OBFUSCATION OF TRANSMISSION FINGERPRINTS FOR SECURE WIRELESS COMMUNICATIONS

by

Hanif Rahbari

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2016

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Hanif Rahbari
entitled Obfuscation of Transmission Fingerprints for Secure Wireless Communications
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 18 December 2015
Marwan Krunz

Date: 18 December 2015
Loukas Lazos

Date: 18 December 2015
Ming Li

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 18 December 2015
Dissertation Director: Marwan Krunz

STATEMENT BY AUTHOR

SIGNED: Hanif Rahbari

ACKNOWLEDGEMENTS

There are a number of people I would like to express my gratitude to for their help and support during my time in graduate school.

First of all, I would like to express my deepest gratitude to my advisor, Professor Marwan Krunz, for his patience and all the help, care, advice, supports and encouragements he gave to me during the past five years. Because of him, my graduate experience has been the one that I will cherish forever. I am fortunate to have him as my advisor who taught me how to think out of the box and conduct meaningful research. Through his professional experience, he has broaden my understanding of the research world and about where to explore and what is necessary to get there. This dissertation would have been impossible without his guidance. Thank you.

I would like to thank Professor Loukas Lazos for his supports and collaborations, and all his insightful suggestions and discussions that made this dissertation better. I also would like to thank Professor Ming Li for his support and important discussions that helped me improve this dissertation, and Professors Tamal Bose and Wei Hua Lin for their suggestions.

I would like to thank all my former and current labmates in the wireless networking group for their help and invaluable friendship. I wish to give special thanks to Rashad Eletreby for his collaborations on part of this work, and to Dr. Diep Nguyen, Dr. Mohammad J. Abdel-Rahman, Wessam Afifi, and Peyman Siyary for their valuable discussions and generous helps. I would also thank Tami Whelan for handling all the paperwork and giving various forms of support during my graduate study.

Last but not least, I would like to thank my parents and my dear wife Ghazal Dehghani for supporting and encouraging me throughout my Ph.D. study. Their love and care gave me strength to overcome difficulties throughout this endeavor.

DEDICATION

This dissertation is dedicated to my wife Ghazal Dehghani, who has been proud and supportive of my work with a lot of patience and who has shared the many uncertainties, challenges and sacrifices for completing this dissertation. I am truly thankful for having you in my life. This work is also dedicated to my parents: my mother, M. Zahabioun, who endured the six years I have been studying abroad and who supported me throughout her life by every means, specially by providing excellent educational environment for me; and my father, A. Rahbari, who has been my role-model for honesty, persistence, and personal sacrifices, and who instilled in me the inspiration to do works that benefit the people. I am truly blessed for being your son.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Our world of people and objects is on the verge of transforming to a world of highly-interconnected wireless devices. Incredible advances in wireless communications, hardware design, and power storage have facilitated hasty spread of wireless technologies in human life. In this new world, individuals are often identified and reached via one or multiple wireless devices that they always carry (e.g., smartphones, smart wearable, implantable medical devices, etc.), and their biometrics identities are replaced by their digital fingerprints. In near future, vehicles will be controlled and monitored via wireless monitoring systems and various physical objects (e.g., home appliance and retail store items) will be connected to the Internet. The list of these changes goes on. Unfortunately, as different aspects of our lives are being immerged in and dependent to wireless devices and services, we will become more vulnerable to wireless service/connection interruptions due to adversarial behavior and our privacy will become more potent to be exposed to adversaries. An adversary can learn the procedures of a wireless system and analyze its stages, and accordingly, launch various attacks against the operations of the system or the privacy of the people. Existing data confidentiality and integrity services (e.g., advanced encryption algorithms) have been able to prevent the leakage of users' messages. However, in wireless networks, even when upper-layer payloads are encrypted, the users' privacy and the operation of a wireless network can be threatened by the leakage of transmission attributes at the physical (PHY) layer. Examples of these attributes are payload size, frequency offset (FO), modulation scheme, and the transmission rate. These attributes can be exploited by an adversary to launch passive or active attacks. A passive attacker may learn about the interests, sexual orientation, political views, and patentable ideas of the user through analyzing these features, whereas an active attacker exploits captured attributes to

launch selective packet jamming/dropping and disrupt wireless services. These call for novel privacy preserving techniques beyond encryption.

In this dissertation, we study the vulnerability of current wireless systems to the leakage of transmission attributes at the PHY layer and propose several schemes to prevent it. First, we design and experimentally demonstrate with USRPs an energy-efficient and highly disruptive jamming attack on the FO estimation of an OFDM system. OFDM is the core multiplexing scheme in many modern wireless systems (e.g., LTE/5G and 802.11a/n/ac) and is highly susceptible to FO. FO is the difference in the operating frequencies of two radio oscillators. This estimation is done by the receiver using the publicly-known frame preamble. We show that the leakage of FO value via the preamble can facilitate an optimally designed jamming signal without needing to know the channel between the transmitter and the legitimate receiver. Our results show that the jammer can guarantee a successful attack even when its power is slightly less than the transmitter's power. We then propose four mitigation approaches against the proposed FO attack.

Next, we consider certain transmission attributes that are disclosed via unencrypted PHY/MAC headers. Example of these attributes are payload size, transmission rate, and MAC addresses. Beyond unencrypted headers, the adversary can estimate the frame size and transmission rate through identifying the payload's modulation scheme and measuring the transmission time. To prevent the leakage of these attributes, we propose Friendly CryptoJam scheme, which consists of three components: First, a modulation-aware encryption scheme to encrypt the headers. Second, an efficient modulation obfuscation techniques. Specifically, the proposed modulation obfuscation scheme embeds the modulation symbols of a frame's payload into the constellation of the highest-order modulation scheme supported by the system. Together with effective PHY/MAC header encryption at the modulation level, the proposed obfuscation scheme hides the transmission rate, payload size, and other attributes announced in the headers while avoiding any BER performance loss. Compared with prior art, Friendly CryptoJam enjoys less complexity and less susceptibility to FO estimation errors. The third component is a novel PHY-level identification method. To facilitate PHY/MAC header encryption when a MAC layer sender identifier cannot be used (e.g., due to MAC address encryption), we propose two preamble-based sender identification methods, one for OFDM and one for non-OFDM systems. A sender identifier is special message that can be embedded in the frame preamble. The extent of the applications of our embedding scheme goes beyond identifier embedding and include embedding part of the data frame, the sender's digital signature, or any meta-data that the sender provides. Our message embedding method can further be used to mitigate the FO estimation attack because the jammer can no longer optimize its jamming signal with respect to a fixed preamble signal. In addition, we considered friendly jamming technique in a multi-link/hop network to degrade the channels of the eavesdroppers and prevent successful decoding of the headers, while minimizing the required jamming power by optimally placing the friendly jamming devices.

CHAPTER 1

Introduction

The proliferation of wireless technologies in various aspects of human life has been phenomenal. By 2015, it is expected that nearly two billion mobile phones and more than 300 thousand tablets will be sold annually worldwide [2], and that the WiFi market size will grow from USD 14.8 billion in 2015 to USD 33.6 billion by 2020. Beyond smartphones and laptops, wireless communications and networking technologies are being used in medical services and devices, vehicle control and vehicle-to-vehicle communications, border control, just to name a few. As we continue to depend on such a rapidly expanding wireless ecosystem, we are challenged with serious threats related to user privacy, data confidentiality, and system availability. Due to the broadcast nature of the wireless medium, user communications are exposed to eavesdropping and privacy attacks. Unauthorized parties equipped with commodity radio hardware can easily eavesdrop on wireless transmissions.

Encryption is the common way for providing message confidentiality and user privacy. For example, at the application layer, encryption algorithms and protocols, such as HTTPS and SSH, provide message confidentiality. At the transport and network layers, the corresponding headers and payloads are encrypted using protocols such as TLS and IPSec. At the data link (MAC) layer, WPA2 is used for 802.11 frames. 3G/UMTS and 4G LTE cellular systems also support message confidentiality through encryption.

Although cryptography and encryption can be used at the upper layers to protect the confidentiality of any protocol data unit (PDU), they are not sufficient to prevent the leakage of *side-channel information* (SCI) at the physical (PHY) layer.

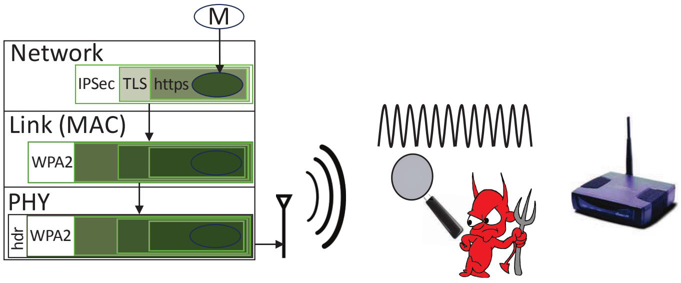


Figure 1.1: Encryption of a protocol's payload (shown with shading) at different layers of the protocol stack. An upper-layer payload and its header are jointly considered the payload for the next lower-layer.

SCI refers to traffic features such as frame type, frame size/duration, modulation scheme, and transmission rate at the frame level and packet size distribution, traffic volume, and inter-packet times at the session level. These features can be determined by eavesdropping on the PHY-layer frame and collectively create a fingerprint of the traffic or the user that generates it. In many wireless systems (e.g., WiFi systems, as standardized by the IEEE 802.11 specifications), parts of the frame (e.g., PHY/MAC headers) must be transmitted in the clear for correct protocol operation and device identification. Specifically, 802.11i/WPA2, the primary security amendment of 802.11, provides confidentiality only for the MAC-layer payload of the data frames and not for PHY/MAC headers (see Figure 1.1). These unencrypted headers can leak certain types of SCI. For example, the PHY header contains the frame size/duration and transmission rate/modulation scheme fields. Parameters such as source and destination MAC addresses, direction of the packet, and packet type (e.g., a retransmission) are specified in the MAC header (see Figure 1.2).

In addition to header fields, certain wireless transmission attributes and radiometric features leak SCI. For example, the modulation scheme used for the frame payload reveals the packet size and the data rate. In digital communications, a bit sequence is modulated into symbols before transmission over the air. The number of possible symbols of a modulation scheme (known as modulation order) relates to

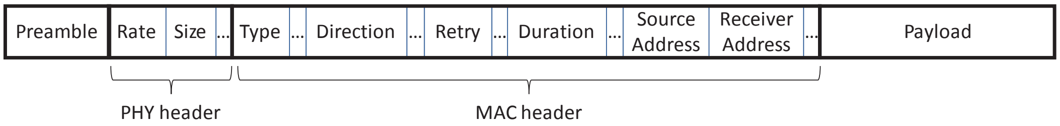


Figure 1.2: Typical 802.11 frame preamble, PHY header, and MAC header.

the number of bits that can be represented by a single symbol. Because of channel noise, symbols must be sufficiently separated so that the legitimate receiver (Rx) can distinguish them from one another. Consequently, a more noisy channel can support fewer bits per symbol, and the transmission of a fixed-size payload can take different durations under different channel conditions. By measuring the frame duration (in seconds) and detecting the modulation scheme, an adversary can estimate the size (in bytes) of the frame payload.

