *interleaving jamming* strategy as follows.

DEFINITION 1 (INTERLEAVING JAMMING). *Interleaving jamming is a multi-carrier jamming strategy that generates interference on data subcarriers $i, i+3, i+6, \ldots, i+3(n-1)$, where i is any starting data subcarrier, and n is the number of subcarriers targeted for jamming.*

In Section 5 we evaluate the effectiveness of this interleaving jamming attack and the impacts of parameters i and n.

4. INTERLEAVING JAMMER DESIGN

In this section, we describe the design of the interleaving jammer that we use to demonstrate the efficiency of this attack against Wi-Fi communications. The jammer is implemented on the low-cost HackRF One software defined radio [16]. In order to enable real-time jamming, we design our jammer to be capable of quickly detecting the transmitted frames and reactively jamming with a variable duration pulse without the need of decoding the whole packet.

4.1 Frame Detection

The frame detection is based on the special format of preamble at the beginning of every transmitted frame. Specifically, the IEEE 802.11a/g/n preamble comprises two parts: short preamble and long preamble. Our frame detection relies on the short preamble, which contains 10 repeated patterns, each of which consists of 16 samples. We emphasize that while the short preamble in the IEEE 802.11n Greenfield mode is different from a/g modes, the repetition of 10 patterns is still preserved. Our detection technique described below is, therefore, able to detect frames transmitted in all modes.

The main idea of frame detection is based on the auto-correlation property of the short preamble. Let $\mathbf{p} = (p_1, p_2, \ldots, p_L)$ be a time-domain pattern of $L = 16$ samples repeated 10 times in the short preamble. At the receiver, we obtain the received signal r_n as a sample sequence consisted of preamble and data parts of transmitted frames separated by the inter-frame spacing (IFS).

$$\{r_n\} = \underbrace{\ldots\ldots\ldots\ldots}_{\text{inter-frame spacing}}, \underbrace{\hat{p}_1, \ldots, \hat{p}_L, \hat{p}_{L+1}, \ldots, \hat{p}_{2L}, \ldots}_{\text{preamble starting with 10 short patterns}}, \underbrace{\ldots, \hat{x}_k, \ldots}_{\text{data}}$$

Let A_n denote the correlation between two consecutive L-sample chunks, and E_n be the energy of the current chunk at each time n: $A_n = \sum_{k=0}^{L-1} r_{n+k+L} r^*_{n+k}$, $E_n = \sum_{k=0}^{L-1} |r_{n+k}|^2$. Due to the auto-correlation property of the preamble, the ratio $|A_n/E_n|$ exceeds a high threshold value when the preamble is found at time n, otherwise it remains low. Specifically, a frame is detected, if $|A_n/E_n| \geq \alpha$. We determine the parameter α based on our testbed experiments, in which $\alpha \in [0.8, 0.9]$ results in best detection.

In practice, false detection may occur due to several reasons. First, the low noise floor E_n, especially when the channel is idle, may unexpectedly result in a high ratio $|A_n/E_n|$. Moreover, if the data part of the transmitted signal contains a repetition of exactly 16 samples, the above condition can also be triggered. To reduce the false positive detection rate, we include into the frame detection two additional mechanisms: power squelch, and plateau detection.

Power squelch: Based on the energy of the signal received during two consecutive chunks, we can quickly differentiate between the channel idle states and transmission activities, thus mitigating the false detection rate due to noise. Specifically, a transmission is identified at time n when $E_n/E_{n-L} > \beta$. In our experiments, β can be chosen between 3 and 25, for which transmission activities are detectable with high accuracy. We note that as the power squelch is an energy-based detection, it does not distinguish Wi-

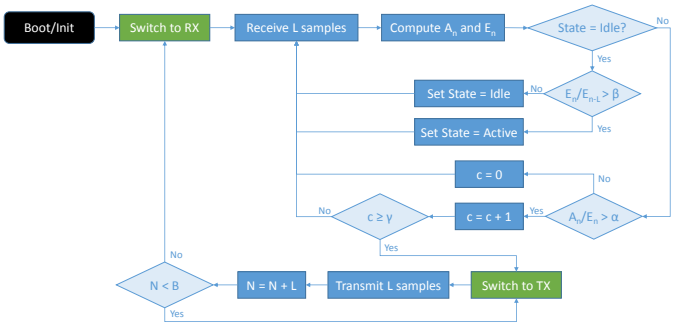


Figure 5: Flow chart of our jammer prototype.

Fi and non-Wi-Fi packets. Wi-Fi packets are recognized by the plateau detection described as follows.

Plateau detection: Based on the auto-correlation of short preamble symbols, a high ratio $|A_n/E_n|$ indicates two repeated patterns to be found. Since repetitions in other parts of the received signal might also result in a high ratio, we increase the detection confidence by requiring the appearance of multiple repeated patterns in a row. The frame is believed to be present in the received signal when the count c of consecutive repeated patterns exceeds a threshold γ. Specifically, if $|A_{n_1}/E_{n_1}| \geq \alpha, \ldots, |A_{n_c}/E_{n_c}| \geq \alpha$ for $c \geq \gamma$, $n_i = n_1 + (i-1)L$, then the frame is detected. We use $\gamma = 4$ in our prototype, as we found that it leads to the best detection performance.

4.2 Real-time Jamming

The HackRF, on which we build the jammer, is a software defined radio capable of capturing wireless signals, converting them to digital samples and transferring them to a computer for the remaining steps of the receiver chain (including signal processing and data decoding). On the reverse direction, the HackRF takes generated samples from the PC and emits the analog signals using the radio RF front end. While this architecture provides flexibility for developing a variety of useful radio applications, the latency due to sample transfer between the HackRF and the PC over USB 2.0 is in orders of milliseconds, which significantly exceeds the requirements of real-time jamming that needs to quickly react within orders of microseconds. For instance, a 1500-byte TCP packet length is approximately $250\mu s$ at $54\,\mathrm{Mbps}$.

To overcome the timing issue, we modified the HackRF firmware such that all functionalities required for the real-time interleaving jamming are done on the HackRF itself without interacting with the PC. In other words, samples are not transferred over the USB link, but are handled by the HackRF's micro-controller, which also takes care of controlling the radio to transmit jamming signals.