Beside SCI, in many wireless standards, such as 802.11, certain management and control frames are often sent in the clear. Various protocol operations, such as establishing session keys, adjusting the transmission power, and acknowledging the successful reception of a packet, rely on the exchange of these frames.

In the following section, we explain how adversaries can exploit SCI of (encrypted) wireless traffic to launch various attacks against user privacy and functionality of a practical wireless network.

1.1 SCI-based Attacks in Wireless Networks

We classify SCI-enabled attacks into two types: passive and active. Passive attacks refer to SCI analysis performed by an eavesdropper (Eve) to infer private information about a user. Active attacks refer to *selective jamming* of specific packets or parts of a packet, where “significance” of a packet or a part of it is determined based on the consequences of jamming that packet or part. The mechanisms for acquiring and analyzing SCI will be discussed below.

1.1.1 Passive Attacks

The privacy of a wireless user can be violated by overhearing and analyzing encrypted traffic at the PHY layer. We first explain how users can be identified based on device-specific features. We then explain how users' activities can be used to profile and track them.

Device identification and user tracking– Eve can fingerprint a wireless device or its user by exploiting device identifiers embedded in unencrypted headers, the device's intrinsic characteristics and impairments, or captured SCI. Using a device's fingerprint, the adversary can easily track the user's geographical location or determine his online activity. An example of such tracking was demonstrated in a software program called, *Snoopy* [3], which was deployed on a low-altitude flying drone to track users based on their fingerprints, steal their confidential information, or launch a man-in-the-middle attack by spoofing already trusted access points. Snoopy does not require a visual sensor; instead, it uses an antenna to observe WiFi encrypted communications. The globally unique MAC address at the link layer also acts as a plaintext device identifier. The case of bomb-proof trash cans in London a couple of years ago is an example of MAC-address-based tracking. The trash can suppliers had installed a device in the cans to collect information from smartphones of people walking in London's Square Mall, based on the MAC addresses, intending to study people's shopping habits and produce targeted advertisements [4].

The seriousness of these privacy attacks has been recently acknowledged by IEEE and IETF, and accordingly, they formed a new study group to assess the privacy implications of visible MAC addresses and other link-layer privacy issues [5]. To prevent MAC-address-enabled user tracking, this group suggested using randomly generated MAC addresses. Other similar approaches were proposed in [6–8] that are based on a chain of unpredictable but unencrypted time-rolling identifers (e.g., MAC addresses). Although such a radical approach faces several hurdles in the current systems and can take years to finalize [9], it is not adequate for solving the problem of unencrypted PHY headers and the leakage of SCI.

In fact, background activities of Apps installed on a smart phone/tablet or the specific implementation of the wireless card driver can be used to construct a fingerprint. For instance, Eve can create a device-specific traffic fingerprint by analysing the SCI of software programs running for 6 hours in the background of a 3G smartphone [10]. This is because more than 70% of a smartphone's traffic is independent of user interactions and depends only on installed Apps. In fact, by monitoring 15 minutes worth of traffic, the authors in [10] show that it is possible to identify a particular device with 90% success rate among 20 devices running different sets of Apps. Similarly, traffic statistics can characterize an 802.11 device with high probability [11], or the Apps each individual smartphone user in the vicinity is using [12]. Apart from Apps/user-generated traffic, different wireless card vendors often have different implementations of the same protocol on their cards, resulting in vendor-specific inter-frame times, medium access wait (backoff) times, and transmission times [11]. Together, these parameters constitute a vendor-specific fingerprint of the device.

Beside traffic statistics, hardware-specific and electromagnetic characteristics of an RF emitter form a "radiometric" identity of a particular transmitter (Tx). The analog components in a wireless card (e.g., oscillator, baseband filter, amplifier, and antenna) exhibit inherent manufacturing impairments that differ from one card to another. Small variations in these components create distinct artifacts in the emitted signal (e.g., frequency offset and amplitude clipping). The distortions in the captured modulation symbols due to hardware impairments can be exploited to detect a signal's originating device [13].

User's activities and browsing interests– An eavesdropper can also exploit SCI to discern the online activities of a user, his interests, or his search queries [12, 14–17]. For example, through captured SCI, Eve can identify not only the website that a user is browsing, but also the currently active page within a specific website [16].

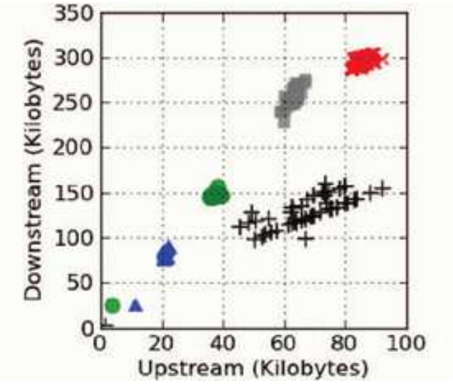


Figure 1.3: Distribution of traces of five websites with respect to uplink/downlink traffic volume, where each symbol of the same shape and color represents the same web page [15].

A typical website is characterized by a nominal uplink/downlink traffic volume and session duration. These coarse-grain traffic features are sufficient to classify websites [15]. In Figure 1.3, an example of such website identification based on only uplink/downlink traffic volume is shown. Even within a given website in which different pages (e.g., company products) are targeted to different users, analyzing the packets size distribution allows for identifying a specific page. As a result, the attacker may be able to conclude, for example, the user's product of interest and may overwhelm him with many commercial ads.

The leakage of private information is not limited to online browsing. An adversary can determine with 80% accuracy the type of user activity (gaming, video streaming, Skype, browsing, etc.) by only eavesdropping for 5 seconds on that user's WiFi traffic [16]. Differences in the traffic statistics of different applications are often large enough to distinguish these applications. Further, the adversary can find out the language used in an encrypted instant messaging application or the user's specific actions during an activity, such as posting a status on Facebook or opening a chat window in Gmail, based on the statistics of the sequence of user-generated packets. Along the same lines, tracking the traffic of two users can reveal if they are communicating with each other.

The sizes (in bytes) and directionality (uplink/downlink) of a sequence of packets exchanged between a mobile user and an access point can also reveal what the user is searching for. Google, Bing, and other search engines offer users suggestions for a searched phrase. This is known as the *auto-suggestion* feature. When a user types the first letter of a keyword, the search engine quickly responds with a list of suggested words. Typing the second letter updates the list of suggestions, and so on. The size of the packet that contains the list of suggestions is highly correlated with the typed letters [18]. Eve can construct a table of different keywords and associate them with the sizes of per-keystroke suggested lists. She can then match the sizes of an observed sequence of packets to one of the entries in the table, and determine the queried word [18]. Even the message length and the language used in an encrypted instant messaging application can be determined based on packet sizes only.

1.1.2 Active Attacks

Besides breaching user privacy, captured SCI can be used by malicious attackers to disrupt communications by selectively jamming wireless transmissions and preventing correct decoding at the receiver (Rx). Jamming includes random attacks, persistent attacks (barrage jamming), and smart/selective attacks in which only a certain packets or parts of a particular packet are jammed. In selective (reactive) jamming, a packet (or part of it) is selected for jamming based on the amount of disruption caused by not delivering this packet to its intended Rx. For example, TCP Acknowledgement (ACK) packets are much shorter in duration than TCP data packets, but are critical for maintaining high TCP throughout by preventing a significant reduction in the congestion window size. Jamming these packets requires less energy than jamming a data packet. At the same time, it can deceive the TCP sender into thinking that the last data packet was not successfully received due to network congestion. Consequently, the sender may unnecessarily reduce its packet transmission rate and retransmit the last packet, which was already received correctly. The attacker can identify the TCP ACK by analyzing the sequence of inter-arrival times and packet sizes. In the case of link-layer ACK packets, the packet type can be identified by inspecting the unencrypted MAC header (see Figure 1.2).

Unencrypted PHY-layer header fields can be intercepted and used to detect and jam data packets transmitted at high rates. Wireless devices adapt their transmission rates based on channel conditions. A good channel prompts the Tx to use a high-order modulation scheme, hence a high data rate. When a packet is not successfully received, the Tx attributes that to channel conditions and accordingly retransmits the packet at a lower rate. This can be exploited by the attacker to jam only high data-rate packets, prompting the Tx to reduce its rate and waste communication resources [19].

In addition to the PHY header, the modulation scheme of the PHY-frame payload may disclose the transmission rate. A modulation scheme is usually associated with two or three data rates of different code rates. For example, in 802.11a, 16-QAM is used for data rates 24 and 36 Mbps. Hence, by determining the modulation scheme, it is rather easy for the adversary to guess the data rate. Combining this rate with the frame length, one can compute the payload size. The frame preamble can also be exploited to detect the arrival of a packet and launch reactive attacks. This preamble is a publicly known signal, prepended to the beginning of a frame to help the Rx detect the frame and estimate various communication parameters (e.g., frequency offset, channel response, etc.). Correct decoding of a frame depends on correct estimation of these parameters. Once a frame is detected, an attacker can jam a vulnerable part of the preamble to disrupt the parameter estimation functions at the Rx [20]. We will discuss the functions of the frame preamble in Chapter 2.

1.2 Main Contributions

To supplement upper-layer security mechanisms and prevent the leakage of SCI at the PHY-layer, Tx and Rx need to employ lightweight PHY-layer security mechanisms. In this dissertation, we adopt a PHY-layer security approach and offer the following contributions:

1.2.1 Design and Implementation of an FO Estimation Attack

Frequency offset (FO) is the difference in the operating frequencies of two radio oscillators, which is resulted from inherent impairment of these oscillators. It is one of the transmission features as it is hardware specific and can be estimated by the Rx using the publicly known frame preamble. In OFDM-based systems (e.g., LTE and 802.11a/g/n/ac), erroneous estimation of the FO at the Rx can be critical, as it results in subcarriers orthogonality violation and so creates interference among OFDM subcarriers, i.e., inter-carrier interference (ICI). The level of ICI determines the amount of inflicted BER: the higher the ICI, the higher BER at the Rx. To improve the FO estimation accuracy, current OFDM systems employ simultaneously multiple FO estimation mechanisms during the transmission of a frame. These mechanisms use certain publicly-known parts of a frame: frame preamble and pilot subcarriers. The possibility of estimating the FO between a Tx-Rx pair by a third device that is overhearing Tx and Rx's communications, and the publicity of the location of the preamble and pilot subcarreirs in the frame creates a vulnerability of OFDM systems. Such vulnerability has been recently exploited to design jamming attacks against FO estimation at the Rx [21, 22]. However, for these attacks to succeed, the jammer must use high power and target several locations in the frame so as to inflict high BER. Furthermore, even with using high jamming power, these works assume that the jammer is able to accurately target (in time) the arrival times of the transmitted frame preamble and the pilots at the Rx. In practice, however, accurately pinpointing their arrival times is not a trivial task.