Figure 5 shows the flow chart of our custom firmware developed for the interleaving jamming prototype on HackRF. The whole radio application, including frame detection and interference generation, is implemented on the NXP LPC4320 micro-controller on the HackRF One's board. The jammer can switch between RX (receiving) and TX (transmitting) modes. Initially at the system boot, the device is set to RX mode in order to listen to the channel for detecting IEEE 802.11 frames. During the listening phase, the micro-controller periodically captures and processes every $L = 16$ samples at a time. The frame detection efficiency heavily relies on the computation of auto-correlation A_n and energy E_n of the chunks of samples. Using the SIMD instructions supported by the ARM Cortex-M4 processor, we achieve both A_n and E_n in a total 128 CPU cycles. As the HackRF's micro-controller is set to run at $204\,\mathrm{MHz}$, it takes roughly $0.64\mu s$ to process each 16 samples, or $0.04\mu s$ per sample. Combining with the remaining operations in the flow chart, the frame detection's running time is close to $0.05\mu s$ per sample, which is the upper-bound required for real-time processing Wi-Fi signals transmitted at $20\,\mathrm{MHz}$.

When a frame is detected, the device switches to the TX mode for transmitting the jamming signal. As interleaving jamming is a multi-carrier jamming attack, the interference is generated in the frequency domain such that each selected subcarrier contains random noise (while the rest are nulled), then the signal is transformed into the time domain for transmitting. To avoid the FFT computation burden, we do not generate the interference on the fly. Instead, we store pre-generated samples in memory and subsequently use them for jamming. Furthermore, we also optimize the TX/RX switching operation in order to minimize the switching time and improve the responsiveness of the jammer. To destroy a frame, it is sufficient to jam a small portion of the frame. The length of the jamming burst, denoted B, is configurable by the adversary from the PC. Immediately after completing the transmission of the B multi-subcarrier interference samples, the jammer turns back to the RX mode to detect the next frames.

Table 2: Jamming attacks investigated in our evaluation.

Attack	Jammed subcarriers
Single jamming	1 single DSC
Range jamming	Set of adjacent DSCs
Whole-band jamming	Whole Wi-Fi channel
Pilot jamming	Pilot subcarriers
Interleaving jamming	Multiple DSCs (cf. Definition 1)

5. PERFORMANCE EVALUATION

To evaluate the efficiency of the interleaving jamming attacks on Wi-Fi communications, we compare it with other types of multi-carrier jamming attacks (cf. Table 2), which are categorized based on the number and pattern of subcarriers selected by the adversary.

Our metrics for the attack efficiency is the Packet Error Rate (PER) observed at the receiver. For a fair comparison between jamming strategies, the packet error rate is evaluated based on the transmitted signal to jamming power ratio (SJR) as measured on the receiver. Our general setup consists of a pair of transmitter and receiver, which are desktop computers equipped with off-the-shelf Wi-Fi cards. The transmitter constantly sends UDP packets to the receiver. The jammer sits nearby to monitor the channel and accordingly jam all detected packets. All experiments are carried out on a $20\,\mathrm{MHz}$ communication of channel 11.

The performance of Wi-Fi communications is dependent on various factors such as the interference level, transmission rate, and channel access mechanisms. To quantify the impact of jamming only, we control the testbed experiments as follows. First, we verify that the natural noise in the environment is at least $40\,\mathrm{dB}$ lower than the Wi-Fi transmitted signal power, therefore it barely affects the reception rate at the receiver. We use the performance obtained in normal conditions without jamming as the baseline for comparison of jamming efficiency. Specifically, if T_{JAM} and T_{NOJAM} denote the number of correct packets seen at the receiver under jammed and unjammed conditions, the PER is calculated as $\mathrm{PER} = \frac{T_{\mathrm{JAM}}}{T_{\mathrm{NOJAM}}}$.

As adaptive transmission rates can result in instable throughput and error rate, we disable the rate adaptation and set a constant transmission rate for each experiment. At the receiver, we run Wireshark in monitor mode to obtain the Physical and MAC layer information of the received packets. In monitor mode, however, the receiver might observe multiple copies of a frame due to unicast re-

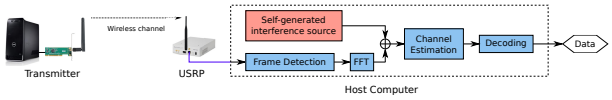


Figure 6: Self-jamming setup: Transmitter (D-Link WDA-1320 Wi-Fi adapter) broadcasts packets on the wireless channel. The receiver is a USRP connected to a Host Computer for receiving transmitted packets with self-generated interference.

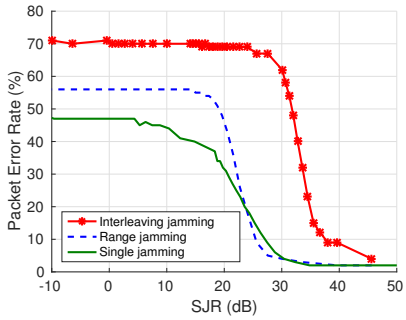


Figure 7: Impact of narrow-band jamming: Single jamming on DSC 0; Range jamming on DSC 0, 1, 2; Interleaving jamming on DSC 0, 3, 6.

transmissions[2], leading to inaccurate throughput results. To avoid this issue, we configure the transmitter to send packets in broadcast mode, therefore disabling the retransmissions.

In each experiment, we gradually decrease the jamming power, while fixing the transmitter's power at a regulated transmit power, to have the SJR varying between $0\,\mathrm{dB}$ and $50\,\mathrm{dB}$. For each value of SJR, results are collected every $1\,\mathrm{s}$ in the total duration of $10\,\mathrm{s}$ per run, and the mean PER is accordingly computed. We note that since UDP transmissions have no flow control and are broadcast at fixed rate, the short duration of $10\,\mathrm{s}$ per run is sufficient for us to obtain stable results. In the following, we study the jamming impact in different scenarios, from ideal to realistic jamming attacks.

5.1 Preliminary Results on SDR Receiver

We consider an ideal jamming signal generator, in which no real jammer is running, but the receiver jams itself during the packet reception process. The motivation and implications of this model is that the constraints of a practical adversary (e.g., timing, energy, detection accuracy, hardware capability, etc.) can be eliminated, with the additional advantage of being able to repeat experiments and apply different jamming techniques to exactly the same received RF signals. Specifically, in this scenario (cf. Figure 6), while the transmitter remains the same as in the general setup (i.e., transmitter is a commercial Wi-Fi card), the receiver is an SDR Wi-Fi receiver that we have developed [39] on a USRP device [10]. It is also able to inject self-generated interference into the samples sequence (received over the air) before decoding the data. In [39], we verified that this custom receiver has a reception performance comparable to commercial Wi-Fi cards, therefore allowing us to readily evaluate the impact of jamming using this self-jamming setup. In this first set of experiments, we transmit 1500-byte UDP packets at a fixed rate of $54\,\mathrm{Mbps}$. The self-generated interference is added to the whole duration of each packet, except for Section 5.1.4, where we only add the interference to a few first OFDM symbols of each packet. We call the former *long-burst* jamming, and the latter *short-burst* jamming. To refer to the burst length, we use the notation s as

[2]In IEEE 802.11 MAC protocol, up to 12 retransmissions are triggered for an unacknowledged frame in unicast mode.

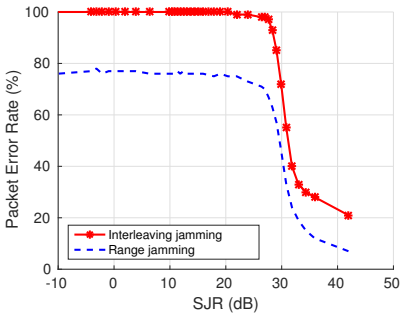


Figure 8: Impact of jamming with 7 subcarriers: Range jamming on DSC 0 to 6; Interleaving jamming on DSC $0, 3, 6, 9, 12, 15, 18$.

the size of burst in number of OFDM symbols, which is computed by $s = B/80$, where B is the burst length in number of samples, previously introduced in Section 4.2. In Sections 5.1.1 to 5.1.3, long-burst jamming is considered with burst length corresponding to $s = 58$ (covering the whole packet).

5.1.1 Narrow-band Jamming

First, we evaluate the impact of narrow-band jamming, where the jamming signal covers only a few subcarriers. Three types of attacks are considered: (a) *Single-carrier jamming* at DSC 0, (b) *Range jamming* at DSC $0, 1, 2$, and (c) *Interleaving jamming* at DSC $0, 3, 6$. The impact on performance of the Wi-Fi link between the transmitter and receiver is shown in Figure 7. We note that to achieve PER of 40%, the Single-carrier jamming attack requires at least $15\,\mathrm{dB}$ more jamming power in comparison with the Interleaving jamming strategy. Interestingly, the Range jamming strategy creates slightly less harm (PER 10%) than Single-carrier jamming when the SJR is higher than $25\,\mathrm{dB}$. It can be explained by the fact that spreading the jammer power over multiple subcarriers weakens the interference in individual subcarriers. In this case, since the jammer is not aware of the interleaving pattern, these low-power individual jamming subcarriers cannot effectively cooperate to destroy the packets. In contrast, with the same low power constraint, the Interleaving jamming is able to corrupt 70% of transmitted packets at the same SJR of $25\,\mathrm{dB}$.

Now we look at the lower SJR conditions (less than $20\,\mathrm{dB}$). Although all three jamming strategies have considerable impacts on the PER (more than 40%), there are specific PER thresholds such that a higher degradation of performance cannot be achieved by increasing jamming power. This is explained by the narrowband jamming constraint, which leaves a large enough portion of data subcarriers intact so that the depth-6 convolutional code is able to correct the errors introduced by the interference. In this experiment, the PER threshold for Single-carrier jamming, Range jamming, and Interleaving jamming are 48%, 56%, 70%, respectively.

Aiming to achieve higher jamming impact, we configure the jammer to jam on more subcarriers. In particular, we compare the Range jamming and Interleaving jamming with 7 subcarriers, where the former attack jams on DSC 0 to 6, while the latter jams on DSC $0, 3, 6, 9, 12, 15, 18$. Figure 8 shows that at high SJR around $30\,\mathrm{dB}$, there is a little difference (roughly $2\,\mathrm{dB}$) in the required jamming power between Range jamming and Interleaving jamming strategies that block up to 50% of packets. In contrast to the previous jamming attacks with 3 DSCs, there is now no clear advantage of Interleaving jamming over Range jamming at low-power jamming. The reason is that on one hand the Range jamming on DSC 0 to 6 now also covers DSC $0, 3, 6$, so it can effectively destroy three consecutive bits in the original data sequence, thus creating more

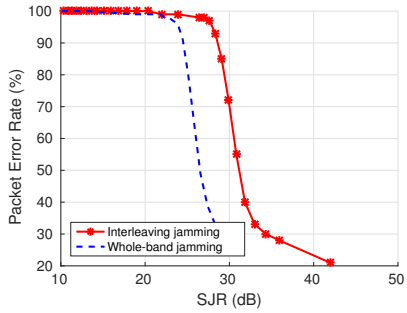


Figure 9: Comparison between Interleaving jamming on DSC $0, 3, 6, 9, 12, 15, 18$ and Whole-band jamming on 20 MHz of Wi-Fi channel.

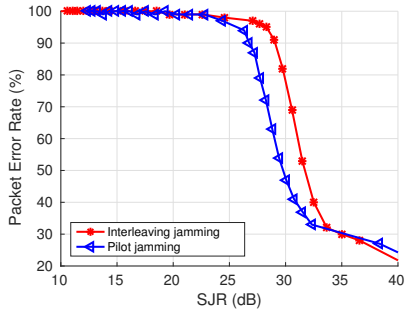


Figure 10: Comparison of performance impact by jamming on pilot subcarriers and Interleaving jamming on 7 data subcarriers.