In this dissertation, we design a very low-energy stealth reactive jamming attack against the FO estimation. The attack lasts for $< 0.5\%$ of the maximum frame duration when the data rate is at its highest value a frame duration and results in 0.5 BER (irrespective of coding scheme). Moreover, the proposed attack does not need to target several parts of the frame to succeed, it targets one of these parts (which is a part of the preamble) in a way that the other estimation mechanisms cannot compensate for erroneous estimate of the FO after the attack. By eavesdropping and estimating the FO between the Tx (Alice) and the Rx (Bob), and also by using knowledge of the preamble structure, we show that the attacker (Eve) can design a jamming signal that causes a shift in the subcarrier indices at Bob. In designing this signal, Eve accounts for unknown channel parameters between Alice and Bob and those between Eve and Bob, and also possible timing errors in estimating the arrival time of the preamble at Bob. We further demonstrate this attack experimentally using our NI USRPs.

To mitigate the severity of the attack and protect OFDM systems, we propose four preliminary mitigation approaches. The first approach is to consider the designated part for FO estimation in the preamble (specified by the standard) together with all other preamble parts that can be used for such estimation. This way, Bob is able to randomize the part he employs for FO estimation and evade the jamming attack. The second approach is to obfuscate the preamble in a way that makes the timing or FO features hard to extract by Eve. Bypassing the jammed part of the preamble and relying on subsequence parts of the frame for FO estimation is the third approach. Finally, we propose a set of new preamble signals that all satisfy Bob's expectations of the preamble for performing preamble functions. Alice can randomly use one of these signals and reduce the severity of the FO attack, which assumes the transmitted preamble signal is always the same. This technique further motivates employing time-varying preambles, which can be used for carrying information bits in the preamble for other applications.

1.2.2 Friendly Jamming in Multi-link/hop Networks

Friendly jamming (FJ) is a PHY-layer technique that is used to degrade the Alice-Eve channel without harming Bob's reception. FJ aims at preventing unauthorized users (e.g., Eve) from successfully decoding frame or the unencrypted headers. Essentially, a FJ signal is a randomly generated artificial noise. To nullify the FJ signal at Bob, Goel and Negi in [23] proposed a technique that requires multiple antennas for FJ to transmit the artificial noise in the null space of the Alice-Bob channel. Alternatively, a bank of relay nodes can be utilized.

In this dissertation, we consider the problem of the placement of distributed single-antenna FJ devices in a multi-link wireless network, e.g., peer-to-peer (P2P) or multihop, and minimize the required jamming power and the number of FJ devices for secure communications. First, we consider a per-link strategy and formulate an optimization problem that aims at jointly optimizing the power allocation and placement of the FJ devices for a given link. We show that our proposed scheme reduces power consumption by 55%–99% compared to the case in which the optimal placement of the FJ devices is not considered. Next, we consider the joint power allocation and placement of FJ devices for all links jointly (network-wide strategy). The exploitation of the FJ devices to simultaneously cover more than one link saves more energy and reduces the number of FJ devices relative to per-link case. We use distributed MIMO techniques to create a null region around all the legitimate receivers in network-wide scenario and accordingly establish and incorporate sufficient conditions on the jamming powers and locations of the FJ devices. Finally, we propose a novel link weight and a corresponding routing metric for the multihop scenario.

1.2.3 Frame Encryption and Modulation Obfuscation

Alice can prevent various SCI-enabled attacks (e.g., traffic classification) if the PHY/MAC header fields are encrypted at the PHY-layer and the payload's modulation scheme is obfuscated. To do so, we propose two post-modulation schemes that require a single antenna: one for header encryption and one for modulation obfuscation. Before introducing these schemes, we note that encrypting PHY/MAC headers is usually not a viable option for the following reasons:

First, it makes it difficult to authenticate/identify the transmitting device. Encryption is based on a shared secret key. In a network of nodes, different pairs of nodes establish distinct keys for different sessions during the association process at the MAC layer. Session participants are identified by globally unique MAC addresses. Each node maintains a table of session keys that are associated with the MAC addresses of the participants of each session. This means that before decoding the MAC address in an incoming frame, a node does not know the sender and intended receiver of that frame; hence, it cannot immediately look up the corresponding decryption key. Second, the decryption process of an encrypted header incurs additional delay and complexity, especially when block ciphering is employed. Specifically, the Rx needs to set its buffer timer and initiate its demodulator according to PHY-header fields. Delay in decrypting the PHY header may prevent timely operation at the Rx.

We address the problem of transmitter identification through a novel approach in which a PHY-level identifier is embedded in the preamble. This identifier varies with time in a way that only the legitimate Rx (Bob) can authenticate it. The identifier embedding scheme maintains the main properties of the preamble that are essential for its normal functions. We propose embedding schemes for both OFDM and non-OFDM systems.

Our encryption scheme is a stream cipher, which requires a one-time pad, and is facilitated by the identifier embedding scheme. Using the identifier as the seed, Alice and Bob synchronously generate a secret random sequence (i.e., a one-time pad). By employing stream ciphering and generating the secret sequence with low overhead, we avoid high decryption delay and complexity.

The proposed modulation obfuscation technique uses the same secret sequence for mapping the original modulation symbols of a frame into the constellation map of the highest-order modulation scheme supported by the system. This way, Eve cannot identify the true modulation scheme. Minimum-complexity trellis-coded modulation (TCM) codes are used to prevent any performance loss due to this mapping. However, the structure of these codes can still be used by Eve to guess the original modulation scheme. Accordingly, we propose a novel way of exploiting TCM using a secret sequence such that Eve cannot distinguish between two modulation schemes that are mapped to the highest-order modulation scheme.

1.2.4 Exploiting Frame Preamble to Modulate User-Information Bits

Frame preamble constitutes up to 10% of the frame duration. Yet, it is never utilized for carrying user-information bits. We design mechanisms to modulate bits in the preamble of widespread OFDM-based WLAN systems without disrupting its normal operations. In these systems, the Rx does not need to completely know the preamble signal. We exploit this feature to construct several new but compliant preamble waveforms. Each waveform can then represent the modulated version of a bit sequence. The bit sequence can be a sender identifier (to facilitate PHY level encryption), sender's digital signature (for link authentication), message authentication code, a part of the payload (for increasing throughput), etc.

To effectively benefit from the new preamble waveforms, we design a special modulation technique called preamble modulation (*P-modulation*) that combines different time shifts and different phase shifts of the standardized preamble signal to generate compliant preamble waveforms. One special feature of these techniques is that they preserve the expected characteristics of the preamble signal. At the

Rx, we exploit the particular pattern that exists in these compliant preambles as well as the mandatory repetitions in the preamble to efficiently separate different waveforms and demodulate the information bits. Moreover, we design a two-step fine synchronization technique to account for the sensitivity of the proposed demodulation to errors. When the reliability of *P-modulation* is expected to be as good as the one for BPSK modulation scheme, *P-modulation* can embed up to 8 and 19 bits in the preamble of systems that operate over 20 MHz and 80 MHz channels (e.g., 802.11ac), respectively. More bits can be embedded if *P-modulation* is contrasted to higher-order modulation schemes. Our simulation and USRP experiment results demonstrate that the performance of *P-modulation* is as good as the performance of the BPSK modulation scheme. Furthermore, our scheme does not have the aforementioned limitations of channel-based, hardware-based, or MAC layer sender identification/authentication methods.

1.3 Dissertation Organization

The remainder of the dissertation is organized as follows. We first give background related to the preamble structure and its functions in Chapter 2. In Chapter 3, we show how the frequency offset estimation in OFDM systems can be attacked using a short-lived jamming signal. Several mitigation approaches are then discussed. The problem of power allocation and friendly jamming device placement in multi-link networks is studied in Chapter 4. Chapter 5 is dedicated to our modulation obfuscation and encryption schemes, which are facilitated by exchanging a preamble identifier in 802.11b systems. We then present in Chapter 6 our preamble modification scheme (in OFDM systems) to carry information bits. Finally, in Chapter 7 we summarize the contributions of this dissertation and suggest several topics for future research.

CHAPTER 2

Background

Every PHY-layer frame starts with a preamble, which is mainly used for frame detection, FO and channel estimation. Because of the role that the frame preamble plays in the design of the schemes in subsequent chapters, in this chapter we explain the special characteristics based on which OFDM-based and non-OFDM-based preambles in IEEE 802.11 systems are designed. We also explain the common functions of PHY frame preamble in these systems.

2.1 Preamble Structure in OFDM-based 802.11 Systems

OFDM-based 802.11 systems include the systems based on 802.11a/g/n/ac standards. In these systems, the preamble begins with two essential fields (see Figure 2.1 for the preamble of an 802.11a system): a short training field (STF) and a long training field (LTF). The STF contains ten identical short training sequences (STSs), which represent ten replicas of a particular periodic function with period $\lambda_{STS} = 0.8$ μs. The LTF consists of two long training sequences (LASS), which represent two cycles of a known periodic function with period $\lambda_{LTS} = 4\lambda_{STS}$, plus a 1.6 μs long cyclic prefix (GI)[1]. The signal in the STF is generated in every one of four subcarriers, and so has a short period. STSs are used for frame detection and coarse FO correction. LTSs, on the other hand, employ all the data subcarriers and are used for channel estimation and fine-tuning the coarse STS-based FO estimation. The preambles in 802.11n and 802.11ac MIMO standards are in essence similar

[1] In MIMO-OFDM systems, these two fields are followed by additional training sequences for MIMO channel estimation [24].