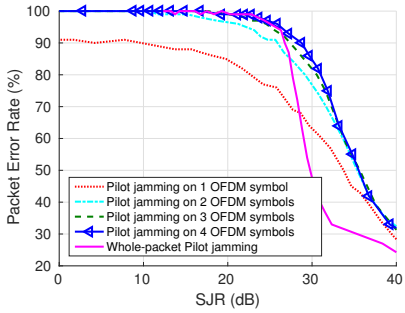


Figure 11: Impact of short-burst jamming on pilot subcarriers.

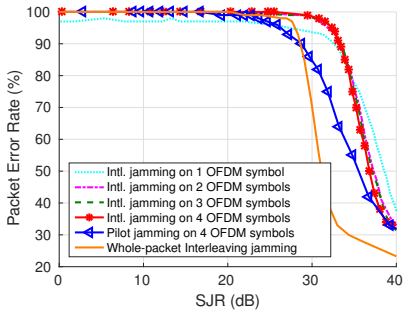


Figure 12: Impact of short-burst Interleaving jamming.

impact than the previous Range jamming on DSC 0 to 2. On the other hand, Interleaving jamming with expanded number of DSCs from 3 to 7 only adds little impact due to low power constraint. However, at SJR lower than $20\,\mathrm{dB}$, the Interleaving jamming can now destroy all transmitted packets, while the Range jamming still lets 20% of packets through. This implies the superiority of Interleaving jamming in completely blocking Wi-Fi packets.

5.1.2 *Whole-band vs. Interleaving Jamming*

To further understand and quantify the efficiency of Interleaving jamming, we compare its performance to the Whole-band jamming. The results in Figure 9 show that to achieve the same jamming impact, the Whole-band jamming requires about $5\,\mathrm{dB}$ more power than the Interleaving jamming on DSC $0, 3, 6, 9, 12, 15, 18$.

5.1.3 *Jamming on Pilot Subcarriers*

In IEEE 802.11, pilot subcarriers are located among the data subcarriers, and used for channel estimation and equalization. Interference mitigation for pilot subcarriers is, therefore, very important for the robustness of OFDM systems [9, 26, 34]. In this subsection, we compare the impact of Pilot jamming and Interleaving jamming. For Pilot jamming, all four pilot subcarriers are jammed. For Interleaving jamming, we select to jam on DSC $0, 3, 6, 9, 12, 15, 18$ similarly as in previous experiments. Figure 10 shows that Pilot jamming results in a slightly less impact (roughly $2\,\mathrm{dB}$ less power efficiency) than Interleaving jamming. It is noted that both are more efficient than Range jamming and Whole-band jamming.

5.1.4 *Short-burst Jamming*

We have so far investigated long-burst jamming scenarios. To obtain more insight about the effectiveness of the interleaving jamming strategy, we now perform another series of experiments, in which the subcarriers are jammed for a short duration spanning few OFDM symbols within each packet. In this subsection, we study the two most efficient jamming strategies observed above: short-burst Pilot jamming and short-burst Interleaving jamming.

First, we compare the performance of the Wi-Fi transmissions under Pilot jamming of different jamming burst lengths. Specifically, the first s OFDM symbols within every packet are jammed, with $s = 1, 2, 3, 4$ for short-burst jamming scenarios and $s = 58$ for whole-packet jamming. We can see from Figure 11 that jamming only one OFDM symbol of every packet appears as the least efficient burst, while jamming the whole packet does not result in high efficiency. We find that Pilot jamming on $3, 4$ OFDM symbols results in the highest attack efficiency, which can corrupt over 80% of the packets at SJR $= 30\,\mathrm{dB}$ and 99% at SJR $= 20\,\mathrm{dB}$.

Now we carry out a similar experiment to study the impact of the short-burst Interleaving jamming strategy, in which burst lengths $s = 1, 2, 3, 4$ are considered. For the sake of comparison, the packet error rates caused by the Interleaving jamming on the whole packet and Pilot jamming on the first 4 OFDM symbols are also included in the results shown in Figure 12.

First, we observe that in contrast to short-burst Pilot jamming strategies, the short-burst Interleaving jamming strategies can block over 95% of packets by using a burst of only 1 OFDM symbol. Interestingly, this attack uses a jamming power as low as 0.1% ($30\,\mathrm{dB}$ less than) the transmitted signal power. Moreover, when the burst length is increased to span the duration of 2, 3, or 4 OFDM symbols, the adversary can block 99% of the transmitted packets at the jamming power level of 0.1% of the transmitter's power. In comparison with short-burst Pilot jamming, the short-burst Interleaving jamming is at least $5\,\mathrm{dB}$ more power efficient at PER of 90%. The gap of $5\,\mathrm{dB}$ is also observed at all PERs when compared to whole packet Interleaving jamming.

In summary, our preliminary results on the Wi-Fi SDR-based receiver setup show that the most efficient jamming attack against Wi-Fi communications is the short-burst Interleaving jamming.

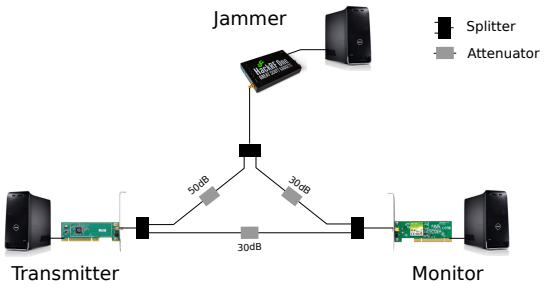


Figure 13: Experiment setting: (a)TX: D-Link WDA-1320; (b)RX: TP-Link TL-WN751ND; (c)Jammer: HackRF

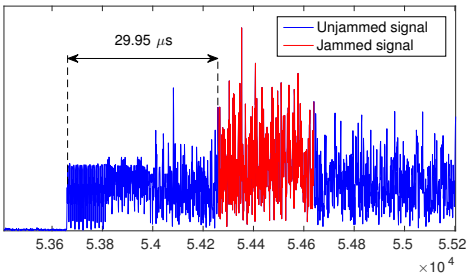


Figure 14: HackRF jammer's response time is $29.95\mu s$.

5.2 Impact of Practical Jammer on Commercial Wi-Fi Cards

Based on the insights gained in Section 5.1, we now evaluate the impact of Interleaving jamming attacks on commercial Wi-Fi cards. In this case, the jammer is implemented on the HackRF One SDR firmware (as described previously). The HackRF device continuously listens to the Wi-Fi channel and jams every detected packet. First, we carry out the experiments in a testbed with controlled attenuation in order to minimize interference from external sources such as other simultaneously ongoing Wi-Fi communications. Our testbed, illustrated in Figure 13, consists of three nodes: the transmitter broadcasting data, the receiver operating in monitor mode for measuring the reception performance, and the jammer running on the HackRF One device. Both the transmitter and receiver consist of commercial off-the-shelf D-Link WDA-1320 Wi-Fi adapters. Three nodes are connected by a triangular topology with attenuators that emulate the path loss in a typical wireless channel. Similarly to Section 5.1, 1500-byte packets are broadcast at 54 Mbps on 20 MHz of channel 11, unless otherwise stated. For each experiment, the configurations of jamming attack (including the number and pattern of jammed subcarriers, jamming power, burst length) are controlled from the PC connected to the HackRF One through a USB port. We emphasize that this USB connection is only used for configuring the jammer. The detection and jamming tasks are handled in real-time by the HackRF itself. Our metric is again the PER with respect to the SJR, in which the latter is computed based on the received signal power obtained at every value of the HackRF One's transmit gain. To verify that these experimental results are not specific to this D-Link card, we also changed the receiver to use a different Wi-Fi adapter manufacturer/model, TP-Link TL-WN751ND, and repeated all experiments. We observe that the results obtained in the two sets of experiments are very similar. Therefore, in the subsequent subsections, we report the results from the TP-Link TL-WN751ND receiver only (for graph readability).

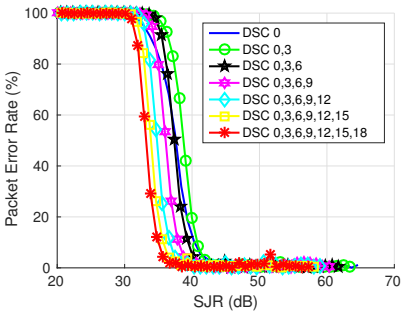


Figure 15: Effect of number of DSCs on attack efficiency of short-burst Interleaving jamming.

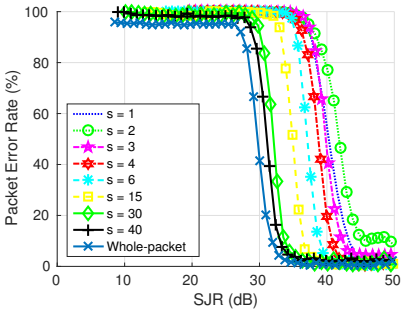


Figure 16: Interleaving jamming with different burst lengths s (measured as the duration of s OFDM symbols).

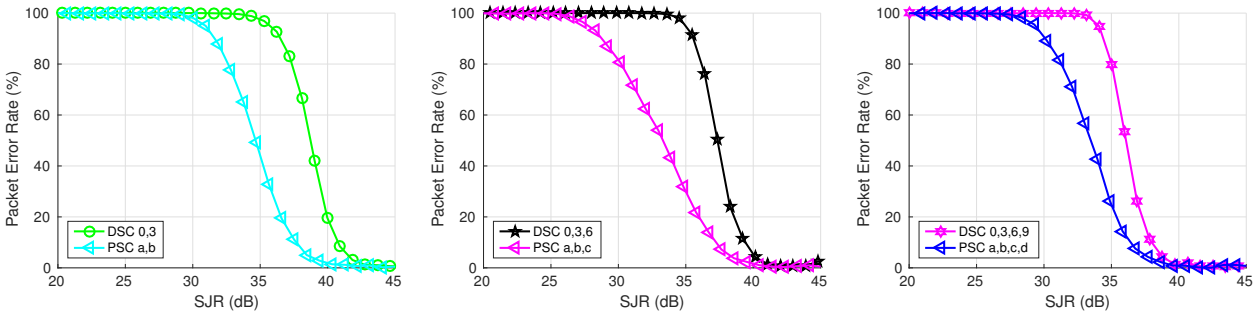
5.2.1 *Response Time*

In practice, the attack efficiency does not only depend on the detection capability, the jamming pattern and power, but also on the response time of the jammer, which is the duration between the time of receiving the first sample of a frame and the time of emitting the first jamming sample. To evaluate the response time of the HackRF jammer, we configure the jammer to apply the maximum power and use our custom Wi-Fi receiver as described in Section 5.1 to receive the transmitted packets. We note that in this experiment, self-jamming is disabled and the receiver simply captures all received samples. For each detected frame, we locate the first jamming sample by looking at an increase in the signal envelope.

Figure 14 illustrates this process, and shows that the response time is $29.95\mu s$, approximately the duration of 7.5 OFDM symbols. By repeating the experiment, we find that the average response time is around $30\mu s$ with insignificant variance. Furthermore, it is also invariant to all attack configurations tested in our evaluation. Since each frame contains a preamble of 4 OFDM symbols and a Physical header of 1 OFDM symbol, this response time implies that the jammer can successfully jam a MAC frame of at least 3 OFDM symbols. Our rough estimation suggests that a small UDP packet containing only 6 bytes of payload and 64 bytes of header (including UDP, IP, LLC, MAC headers) can be jammed by our HackRF implementation if the packet is transmitted at 54 Mbps. In case of TCP transmissions, the hit probability is even higher, because small data chunks are typically combined into one big chunk (Nagle algorithm). Consequently, our jammer can effectively destroy most of the traffic in Wi-Fi networks.

5.2.2 *Effect of Number of Subcarriers*

We evaluate the effectiveness of Interleaving jamming by the number of DSCs selected for jamming. The jamming pattern is configured to $0, 3, \ldots, 3(n-1)$ for $n = 1 \ldots 7$. The adversary



(a) DSC $0, 3$ vs. PSC a, b. (b) DSC $0, 3, 6$ vs. PSC a, b, c. (c) DSC $0, 3, 6, 9$ vs. PSC a, b, c, d.