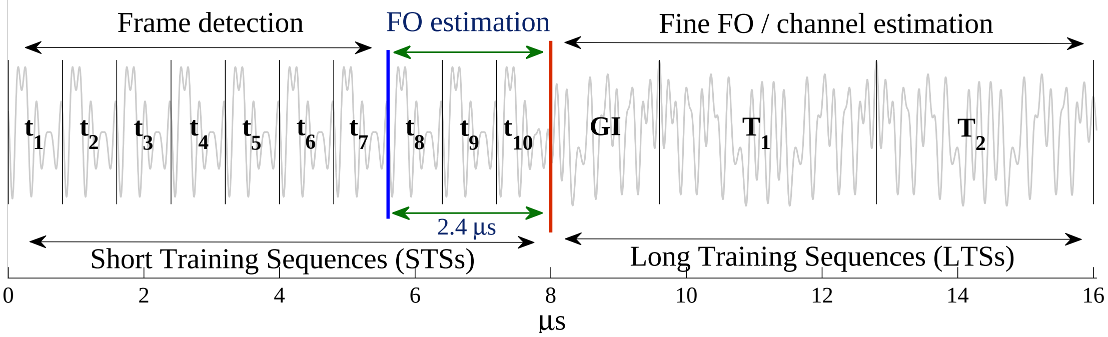

Frame detection
FO estimation
Fine FO / channel estimation
t1
t2
t3
t4
t5
t6
t7
t8
t9
t10
GI
T1
T2
2.4 µs
Short Training Sequences (STSs)
Long Training Sequences (LTSs)
0
2
4
6
8
10
12
14
16
µs

Figure 2.1: Time-domain representation of a common preamble structure in 802.11a/g/n/ac systems (20 MHz bandwidth).

to the preamble of 802.11a but are transmitted over a wider bandwidth (up to 160 MHz). They may also include an additional STF for better automatic gain control (AGC), and multiple LTFs for channel sounding and backward compatibility.

2.1.1 STF Functions

The STF of the preamble is used for frame detection, coarse FO estimation, AGC, diversity selection, and other functions. Accurate frame detection and FO estimation are two key operations that require two identical signals in the STF (e.g., two STSs).

Frame Detection

For a wireless Rx, an increase in the received power is a first indication of an arriving frame. To verify whether this increase is indeed due to an 802.11a/g/n/ac frame and then time-synchronize with it, the Rx checks for the existence of successive identical sequences of a preset length [25]. In Schmidl and Cox's frame detection method, the Rx considers 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 sequence. For example, three STSs with $t_s = 50$ ns sample period (Nyquist rate of 20 MHz) result in $L = 48$ samples. In the 802.11 standards, $1 \leq k \leq 5$. 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:

$$\mathcal{A}(n) = \sum_{i=0}^{L-1} \widetilde{s}^{*}_{n+i} \widetilde{s}_{n+L+i}. \tag{2.1}$$

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

$$\mathcal{M}(n) = \frac{|\mathcal{A}(n)|^2}{\big(\mathcal{E}(n)\big)^2} \tag{2.2}$$

where $\mathcal{E}(n) \overset{\text{def}}{=} \sum_{i=0}^{L-1} |\widetilde{s}_{n+L+i}|^2$ is the received signal energy over the second window. $\mathcal{M}(n)$ is close to zero 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 $\mathcal{M}(n)$ hits the maximum is marked as the beginning of the frame. Because of noise, however, the maximum point may occur later than the actual preamble start time. To account for this, the algorithm first finds $\hat{\mathcal{M}} = \max_n \mathcal{M}(n)$ and then searches for the earliest time before the occurrence of $\hat{\mathcal{M}}$ with an $\mathcal{M}$ value greater than $(1-\epsilon)\hat{\mathcal{M}}$, where $0 < \epsilon < 1$ is a system parameter. That time instant is taken as the beginning of the frame.

FO Estimation

Let Δf be the actual frequency offset between a transmitter (Tx) and an Rx. This FO translates into a time-varying 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. The *de facto* time-domain FO estimation method used in OFDM systems is the one proposed by Schmidl and Cox [25]. We consider it here as a representative FO estimation scheme. This method assumes that the channel does not change during the preamble transmission. Having a sequence $\mathbf{r}$ with two identical halves is the key idea in this method. It works as follows. 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 $\mathbf{r}$, $i = 1, \ldots, 2L$. So $r_i = r_{L+i}$. Ignoring the noise, this equality also holds for the corresponding samples at the Rx as long as there is no FO. However, with an FO of Δf, the phase r_{L+i} relative to r_i is rotated by $\Delta\varphi(t_s) = 2\pi\Delta f L t_s$. Multiplying the conjugate of r_i (i.e., r_i^*) by r_{L+i}, we obtain:

$$s_i \overset{\text{def}}{=} r_i^* r_{L+i} = |r_i|^2 e^{-j2\pi\Delta f L t_s} = |r_i|^2 e^{-j\Delta\varphi(t_s)}. \tag{2.3}$$

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 Rx, denoted by $\widetilde{s_i}$, is:

$$\widetilde{s_i} = |h_i r_i|^2 e^{-j2\pi\Delta f L t_s} + \bar{n_i} \tag{2.4}$$

where $\bar{n_i} \overset{\text{def}}{=} r_i n_{L+i}^* + r_{L+i}^* n_i + n_i n_{L+i}^*$ 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 $\tilde{s_i}$'s, i.e.,

$$\widetilde{\Delta\varphi(t_s)} = \measuredangle\Big(\sum_{i=0}^{L-1} \widetilde{s_i}\Big) \tag{2.5}$$

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

$$\widetilde{\Delta f} = \frac{\widetilde{\Delta\varphi(t_s)}}{2\pi L t_s}. \tag{2.6}$$

Assuming that the channel does not change during the STF, the above method is independent of the channel. It is widely adopted in practical systems. Figure 2.2 shows an example of a sequence of length $2L = 8$ samples. The more the samples used to estimate $\widetilde{\Delta\varphi}$, the more accurate is the estimated FO.

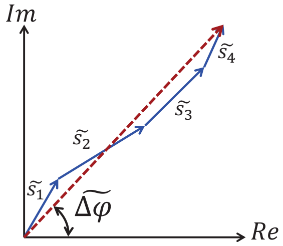


Figure 2.2: Example of phase offset averaged over $L = 4$ $\tilde{s}_i$ terms.

2.1.2 LTF Functions

LASS are used for channel estimation, i.e., estimating the response of the channel, because they are supposed to be almost FO-free after STS-based FO correction. There are two general approaches for channel estimation: Frequency domain and time domain [26]. In both approaches, the a priori 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). The MSE can grow quadratically as a function of the FO estimation error [27].

LTF is also used for fine-tuning the STF-based FO estimation. In LTF-based FO estimation, the same method of Schmidl and Cox is used, but with $L = \lambda_{LTS} = 4\lambda_{LTS}$. Therefore, the FO estimation will be more accurate.

2.2 Preamble Structure in 802.11b Systems

In contrast to OFDM systems, 802.11b systems have a single carrier. Instead of the known periodic STSs and LASS, they exploit a scrambled version of a 128-bit all-one preamble that is spread using an 11-chip Barker sequence (see Table 2.1). For a Barker sequence of length N, its autocorrelation function at lag k, denoted by $\mathcal{L}(k)$, is very low at non-zero lags (orthogonality property). This can be exploited for frame detection and timing. Formally,

$$\mathcal{L}(k) = \Big| \sum_{j=1}^{N-k} b_j b_{j+k} \Big| \leq 1, 1 \leq k < N \tag{2.7}$$

where $\mathbf{b} = \{b_1 b_2 \dots b_N\}$ is a Barker sequence. The receiver cross-correlates this known sequence with the samples of the received signal $\mathbf{r} = \{r_1 r_2 \dots\}$ and computes the square of the cross-correlation value, denoted by $\mathcal{R}(\mathbf{b}, n)$:

$$\mathcal{R}(\mathbf{b}, n) = \Big| \sum_{j=1}^{N} b_j^* r_{j+n-1} \Big|^2 . \tag{2.8}$$

$\mathcal{R}(\mathbf{b}, n)$ is expected to peak when the nth sample of $\mathbf{r}$ marks the beginning of one of the transmitted Barker sequences. To improve the detection accuracy, $\mathbf{b}$ is replaced with a series of identical Barker sequences, one sequence per preamble bit.

Input	Sequence
0	$+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1$
1	$-1, +1, -1, -1, +1, -1, -1, -1, +1, +1, +1$

Table 2.1: DSSS signal spreading based on an 11-chip Barker sequence for DBPSK modulation (IEEE 802.11b standard).

The 802.11b preamble consists of several repetitions of a publicly known pattern. Similar to OFDM systems, FO estimation in 802.11b involves detecting the arrival of at least two identical parts of the preamble[2] and then applying the same Schmidl and Cox method. After compensating for δ_f, the same preamble sequence is used for channel estimation by comparing the clean version of the preamble with its received value.

[2] Scrambling transforms an all-one preamble bit sequence into a sequence of zero's and one's. Methods like [28] are used to detect the zero's and change them into one's.

CHAPTER 3

Jamming of Frequency Offset Estimation in OFDM Systems

3.1 Introduction

Communication between two wireless devices involves several concerted functions at the PHY layer, including time synchronization, FO correction, channel estimation, channel coding, modulation, interleaving, and others [1]. PHY-layer functions are designed to combat oscillator imperfections and wireless channel impairments, and to decode wireless signals that are corrupted by a limited amount of interference. However, due to the leakage of SCI, wireless transmissions still remain vulnerable to intentional interference attacks, commonly referred to as jamming.

One measure of the effectiveness of a jamming attack is its duty cycle, i.e., the fraction of the frame that needs to be jammed so that the frame is discarded at the Rx [29, 30]. This metric is directly related to the jammer's distance to the Rx, energy budget, and the ability to disrupt concurrent transmissions. A jammer that remains active for a longer period can corrupt more bits and defeat stronger error correction codes (ECCs), at the expense of higher energy consumption and fewer targeted communications. This more potent jammer is also easier to detect [31], localize, and physically remove using jammer localization methods [30].

In this chapter, we investigate an extremely low duty cycle jamming model that is facilitated by public knowledge of the frame structure and PHY-layer functions. Our goal is to demonstrate how an adversary can inflict the highest possible number of decoding errors at the Rx, without jamming the corresponding header or payload

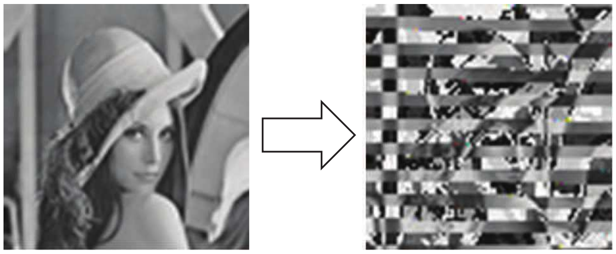


Figure 3.1: Effect of uncompensated FO on a bitmap image over a noiseless channel (FO = 0.32% of the subcarrier spacing).

symbols. PHY-layer standards usually employ publicly known sequences for their preambles at the beginning of a frame to acquire important communication parameters, such as the transmission timing, channel, and FO [1]. These parameters are used to align received symbols. An adversary may exploit the publicity of the preamble to construct a reactive jamming attack and target the estimation of these critical parameters. In particular, we demonstrate the feasibility of an energy-efficient and low duty cycle attack against the FO estimation process of IEEE 802.11 OFDM-based devices (including 802.11a, .11g, .11n, .11ac, and 11ah), all of which exploit the same preamble structure. Our results can be extended to other OFDM-based systems, including 802.16e/m (WiMAX), LTE, and 5G.