Figure 17: Comparison between Interleaving jamming and Pilot jamming (both have short burst length $s = 4$).

uses a fixed burst length of $s = 4$ OFDM symbols. Figure 15 shows that jamming on two DSCs 0 and 3 is the most effective attack that can create a PER of 20% at SJR of $40\,\text{dB}$ (i.e., the jamming power is only 0.01% of the transmitted signal power). When the SJR decreases to $35\,\text{dB}$ (i.e., the jamming power increases to 0.03% of the transmitter's power), the PER rises to 95%. We also observe that jamming on more DSCs tends to decrease the attack efficiency, because less power is distributed to each individual subcarrier. Nevertheless, even the weakest jamming pattern with 7 DSCs can still block all packets (PER $= 100\%$) at SJR $= 30\,\text{dB}$.

5.2.3 *Short-burst Interleaving Jamming*

In this experiment, we study the impact of the jamming burst length on the performance of the Wi-Fi link. We fix the pattern of jammed subcarriers to be DSC $0, 3$ and vary the burst length from $s = 1$ OFDM symbol to $s = 58$ (the total number of OFDM symbols contained in a 1500-byte packet transmitted at $54\,\text{Mbps}$). It is seen in Figure 16 that the burst length of $s = 2$ results in the most powerful jamming attack, whereas extending the jamming period gradually reduces the efficiency. We notice that the efficiency gap between the best and worst attacks is up to $12\,\text{dB}$. These results also match with those of SDR-receiver self-jamming experiments (Figure 12), in which $s = 2$ appears to be the most efficient one. This indicates that very short jamming bursts are sufficient for an effective attack, whereas jamming in more time simply wastes the energy without achieving more impact.

5.2.4 *Interleaving vs. Pilot Jamming*

Recall that in Section 5.1, we discovered that the two most effective attacks among those investigated are Interleaving jamming and Pilot jamming, among which the former outperforms the latter by around $2\,\text{dB}$ in case of whole-packet jamming, and around $5\,\text{dB}$ in case of short burst jamming. To see whether the advantage of Interleaving jamming holds with the HackRF jammer against commercial receivers, we compare these two attacks in different scenarios, where we change the number of jammed subcarriers from 2 to 4. The following cases are evaluated: (a) DSC $0, 3$ vs. PSC a, b; (b) DSC $0, 3, 6$ vs. PSC a, b, c; (c) DSC $0, 3, 6, 9$ vs. PSC a, b, c, d, where the locations of pilot subcarriers (PSC) are specified by IEEE 802.11 as in Table 3.

Table 3: Pilot subcarrier (PSC) and data subcarrier (DSC) locations.

Pilot subcarrier	Location
PSC a	between DSC 4 and 5
PSC b	between DSC 17 and 18
PSC c	between DSC 29 and 30
PSC d	between DSC 42 and 43

Figure 17 shows that Interleaving jamming attacks outperform Pilot jamming, despite the fact that both jam on the same number of subcarriers. At PER around 95%, the efficiency gap is at least $10\,\text{dB}$ (Figures 17a and 17c) and up to $15\,\text{dB}$ (Figure 17b).

5.2.5 *Impact on Coding Rates*

So far, the experiments we carried out were performed for the transmission rate $54\,\text{Mbps}$, which uses the highest coding rate of $3/4$. In this subsection, we evaluate the jamming impact on lower coding rates of $2/3$ and $1/2$. Specifically, we configure the transmitter to use three different rates: $54\,\text{Mbps}$, $48\,\text{Mbps}$ and $24\,\text{Mbps}$ corresponding to coding rates $3/4$, $2/3$ and $1/2$, respectively (cf. Table 1). In this experiment, we again compare the attack efficiency between Interleaving and Pilot jamming, both of which are short-burst jamming ($s = 4$) on 2 subcarriers (DSC $0, 3$ vs. PSC a, b).

We see from Figure 18 that at any transmission rate, Interleaving jamming is more power efficient than Pilot jamming. While at $24\,\text{Mbps}$ (coding rate of $1/2$), only $2 - 3\,\text{dB}$ are gained by Interleaving jamming, the gap increases to $10\,\text{dB}$ for higher transmission rates. The implication is that when less redundancy is produced in the coded data sequence, Interleaving jamming is very effective to destroy the packets due to the vulnerability of the interleaver structure in IEEE 802.11.

5.2.6 *Over the Air Experimental Results*

In this experiment, we evaluate the impact of Interleaving jamming in an open environment, where the wireless channel can be affected by other factors such as parallel communications, channel distortions, fading, or multipath effect. We perform this experiment by removing all the RF cables and attenuators between all nodes. The transmitter, receiver, and jammer are within $2\,\text{metres}$ range of each other. Since our HackRF jammer does not parse each detected frame, it may also jam packets coming from external Wi-Fi transmitters during the experiment. On the receiver end, however, we only count the correctly received packets originated from our transmitter and use this statistic to compute the results shown in Figure 19. For comparison, we also carry out the experiment for the Pilot jamming strategy. Both attacks are short-burst of length $s = 4$ and jam on two subcarriers: DSC $0, 3$ for Interleaving jamming and PSC a, b for Pilot jamming. The Wi-Fi transmission is configured to operate at a fixed rate of $54\,\text{Mbps}$.

Figure 19 shows that Interleaving jamming destroys about 95% of packets at SJR $= 30\,\text{dB}$, which is a slight drop in comparison with Figure 18, where it blocks all packets at the same SJR. Nevertheless, the Interleaving jamming in wireless environment is still more power efficient than Pilot jamming by roughly $8\,\text{dB}$.

In summary, our practical jammer with Interleaving jamming

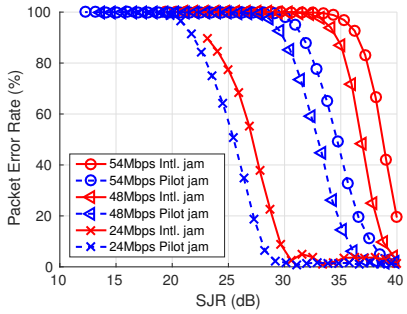


Figure 18: Comparison between Interleaving and Pilot jamming in different coding rates.