The jamming of OFDM systems has recently been the subject of extensive research (e.g., [20–22, 32–35]). These works often consider vulnerabilities in time synchronization or susceptibility to ICI. For example, the authors in [20] proposed several jamming attacks against OFDM time synchronization, including barrage attacks, false preamble timing, and preamble warping. In the barrage attack, white noise is transmitted to decrease the SNR during synchronization. In false preamble timing, the jammer forges a preamble to fool the Rx about the true start time of the frame. A similar technique was used in [33] against an 802.11b Rx to hamper the network throughput. Preamble warping tries to destroy the time-domain correlation (used for time acquisition) within the preamble.

3.1.1 Existing Attacks Against FO Estimation

In OFDM systems, frequency synchronization errors are more devastating than timing errors [26]. When two radios are tuned to the same target frequency, their oscillators cannot be exactly aligned to that frequency due to hardware imperfections. FO is the inherent difference between the actual frequencies of these two oscillators. In OFDM, FO is usually normalized to the inter-subcarrier frequency interval, called *subcarrier spacing*. Without frequency synchronization, the performance of OFDM degrades severely because all subcarriers will move away from their expected frequencies, resulting in subcarriers' orthogonality violation, ICI [26], and channel estimation errors [27, 36].

To appreciate the significance of correct FO estimation, we conduct a simulation experiment in which a frame containing a bitmap image is transmitted between two nodes. Figure 3.1 depicts the effect of a small FO estimation error (0.32% of subcarrier spacing) on the transmitted image (left) when 48 subcarriers are used at a rate of 6 Mbps. The received image (right) exhibits noticeable degradation in the form of image block misplacement. In practice, FO can be even larger than the subcarrier spacing [1].

A few jamming schemes have been proposed in the literature (e.g., [21, 22, 33]) with the goal of inflicting ICI. Phase warping and differential scrambling attacks [21] consider the preamble structure of Schmidl and Cox [25], which is different from the one used in 802.11 OFDM-based standards, and in essence try to alter preamble symbols in a heuristic fashion without providing any success guarantees. Gummadi *et al.* [33] showed the vulnerability of 802.11a clock (frequency) synchronization to a certain narrow-band jamming pattern that interferes with the entire preamble. In [22] the jammer transmits multiple asynchronous subcarriers to cause ICI in an OFDM symbol. These attacks may fail if robust ECC, interleaving methods, or additional FO estimation mechanisms are employed at the Rx.

3.1.2 Main Contributions and Chapter Organization

We design an energy-efficient jamming attack that interferes with a small portion of the preamble, i.e., one of the parts used for FO estimation, and causes one or two units *shift* of the subcarrier indices (e.g., every subcarrier takes the position of its next/previous subcarrier). To make this design possible, the adversary (Eve) must first estimate the FO between the legitimate transmitter (Alice) and intended receiver (Bob), and then quickly detect the transmission of a target frame. We provide an adaptive frame detection method to facilitate fast detection at Eve. The superposition of the jamming signal with the preamble are designed to delude Bob into estimating an FO that is sufficiently far from the true FO, so that Bob decodes wrong symbols, i.e., the symbols of adjacent subcarriers. The idea is to come up with a structure that is similar to the actual preamble so as to control the FO embedded in the jamming sequence. The superposition of these two signals with different FOs at the Rx achieves sufficient FO estimation error. We derive the amount of FO estimation error needed to guarantee erroneous OFDM demodulation and accordingly, develop an optimal attack strategy. To ensure that the jamming signal is independent of the Alice-Bob channel parameters (which are unknown to Eve), we propose a *pairing* scheme for the jamming sequence. The jamming attack should also account for timing errors in frame detection at Eve while keeping the jamming signal channel-independent. For this purpose, a *chaining* scheme is designed on top of the pairing scheme to account for other possible frame start times.

Consequently, not only the channel estimation is automatically corrupted at Bob, but more importantly, all the frequency subcarriers are shifted forward or backward. Hence, Bob will have a shifted version of the bitstream transmitted in every OFDM symbol. Combined with a faulty channel estimation and thus demodulation errors, the bits become irrecoverable. We further optimize the power of this jamming attack and experimentally evaluate its performance on a USRP testbed. In contrast to previous attacks on the frame preamble, ours in essence does not aim at necessarily causing ICI. It is also different from the attacks in [21,22,33] in that it is *channel-independent* and *energy-efficient*, i.e., only a small portion of the preamble is jammed irrespective of the jammer's location. This short-lived attack lasts for less than 3 μs per frame (equivalent to, for example, about 0.5% of 802.11a's maximum frame duration when the data rate is at its highest value). Note that this is even shorter than the duration of an OFDM symbol (4 μs). Our proposed attack also disarms all the provisioned FO estimation methods by just efficiently defeating one of them. Our work focuses on the 802.11 OFDM-based wireless systems, and efficiently exploits their FO vulnerability for the first time.

The chapter is organized as follows. In Section 3.2, we provide background on the consequences of FO in OFDM-based 802.11 OFDM systems. The system model, assumptions, and evaluation metrics are given in Section 3.3. The proposed attack and the optimal jamming strategy are presented in Section 3.4 and related issues are discussed in Section 3.5. Section 3.6 demonstrates the effectiveness of the attack through simulations and experiments. Finally, we propose possible remedies in Section 3.7.

3.2 FO in OFDM Systems

In OFDM, a bitstream is split into several substreams, each of which is digitally modulated and transmitted over one of the orthogonal frequency channels (subcarriers). For example, 802.11a/g defines 64 subcarriers with subcarrier spacing $f_\Delta = 312.5$ kHz within a bandwidth of 20 MHz. Only 48 of these subcarriers are used for data. Four other subcarriers carry pilot signals and the remaining 12 subcarriers are not used. So an 802.11a/g OFDM symbol is transmitted over 52 subcarriers.

ICI in OFDM systems creates significant BER at the Rx [37] (see Figure 3.2). To prevent ICI, the Rx uses the PHY-layer preamble to estimate the FO (same for

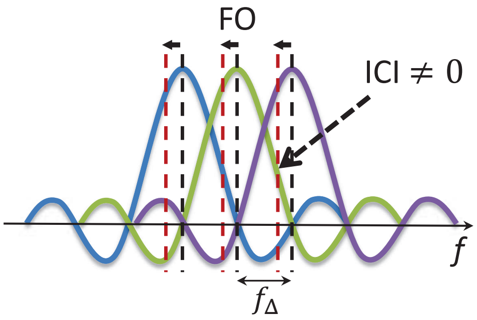


Figure 3.2: Inter-carrier interference (ICI) as a result of uncorrected FO in a system with three subcarriers.

all subcarriers) and adjust the subcarriers to their expected orthogonal frequency bins. If the offset is less than half of the frequency distance between the subcarriers, the Rx can safely identify the frequency bin that each subcarrier belongs to.

As explained in Chapter 2, the preamble in OFDM-based 802.11 systems contains two essential fields: STSs and LTSs (see Figure 6.1). The periodic function in an STS is constructed by superposing only the subcarriers whose frequencies are integer multiples of $4f_\Delta$. As a result, the minimum subcarrier spacing between any two STS-enabled subcarriers is $4f_\Delta$, and hence their period is $\lambda_{STS} = \lambda_{LTS}/4$. STSs are used for frame detection and coarse FO correction. LTSs, on the other hand, employ all the data subcarriers and are used for channel estimation and fine-tuning the coarse STS-based FO estimation. In addition to causing ICI, a linear increase in the phase offset during the LTSs due to FO (i.e., $\Delta\varphi(t) = 2\pi\Delta f t$) results in incorrect channel phaser estimation. To compensate for channel impairments, the inverse of the phaser is multiplied to the received samples. As a result, all received modulated samples will be rotated equally on the constellation map, leading to more bit errors. Beyond channel estimation errors, accumulation of the phase offset can significantly change the phase of some of the symbols, especially in long frames.

Regarding the phase of a complex number such as $\widetilde{s_i}$ during FO estimation, the Rx observes a value between $-\pi$ and π. In other words, the Rx cannot distinguish $\Delta\varphi$ from $\Delta\varphi \pm 2k\pi$ in (2.6), for any integer k. The phase offset of 2π corresponds to $\frac{1}{Lt_s}$ offset, i.e., one subcarrier spacing. In particular, consider a subcarrier and two FOs from it, 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 Rx interprets $\Delta f_1 + \frac{1}{Lt_s}$ as Δf_1 and will mistakenly adjust Δf_2 to the neighboring subcarrier bin. *In general, the phase is unambiguous and correctable as long as* $|\Delta f| < \frac{1}{2Lt_s}$ *(half a subcarrier spacing).* This also implies that a longer period of a cycle reduces the range of FO that can be corrected unambiguously. Given a fixed sampling interval, a longer period results in higher L.

Let th_s and th_l be the maximum $|\Delta f|$ values that STSs and LTSs can correct unambiguously, respectively. In the 802.11a/g, two of the last three STSs are chosen to form a sequence with two identical halves for coarse FO estimation. Since the number of samples of an LTS is four times the number of samples of an STS, then $th_l = th_s/4 = f_\Delta/2$.

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, the Rx then computes $\widetilde{\Delta f_l}$ to fine-tune the coarsely estimated FO. This explains one of the reasons for concatenating short and a long training fields in 802.11 systems. Consequently, if the actual FO is larger than th_s, this FO estimation method fails to fully compensate for it.

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 gradually rotates the phase of the received symbols on the constellation map and may increase the BER, specially in the long frames. A predetermined subset of subcarriers with known values (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 [26]. In addition, known pilot subcarriers can be used for tracking channel variations.

3.3 Model and Assumptions

We consider a link between Alice (the Tx) and Bob (the Rx). The adversary (Eve) is in the transmission ranges of both Alice and Bob. Alice transmits an 802.11 OFDM frame and Bob uses a few of the first STSs for frame detection. He chooses two of the last three STSs, in conformity with the standard (see Figure 6.1) and employs the Schmidl and Cox method for FO estimation. Once Bob estimates the coarse FO using STSs and compensates for $\widetilde{\Delta f_s}$, he assumes, by default, that the residual FO is less than th_l and then estimates it using LTSs. According to the 802.11 standard, Bob does not perform any kind of boundary check during the LTS- and pilot-based FO estimation processes.

Eve aims at irrecoverably corrupting Alice's frame at Bob using the lowest possible jamming effort. Eve is aware of the PHY-layer protocol and the FO correction mechanism at Bob. She makes no assumptions about the channel parameters or Alice's transmission power. If the oscillators are either stable or accurate, Eve initially eavesdrops on Alice's and Bob's preamble transmissions (e.g., data-ACK exchanges) for a while. Through averaging, she estimates their FOs relative to Eve, denoted by Δf_{ae} and Δf_{be}, respectively[1].

The metrics of interest are coarse and final estimated FOs at Bob, symbol error rate (SER), the BER after demodulation but before decoding, and the jamming effort (defined as the jammer's duty cycle [29]). These metrics will be studied with respect to the SNR, modulation scheme, and signal-to-jamming ratio (SJR) at Bob.

[1] In general, oscillators exhibit numerous instabilities, due to aging, temperature, acceleration, ionizing radiation, power supply voltage, etc. Thus, the Rx must update the FO estimate on a per-frame basis, even if the frame sender is already known. This is specially the case with non-stable oscillators. In this case, Eve can perform FO estimation along with fast frame detection to optimally design the jamming signal for each frame (see Section 3.4.1).

3.4 Proposed Frequency Offset Estimation Attack

In this section, we describe in detail an attack on the FO estimation. Eve launches this attack in two phases: (1) Eavesdropping on the channel to detect the start of Alice's frame transmission and acquire its timing information; and (2) jamming the last three STSs of the preamble, which are used for coarse FO estimation.

3.4.1 Phase 1: Adaptive Fast Frame Detection

To pinpoint the last three STSs in time and corrupt the FO estimation at Bob, Eve must detect Alice's frame and synchronize with its arrival at Bob. The detection should be fast enough to allow sufficient time for processing, switching to transmission mode, and jamming the last three STSs. Referring to the frame detection mechanism in Section 3.2, Eve chooses the minimum possible window size (one STS, $L = 16$) and reduces the capture time to $2.5\lambda_{STS} = 2\ \mu s$ to make sure that at least the first two STSs are captured.

To account for the higher detection inaccuracy due to the small window size, Eve assumes that the actual start time belongs to the first $V = \log_2(L)^2$ sample indices $i_0, i_1, \ldots, i_{V-1}$ that are greater than $(1-\epsilon)\hat{\mathcal{M}}$ and finds all of them, instead of just looking for the first one. She sets ϵ to a value less than $1/L$, the contribution of a preamble sample pair in $\mathcal{M}(n)$. This is an attempt to exclude the samples located more than one index before the actual frame start time. If there are less than V sample values greater than the threshold, Eve adaptively decreases the threshold by finding the smallest ϵ that guarantees the existence of V candidates[3].

3.4.2 Phase 2: Preamble Jamming

Based on i_0, Eve computes the arrival time of the last three STSs of the preamble and generates a jamming signal that would be aligned with those STSs. The energy-

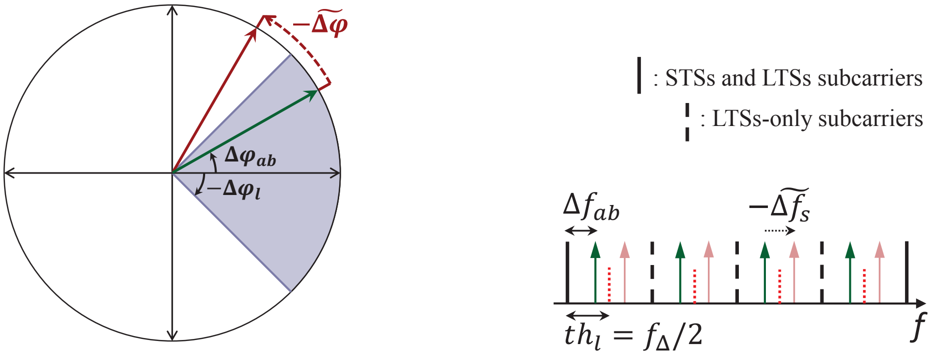
[2] The reason of this specific number will be explained in Section 3.4.2.

[3] Eve may also apply the synchronization method in [38] to improve the detection accuracy.

efficient jamming sequence is designed to defeat all STS-, LTS-, and pilot-based FO corrections without jamming the LTSs and pilot subcarriers. For this attack to be successful, Eve has to account for unknown channel parameters and frame-detection timing errors. More specifically, the jamming sequence is designed to achieve the following goals:

Forcing Bob to make a destructive error

By default, Bob assumes that the FO to be estimated using LTSs is less than th_l. If Eve deceives Bob into erroneously push the FO beyond th_l after receiving the STSs instead of reducing it, then she achieves her goal without needing to jam the LTSs.



(a) Phase domain: The shaded area represents the LTS-based correctable range. A wrong phase estimation $\widetilde{\Delta\varphi}$ can move $\Delta\varphi_{ab}$ out of the correctable range.

(b) Frequency domain: Incorrect estimation of $\widetilde{\Delta f_s}$ can move Δf_{ab} out of the LTS-based correctable range.

Figure 3.3: Phase and frequency offsets as observed during the STSs.

Without loss of generality, Eve assumes i_0 is the correct start time of the frame (we will relax this assumption later). Let $\Delta f_{eb} = -\Delta f_{be}$ and $\Delta f_{ab} = \Delta f_{ae} - \Delta f_{be}$ represent Bob's estimates of Eve-to-Bob and Alice-to-Bob FOs, respectively. Let $\Delta\varphi_{ab}$, $\Delta\varphi_{eb}$, and $\Delta\varphi_l = \pi/4$ be the phase offsets corresponding to Δf_{ab}, Δf_{eb}, and th_l, respectively, after a single STS (0.8 μs). To cause incorrect 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:

$$|\Delta\varphi_{ab} - \widetilde{\Delta\varphi}| > \Delta\varphi_l. \tag{3.1}$$

Figure 3.3(a) and 3.3(b) show an example of such a situation in the polar coordinates and frequency domain, respectively.

Eve's jamming signal needs to satisfy (3.1). Let g be the Eve-to-Bob channel coefficient. We assume that during Eve's jamming period, g is the same for all the jamming samples that belong to the jamming sequence $\mathbf{u}$, denoted by $u_i, i = 1, \ldots, 2L$. Let $\tilde{r}_i = hr_i$ and $\tilde{u}_i = gu_i$. We consider two different approaches for generating the jamming sequence:

1) Random noise: A simple way 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 after the superposition and ignoring the noise term in (2.4), we have:

$$\begin{aligned}\mathcal{A}_{\text{random}} &\overset{\text{def}}{=} \sum_{i=0}^{L-1} \widetilde{s}_i = \sum_{i=0}^{L-1} (\tilde{r}_i + \tilde{u}_i)^* (\tilde{r}_i e^{-j\Delta\varphi_{ab}} + \tilde{u}_{L+i}) \\ &= \sum_{i=0}^{L-1} |\tilde{r}_i|^2 e^{-j\Delta\varphi_{ab}} + \sum_{i=0}^{L-1} \tilde{r}_i{}^* \tilde{u}_{L+i} \\ &\quad + \sum_{i=0}^{L-1} \tilde{u}_i^* (\tilde{r}_i e^{-j\Delta\varphi_{ab}} + \tilde{u}_{L+i}).\end{aligned} \tag{3.2}$$

The phase and amplitude of the 2nd and 3rd terms in (3.2) (and hence $\widetilde{\Delta\varphi}_{\text{random}} \overset{\text{def}}{=} \measuredangle\mathcal{A}_{\text{random}}$) 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. Hence, $\widetilde{\Delta\varphi}_{\text{random}}$ may not satisfy (3.1), so FO jamming with a random signal cannot provide any FO distortion guarantees to beat LTS-based FO estimation.

2) Fake preamble: A more effective jamming approach that exploits both knowledge of the FO estimation algorithm and Δf_{ab} is to construct a fake preamble with "identical halves". For now, assume that the samples of the jamming signal $u_i, i = 1, \ldots, 2L$ can take any arbitrary value as long as the signal conforms to the protocol bandwidth requirement. The preamble phase warping attack in [21] is a special case of this approach, where the jamming signal is a random frequency-shifted version of an arbitrary fake preamble. The advantage of having identical halves is that we can control and carefully calculate a desired FO for $\mathbf{u}$ based on how Bob estimates Δf_{ab}. Here, we also note that the channel response between Eve and Bob does not change the FO. Before we explain how a desired FO (and hence Δf_{eb}) is determined, consider the superposition of Alice's signal and Eve's jamming at Bob. Dropping the index i from (2.4) and ignoring the noise term, we have:

$$\begin{aligned}\tilde{s} = (\tilde{r} + \tilde{u})^*(\tilde{r}e^{-j\Delta\varphi_{ab}} + \tilde{u}e^{-j\Delta\varphi_{eb}}) = e^{-j\Delta\varphi_{ab}} \times \\ \Big[\underbrace{|\tilde{r}|^2 + |\tilde{u}|^2 e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} + \tilde{r}^*\tilde{u}e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})} + \tilde{u}^*\tilde{r}}_{\mathcal{B}}\Big].\end{aligned} \tag{3.3}$$

Thus, the estimated phase offset at Bob is:

$$\widetilde{\Delta\varphi} = \measuredangle\tilde{s} = \Delta\varphi_{ab} + \measuredangle\mathcal{B} + \measuredangle\bar{n}. \tag{3.4}$$

Note that the phase estimation error $\varphi_e \overset{\text{def}}{=} \measuredangle\mathcal{B}$ is a function of SJR and $\Delta\varphi_{eb}$, and jamming will have no effect if $\varphi_e = 0$.

Upon calculating $\widetilde{\Delta\varphi}$ and $\widetilde{\Delta f_s}$, Bob changes the FO for the rest of the frame to $\Delta f_{ab} - \widetilde{\Delta f_s}$. According to (3.1), Eve is successful if she can ensure that $\Delta\varphi_{eb}$ satisfies the following:

$$|\Delta\varphi_{ab} - \widetilde{\Delta\varphi}| > \Delta\varphi_l \Rightarrow |\varphi_e + \measuredangle\bar{n}| > \Delta\varphi_l = \frac{\pi}{4}. \tag{3.5}$$

Eve can guarantee a desired φ_e only if SJR→ $-\infty$. Otherwise, even if she 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. That means that Eve is still unable to guarantee a successful attack, which is also the case in the preamble phase warping attack.