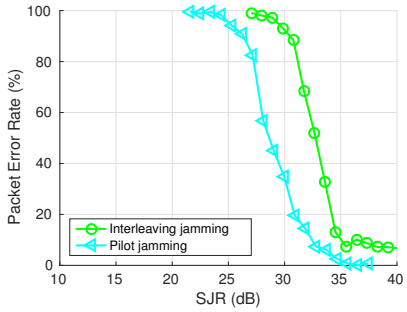


Figure 19: Comparison between Interleaving and Pilot jamming in wireless environment.

strategy is significantly effective against Wi-Fi communications. It can block 95% of transmitted packets by using a jamming power equal to 0.1% of the transmitter's power.

6. COUNTERMEASURES

As Wi-Fi protocol is not designed to combat malicious interference, protecting the communications against Interleaving jamming attacks requires modifications to the standard. One possible approach is to randomize the interleaving mapping cryptographically such that only the transmitter and receiver, who share a common secret, can understand and the de-interleave the received sequence, therefore preventing the adversary from generating jamming patterns that result into interference bursts post de-interleaving [38]. An alternative short-term solution that can reduce the practicality of interleaving jamming consists of making the interleaving structure dependent on the IEEE802.11 frame (e.g., scrambling seed), and permute both over time and frequency subcarriers.

7. RELATED WORK

Over the last few years, the wireless community made significant progress characterizing the potential of smart-jamming attacks against general wireless systems and IEEE 802.11 in particular. A variety of attack and mitigation techniques were developed including reactive jammers [37, 40], channel adaptation [14, 15, 17, 42], keyless spread spectrum [5, 21, 22, 35], broadcast and control channels resiliency [1, 8, 29, 31, 36], MAC resiliency [2, 7, 19, 24], and even communication through silence [30, 41].

In the context of Wi-Fi, previous work demonstrated the feasibility of building reactive jammers [4], understanding IEEE 802.11 MAC and Link layer vulnerabilities [3, 25, 27], and spatial access control [4, 18], but only limited work investigated the vulnerabilities of Wi-Fi that are specific to its physical layer.

The potential existence of interleaving jammers against communication systems was first conjectured in the time domain in our previous work [20]. In the same paper, the jamming efficiency against IEEE 802.11a was estimated for such attacks that target the whole OFDM symbol but did not investigate the unique characteristics of the interleaver. In this work, we demonstrate that interleaving jamming attack is practical in the frequency domain and is even much more power efficient by destroying sub-OFDM symbols with a careful selection of subcarriers.

Other recent works [23, 28, 32] have demonstrated jamming attacks on IEEE 802.11a preambles, which aim to disturb the synchronization mechanism at the receiver, leading to incorrect packet decoding. Based on the reported results in [32], where the optimal frequency offset attack achieved a bit error rate (BER) of 0.5 at short-lived samples' SJR of 1.46 dB, our rough computation suggests that this could be roughly equivalent to blocking all packets at an average SJR of 21.46 dB (for 1500-byte UDP packets). In contrast, the OFDM symbol timing attack [23], which generates fake preambles to deceive the receiver, achieved a similar performance at SJR around 12.5 dB. While it is difficult to give a direct comparison (as the previous work did not report the attack performance in a real Wi-Fi system with important components such as encoding, interleaving), our real Wi-Fi experiments indicate that the proposed interleaving jamming was able to destroy all packets at an average SJR of at least 25 dB and up to 32 dB.

Regarding the timing requirement for the attacks, frequency and timing synchronization jamming requires fast hardware and software solutions in order to perform the responsive jamming within the very short duration of the preambles (e.g., 16 us for 20 MHz), therefore limiting the practicality of such attacks on low cost radios. In contrast, the interleaving jamming can be performed on any sub-OFDM symbol, making the attack easier for the adversary. Another condition for the frequency offset attack [32] is that the adversary needs to measure and estimate the frequency offset between the transmitter and receiver (based on data/ack exchange) before the attack can be performed. Moreover, when multiple transmitter-receiver pairs are present with different frequency offsets, it is difficult for the adversary to properly perform the attack, as the source of the current frame is only known after the MAC header is decoded.

8. CONCLUSION

We devised a new jamming strategy that exploits the IEEE 802.11 interleaving mechanism in order to actively introduce burst errors to the Wi-Fi receiver's convolutional decoder resulting in a significant impact on the Wi-Fi link performance. Our short-burst Interleaving jamming strategy can destroy more than 95% of the transmitted packets by using a jamming power equal to only 0.1% of regular transmitted signal power. When the jamming power is increased to the fraction of 1%, our strategy can completely block all packets. In comparison with jamming strategies that are unaware of the interleaving structure, we can achieve the same jamming impact with at least 5 dB and up to 15 dB more power efficiency. We note that this attack can be combined with other techniques (e.g., targeting the rate adaptation mechanism) for higher efficiency and stealth. We also demonstrated that the Interleaving jamming is practical enough for implementation on a low-cost SDR platform such as the HackRF One.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. NSF/CNS-1409453.

References

[1] G. N. A. Chan, X. Liu and B. Thapa. Broadcast control channel jamming: Resilience and identification of traitors. In *IEEE International Symposium on Information Theory (ISIT)*, 2007.

[2] B. Awerbuch, A. W. Richa, and C. Scheideler. A jamming-resistant MAC protocol for single-hop wireless networks. In *Proceedings of the 27th ACM Symposium on Principles of Distributed Computing*, PODC'08, pages 45–54, 2008.

[3] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa. On the performance of IEEE 802.11 under jamming. In *INFOCOM*, pages 1265–1273, 2008.

[4] D. S. Berger, F. Gringoli, N. Facchi, I. Martinovic, and J. B. Schmitt. Gaining insight on friendly jamming in a real-world IEEE 802.11 network. In *7th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec'14, Oxford, United Kingdom, July 23-25, 2014*, pages 105–116, 2014.

[5] A. Cassola, T. Jin, G. Noubir, and B. Thapa. Efficient spread spectrum communication without pre-shared secrets. *IEEE Transactions on Mobile Computing*, 2013.

[6] A. Cassola, W. Robertson, E. Kirda, and G. Noubir. A practical, targeted, and stealthy attack against wpa enterprise authentication. NDSS, 2013.

[7] S. Chang, Y. Hu, and N. Laurenti. Simplemac: a jamming-resilient mac-layer protocol for wireless channel coordination. In *The 18th Annual International Conference on Mobile Computing and Networking, Mobicom'12, Istanbul, Turkey, August 22-26, 2012*, pages 77–88, 2012.

[8] J. T. Chiang and Y. Hu. Cross-layer jamming detection and mitigation in wireless broadcast networks. In *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking, MOBICOM 2007, Montréal, Québec, Canada, September 9-14, 2007*, pages 346–349, 2007.

[9] A. Coulson. Narrowband interference in pilot symbol assisted ofdm systems. *Wireless Communications, IEEE Transactions on*, 3(6):2277–2287, Nov 2004.

[10] Ettus Research. Universal Software Radio Peripheral.

[11] FCC. FCC enforcement bureau steps up education and enforcement efforts against cellphone and gps jamming, 2013.

[12] FCC. FCC fines jammers, 2013.

[13] FCC. Marriott hotels fined $600,000 by FCC for jamming Wi-Fi hotspots, October 2014.

[14] K. Firouzbakht, G. Noubir, and M. Salehi. On the capacity of rate-adaptive packetized wireless communication links under jamming. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, 2012.

[15] K. Firouzbakht, G. Noubir, and M. Salehi. On the performance of adaptive packetized wireless communication links under jamming. *IEEE Transactions on Wireless Communications*, 13(7), 2014.

[16] Great Scott Gadgets. Hackrf one. https://greatscottgadgets.com/hackrf/.

[17] L. Jia, X. Liu, G. Noubir, and R. Rajaraman. Transmission power control for ad hoc wireless networks: throughput, energy and fairness. In *Proceedings of IEEE Wireless Communications and Networking Conference, 2005*, 2005.

[18] Y. S. Kim, P. Tague, H. Lee, and H. Kim. A jamming approach to enhance enterprise wi-fi secrecy through spatial access control. *Wireless Networks*, 21(8):2631–2647, 2015.

[19] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *INFOCOM*, 2007.

[20] G. Lin and G. Noubir. On link layer denial of service in data wireless lans. *Wireless Communications and Mobile Computing*, 5(3):273–284, 2005.

[21] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. ACSAC'10.

[22] S. Liu, L. Lazos, and M. Krunz. Time-delayed broadcasting for defeating inside jammers. *IEEE Trans. Dependable Sec. Comput.*, 12(3):351–365, 2015.

[23] C. Mueller-Smith and W. Trappe. Efficient ofdm denial in the absence of channel information. In *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pages 89–94, Nov 2013.

[24] R. Negi and A. Perrig. Jamming analysis of MAC protocols. Technical report, Carnegie Mellon University, 2003.

[25] G. Noubir, R. Rajaraman, B. Sheng, and B. Thapa. On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming. In *Proceedings of the fourth ACM conference on Wireless network security*, WiSec '11, pages 97–108, New York, NY, USA, 2011. ACM.

[26] S. Ohno, E. Manasseh, and M. Nakamoto. Preamble and pilot symbol design for channel estimation in ofdm systems with null subcarriers. *EURASIP Journal on Wireless Communications and Networking*, 2011(1), 2011.

[27] C. Orakcal and D. Starobinski. Jamming-resistant rate adaptation in wi-fi networks. *Performance Evaluation*, 75-76, 2014.

[28] M. J. L. Pan, T. C. Clancy, and R. W. McGwier. Phase warping and differential scrambling attacks against ofdm frequency synchronization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2886–2890, May 2013.

[29] R. D. Pietro and G. Oligeri. Freedom of speech: thwarting jammers via a probabilistic approach. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, New York, NY, USA, June 22-26, 2015*, pages 4:1–4:6, 2015.

[30] R. D. Pietro and G. Oligeri. Silence is golden: Exploiting jamming and radio silence to communicate. *ACM Trans. Inf. Syst. Secur.*, 17(3):9:1–9:24, 2015.

[31] C. Pöpper, M. Strasser, and S. Capkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *IEEE Journal on Selected Areas in Communications*, 28(5):703–715, 2010.

[32] H. Rahbari, M. Krunz, and L. Lazos. Security vulnerability and countermeasures of frequency offset correction in 802.11a systems. In *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, pages 1015–1023, 2014.

[33] B. Schneier. Fake cell phone towers across the US, Sep. 2014.

[34] A. Stamoulis, S. Diggavi, and N. Al-Dhahir. Intercarrier interference in mimo ofdm. *Signal Processing, IEEE Transactions on*, 50(10):2451–2464, Oct 2002.

[35] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pages 64–78, 2008.

[36] P. Tague, M. Li, and R. Poovendran. Mitigation of control channel jamming under node capture attacks. *IEEE Trans. Mob. Comput.*, 8(9):1221–1234, 2009.

[37] T. D. Vo-Huu, E.-O. Blass, and G. Noubir. Counter-jamming using mixed mechanical and software interference cancellation. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '13, pages 31–42, New York, NY, USA, 2013. ACM.

[38] T. D. Vo-Huu and G. Noubir. Mitigating rate attacks through crypto-coded modulation. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, pages 237–246, New York, NY, USA, 2015. ACM.

[39] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir. Swifi: An open source sdr for wi-fi networks high order modulation analysis. Technical report, 2015.

[40] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. Short paper: reactive jamming in wireless networks: how realistic is the threat? In *Proceedings of the Fourth ACM Conference on Wireless Network Security, WISEC 2011, Hamburg, Germany, June 14-17, 2011*, pages 47–52, 2011.

[41] W. Xu, W. Trappe, and Y. Zhang. Anti-jamming timing channels for wireless networks. In *Proceedings of the First ACM Conference on Wireless Network Security, WISEC 2008, Alexandria, VA, USA, March 31 - April 02, 2008*, pages 203–213, 2008.

[42] W. Xu, W. Trappe, and Y. Zhang. Defending wireless sensor networks from radio interference through channel adaptation. *TOSN*, 4(4), 2008.