Designing a channel-independent jamming signal

To address the aforementioned challenge, Eve takes advantage of Alice's known preamble samples and the product sum in (2.5) to cancel out the terms with unknown phases. Eve first chooses $L/2$ non-overlapping pairs of samples. Without loss of generality, let Eve pair the samples in order and let (u_1, u_2) be the first pair of samples in the jamming sequence. 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 (3.3) are eliminated. Thus,

$$u_2 = -\frac{r_1^*}{r_2^*} u_1 \tag{3.6}$$

which implies that

$$\begin{aligned} &\widetilde{s_1} + \widetilde{s_2} = e^{-j\Delta\varphi_{ab}} \times \\ &\Big[|\tilde{r}_1|^2 + |\tilde{r}_2|^2 + \left(|\tilde{u}_1|^2 + |\tilde{u}_2|^2|\right) e^{-j(\Delta\varphi_{eb} - \Delta\varphi_{ab})}\Big]. \end{aligned} \tag{3.7}$$

The requirement in (3.6) is similarly 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, denoted by $\mathcal{A}_{\text{fake}}$, becomes:

$$\begin{aligned} \mathcal{A}_{\text{fake}} = \sum_{i=0}^{L-1} \widetilde{s}_i = \\ e^{-j\Delta\varphi_{ab}} \underbrace{\left[\sum_{i=0}^{L-1} |\tilde{r}_i|^2 + \sum_{i=0}^{L-1} |\tilde{u}_i|^2 \, e^{-j(\Delta\varphi_{eb}-\Delta\varphi_{ab})} \right]}_{\mathcal{C}(\Delta\varphi_{eb}-\Delta\varphi_{ab})}. \end{aligned} \tag{3.8}$$

Now $\mathcal{A}_{\text{fake}}$ is a function of the difference between $\Delta\varphi_{ab}$ and $\Delta\varphi_{eb}$ only. So Eve can determine a desired value of $\Delta\varphi_{eb}$ in a way that makes $|\measuredangle\mathcal{C}(\Delta\varphi_{eb} - \Delta\varphi_{ab})| > \Delta\varphi_l$, which satisfies (3.5).

Robustness to errors in frame start time

We now relax the assumption that Eve can precisely determine the true frame start time and consider a scenario in which she compiles a short list of possible frame start times besides i_0, as explained in Section 3.4.1. Thus far, we have required the jamming sequence to have identical halves with a $\Delta\varphi_{eb}$ that satisfies (3.5) and the even samples to be a function of odd samples (pairing rule). Eve could still benefit from the remaining free, unassigned samples (i.e., odd samples) to cancel out channel-dependent terms for other possible start times. We generalize the pairing technique to larger sets of samples and define the following *chaining rule* to account for $V-1$ other start times $i_1, i_2, \ldots, i_{V-1}$.[4]

Let $\mathbf{m} = \{m_0, \ldots, m_{V-1}\}$ where $m_j = i_j - i_0$. First, Eve extends her jamming sequence by appending (cyclically postfixing) the first m_{V-1} jamming samples to this sequence. So for any candidate frame start time i_j, the jamming signal will be fully superposed on Alice's three STSs because the jamming signal is cyclically extended already by $m_{V-1} > m_j$ samples. Next, Eve assumes that i_1 is the correct

[4] Eve can precompute and then account for the propagation delays by timestamping the data-ACK exchanges between Alice and Bob and estimating the Eve-to-Bob distance. The chaining rule can also be leveraged to account for errors in estimating these delays.

frame start. In this case, the superposition of the jamming signal on Alice's three STSs will be different from the previous case (i.e., the jamming sequence is slid with respect to Alice's STSs) and (3.6) is no longer sufficient to eliminate the last two channel-dependent terms within $\mathcal{B}$ in (3.3). Instead, Eve can find pairs of yet free samples and, similar to the pairing rule, define one of the samples of each of such pairs based on the other sample of that pair and also the corresponding samples in $\mathbf{r}$. After this step, half of the free samples will be given values. Eve repeats the same procedure for the rest of the frame start times and free samples. Based on these hierarchical dependencies among the samples u_i, Eve constructs a binary *chaining tree* in which the dependency between two samples is mapped to a parent-child relationship. Note that an unassigned (free) sample may already have a chain of other dependent sample(s). The value of the dependents will be updated whenever that sa

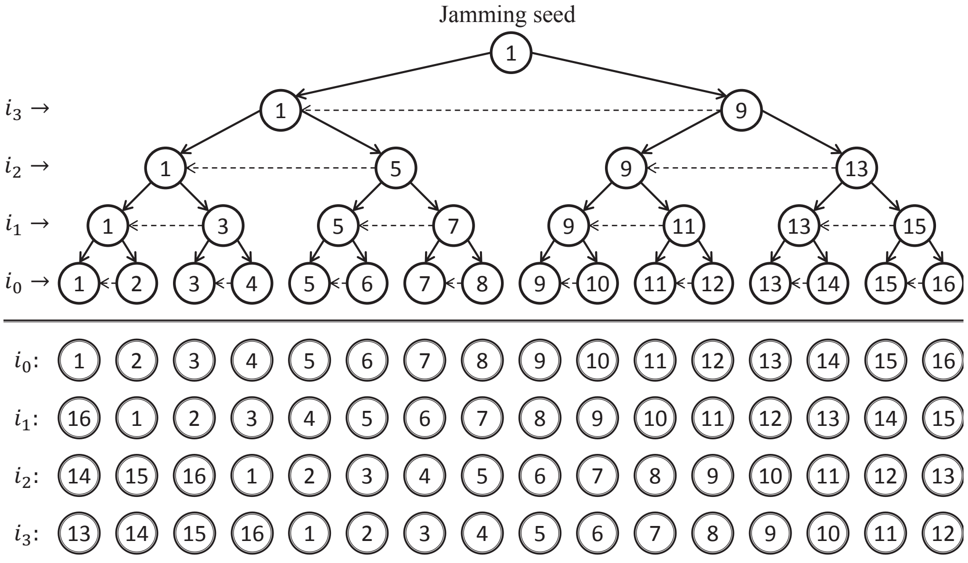

Jamming seed
1
i_3 →
1 9
i_2 →
1 5 9 13
i_1 →
1 3 5 7 9 11 13 15
i_0 →
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i_0: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i_1: 16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
i_2: 14 15 16 1 2 3 4 5 6 7 8 9 10 11 12 13
i_3: 13 14 15 16 1 2 3 4 5 6 7 8 9 10 11 12

Figure 3.4: Cascaded chaining and pairing of the samples towards the jamming seed. Jamming samples are shown on the tree and the shifted versions of Alice's preamble on the bottom. Horizontal dashed lines represent direct dependency between samples.

An example is depicted in Figure 3.4 with $\mathbf{m} = \{0, 1, 3, 4\}$. Without loss of generality, we assume Alice's preamble sequence is shifted instead of the jamming sequence. The tree in this figure shows how the jamming samples are being chained together and used to construct the tree from the bottom to the top. A pair of free samples are considered as siblings. The left child specifies the value of its right sibling based on m_j and then the left child is copied to its parent node. So the right child depends on its sibling. To explicitly define the dependency between the two sibling samples, all their dependent samples must also be taken into account because their values in (3.3) are affected by their parents' values. For example, when $j = 1$, Eve may select two free samples u_1 and u_3 (together with their dependents u_2 and u_4) to eliminate the channel-dependent terms:

$$u_1^* r_{16} + u_2^* r_1 + u_3^* r_2 + u_4^* r_3 = 0 \tag{3.9}$$

which implies the dependency of u_3 to u_1 (u_2 and u_4 are substituted by their corresponding pairing rule dependencies on u_1 and u_3):

$$u_3^* = -\frac{r_4(r_2 r_{16} - r_1 r_1)}{r_2(r_2 r_4 - r_3 r_3)} u_1^*. \tag{3.10}$$

Now the value of the dependent of u_3 (u_4 in this example) is updated to maintain its dependency relationship with the right sibling u_3.

A pseudocode of the chaining rule, which also contains the pairing rule, is provided in Algorithm 1. The algorithm iterates for each m_j, $j = 1, \ldots, V - 1$. At each iteration and for each pair of free samples, the right subtree (the right siblings of all its $2^j - 1$ dependents) is multiplied by a coefficient x (defined in line 8) such that the summation of the corresponding 2^j product terms in (3.3) and the 2^j terms corresponding to the left subtree is zero. The horizontal arrows in Figure 3.4 show the dependence of the right subtrees on their left subtrees. As a result, $L/2^j$ samples are assigned at each iteration and the algorithm terminates after $V = \log_2(L)$

Algorithm 1 Chaining and pairing rules combined

```
**Input**: $L, V, \mathbf{r}[1 \dots L], \mathbf{m}[0 \dots V-1]$
**Initialize:** $\mathbf{u} = \mathbf{0}$
**for** $j \leftarrow 0, V-1$ **do**
    $k \leftarrow 2^j$
    **while** $k < L$ **do**
        $\mathbf{t}$ = circularly shifted $\mathbf{r}$ by $m_j$
        $x = -\sum_{i=k-2^j+1}^{k} u_i t_i^* / \sum_{i=k+1}^{k+2^j} u_i t_i^*$
        $[u_k, \dots, u_{k+2^j-1}] = [u_k, \dots, u_{k+2^j-1}] * x$
        $k = k + 2^{j+1}$
    **end while**
**end for**
**Return** $\mathbf{u}$
```

iterations. In the end, all but one of the samples (u_1 in our example) will be a right sibling at least once at some point in the tree and so are assigned. We call the remaining free sample the *jamming seed*, to which all the samples are chained either directly or recursively. The jamming seed can be used to control the jamming power.

3.4.3 Effects of LTSs on FO and Channel Estimation

LTSs are used for fine FO and channel estimation. As explained in Section 3.2, the phase offset from the LTS-based FO corresction perspective is between $-\pi$ and π, which means that the true FO after STS-based correction has to be between $-th_l$ and th_l. So LTSs can correct up to $th_l = f_\Delta/2$ FO, and any remaining phase offset will be an integer multiple of 2π, which corresponds to $2k\,th_l = kf_\Delta$, $k = 1, 2, \dots$. In other words, the LTSs at Bob round up the FO manipulated by $\widetilde{\Delta f_s}$ to the nearest multiple of $2th_l$ and avoid ICI by adjusting the subcarriers to the closest, though incorrect, frequency bins. Consequently, in this attack all the subcarriers will be shifted forward or backward, replacing neighboring subcarriers. Bob eventually demodulates the bits of all OFDM symbols, but he is unaware that these symbols have been shifted and misplaced. A simple example with four subcarriers is provided in Figure 3.5. Each subcarrier carries two bits (QPSK-modulated symbols). In the shifted version, two unknown bits are added in the beginning and the rest of the sequence is shifted to the right, although the bits are correctly demodulated. Therefore, when the bits of different OFDM symbols are concatenated to reconstruct the original bit sequence, the entire sequence will look shuffled and out-of-order compared to the original bit sequence. A shifted version of an arbitrary bit sequence will result in very high BER.

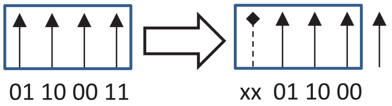


Figure 3.5: Example of the FO attack on four subcarriers (left): The attack shifts the subcarriers and the corresponding bits to the right.

An STS-based FO estimation error also affects the channel estimation process, which is applied across the LTSs, specially if Bob estimates the channel irrespective to the outcome of the fine FO estimation. To elaborate, the phase offset accumulates over time, causing different LTS samples to have different phase offsets. However, Bob complacently tries to interpret this time-varying phase offset as a fixed-value channel phasor by minimizing the MSE. Hence, his attempt to model the FO as if it is a channel parameter results in an incorrect estimated channel phasor, which after equalization rotates the payload's modulation symbols on the constellation map.

3.4.4 Optimal Jamming Strategy

Let $\varPhi_{eb} \overset{\text{def}}{=} \Delta\varphi_{eb} - \Delta\varphi_{ab}$. If the SJR at Bob is known, Eve can achieve the maximum possible $|\measuredangle\mathcal{C}(\varPhi_{eb})|$ value by optimally selecting $|\varPhi_{eb}|$. This maximization allows Eve to inflict the maximum subcarrier shift and overcome possible FO estimation inaccuracies due to noise at Eve or Bob. To calculate the optimal $|\varPhi_{eb}|$, we represent the total received jamming energy $|\tilde{u}|^2$ and signal energy $|\tilde{r}|^2$ in polar coordinates, as shown in Figure 3.6. Using geometric arguments, we find the maximum $|\measuredangle\mathcal{C}|$, where

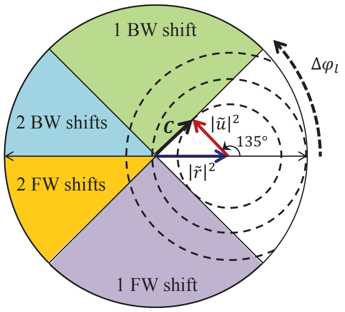


Figure 3.6: Superposition of Alice's and Eve's signals at Bob and the resulting subcarrier shifts. The minimum feasible $|\tilde{u}|^2$ occur when the vector $|\tilde{u}|^2$ is perpendicular to an edge of the 1-shift regions. The parts of a contour crossing the shaded areas show the feasible phases for a given $|\tilde{u}|^2$.

$\mathcal{C} = |\tilde{r}|^2 + |\tilde{u}|^2 e^{-j(\Phi_{eb})}$. Each circular contour in this figure shows the end points of the vector $\mathcal{C}$ for a given SJR but different Φ_{eb} values.

As long as $|\tilde{u}| < |\tilde{r}|$, $|\measuredangle\mathcal{C}|$ reaches its maximum when the vector $\mathcal{C}$ is tangent to the contour circle. In a right triangle, this implies

$$|\measuredangle\mathcal{C}| = \arcsin\frac{|\tilde{u}|^2}{|\tilde{r}|^2} \tag{3.11}$$

and

$$|\Phi_{eb}| = \pi/2 + \measuredangle\mathcal{C}. \tag{3.12}$$

When $|\tilde{u}| \geq |\tilde{r}|$, the maximum $\measuredangle\mathcal{C}$ equals to π, which is always achieved when $|\Phi_{eb}| = \pi$. In Figure 3.7, we plot the corresponding optimal $|\Delta f_{eb} - \Delta f_{ab}|$ during the STSs for different SJR values. Based on $\measuredangle\mathcal{C}$, we also derive the resulting number of subcarrier-spacings shift after LTSs. Note that phase offsets $\pi/2$ and π correspond to FOs of one and two f_Δ's, respectively. From the STSs perspective, LTSs adjust a phase offset to its closest multiple of $2\varphi_l$. So when $|\measuredangle\mathcal{C}| > 3\varphi_l$, the attack results in a shift of two subcarriers.

The jamming sequence can be designed to minimize the total jamming energy $\sum_{i=0}^{L-1} |\tilde{u}_i|^2$, subject to the constraint of at least one subcarrier shift, i.e., $|\measuredangle\mathcal{C}(\Phi_{eb})| \geq$

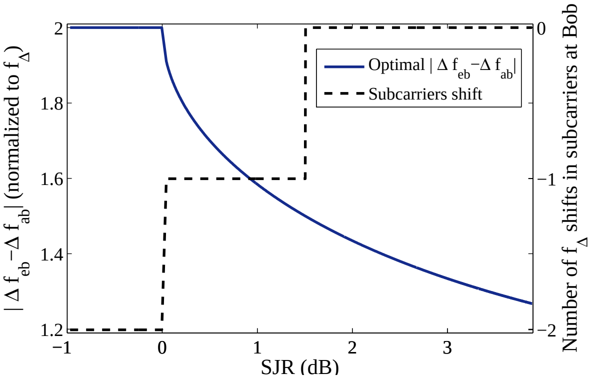


Figure 3.7: Optimal $|\Delta f_{eb} - \Delta f_{ab}|$ and resulting amount of subcarrier shift for different SJR values.

$\Delta\varphi_l$. The shaded area in Figure 3.6 shows the feasible region. According to (3.11) and the geometry in Figure 3.6, we conclude that:

1. The energy minimization problem is feasible as long as

$$\text{SJR} = \frac{\sum_{i=0}^{L-1} |\tilde{r}_i|^2}{\sum_{i=0}^{L-1} |\tilde{u}_i|^2} \leq \frac{1}{\sin(\Delta\varphi_l)} = \sqrt{2} \approx 1.5 \text{ dB}. \tag{3.13}$$

2. The minimum jamming energy is achieved when

$$|\Delta\varphi_{eb} - \Delta\varphi_{ab}| = |\pi/2 + \Delta\varphi_l| = 3\pi/4, \tag{3.14}$$

   or equivalently, $|\Delta f_{eb} - \Delta f_{ab}| = 1.5\, f_\Delta$.

Equation (3.14) says that the phase offset of Eve's signal as perceived by Bob should have phase difference of $|\pi/2 + \Delta\varphi_l|$ relative to Alice's signal. Even if $\Delta\varphi_{eb}$ does not satisfy (3.14), Eve can augment the hardware-dependent Δf_{eb} and obtain an *effective* Δf_{eb} by imposing an artificial FO of Δf_n on the jamming sequence before it is transmitted by the oscillator. This is achieved by multiplying the samples of the jamming sequence by $e^{-j2\pi\Delta f_n\, i\, t_s}$, where Δf_n is given by:

$$\Delta f_n = \pm 1.5\, f_\Delta - \Delta f_{eb} + \Delta f_{ab}. \tag{3.15}$$

The optimal $|\Phi_{eb}|$ that minimizes the jamming energy is particularly important in designing the optimal jamming strategy because the SJR at Bob is usually unknown to Eve. The optimal jamming strategy to deal with this situation is to consider the worst-case (highest) SJR under which the attack is successful and then set the effective FO according to (3.14). Therefore, Eve always sets Φ_{eb} to $\pm(\pi/2 + \Delta\varphi_l)$.

3.5 Discussion

OFDM-based 802.11 systems employ interleaving and adaptive modulation and coding (AMC) schemes to increase resiliency against jamming and bit errors. However, the achieved BER value of the aforementioned FO attack (~ 0.5) is high enough that the mutual information between the transmitted and received sequences is zero, and hence practical coding schemes cannot recover the frame. After an unsuccessful transmission and subsequent data rate reduction, Alice may increase her transmit power for the whole frame. In the case of the proposed FO attack, such an increase is unnecessary and inefficient for the payload, which constitutes up to 99.9% of a frame. In addition, an intelligent jammer can track Alice's power increase (e.g., by overhearing management frames), adjust the jamming power to always achieve the optimal SJR, and force the dropping of subsequent transmissions.

It may also be argued that because pilot subcarriers are transmitted on known frequencies, Bob can compare the known symbols of the pilot subcarriers with the received symbols on different subcarriers to identify a possible subcarrier shift. However, because channel estimation is distorted, locating the corrupted pilot subcarriers at Bob is quite challenging. Furthermore, these pilot subcarriers cannot be easily used for channel estimation (we leave the investigation of this problem to a future work).

Moreover, we note that jamming the LTSs after jamming the STSs strengthens the attack by further distorting the channel estimation process. However, jamming the LTSs alone cannot lead to a subcarrier shift even though it involves more jamming effort (8-μs duration on 48 subcarriers) than jamming three STSs (≤ 3 μs on 12 subcarriers). Furthermore, with LTSs jamming, pilot subcarriers can still be used to estimate the channel and correct any residual FO.

The system model in this chapter assumes a single Tx-Rx-pair (i.e., Alice and Bob, and hence their FO, are known). In the case of multiple Tx-Rx pairs, Eve can construct a database of the FOs between different Tx-Rx pairs. Benefiting from CSMA/CA channel access mechanism, Eve can consider one transmission at a time and then leverage protocol semantics (e.g., data-ACK exchanges) to guess the Tx and Rx of an upcoming transmission. Further investigation of this issue is left for future work.

3.6 Performance Evaluation

In this section, we evaluate the effectiveness of the FO estimation attack through simulations and USRP experiments. We implemented the 802.11a/g preamble (including both short and long training sequences) by extending the PHY-layer library functions of LabVIEW. Alice appends 1500 modulated random bits to the frame preamble. Pilot-based channel and FO estimation and channel coding were not implemented to concentrate on the specific effects of the FO attack on received uncoded bits. The impact of coding and pilot subcarriers was discussed in Section 3.5.

We assume that Bob uses the STSs t_9 and t_{10}, as defined in Figure 6.1, for coarse FO estimation, followed by fine FO estimation using LTSs. Channel estimation is performed over the first LTS using the time domain method [26]. We first evaluate the performance under a simulated AWGN channel model and later in a multi-path indoor environment. (More results are provided in [39].) We vary the SJR, the SNR (noise level), the modulation scheme, and Eve's effective FO, denoted by D_{eb}. In particular, we consider BPSK, QPSK, and 16-QAM modulation schemes. We measure $\widetilde{\Delta f_s}$ as well as final estimated FO, SER, and BER.

We compare three cases: 1) jamming the last STSs with a random signal (see Section 3.4.2), 2) FO attack with pairing rule only ($V = 1$ and $L = 16$ for frame detection), and 3) the entire FO attack including the chaining rule, with $L = 16$ and $V = \log_2 L$. The purpose of evaluating the second case is to study the impact of the chaining rule. The jamming duration for the second case is always equal to the sum of the durations of t_8 and t_9. However, it is not constant when the chaining rule is applied, and depends on m_{V-1}.

3.6.1 Simulations

We consider an AWGN channel model without signal attenuation. In our simulations, the SJR is normalized to the energy of two full STSs. However, the chaining rule results in a variable-length cyclic postfix extension, which sometimes has a slightly higher sample power than the average sample power over an STS.

Frame Detection and Jamming Duration

Initially, we assess the accuracy of our adaptive fast frame detection method at Eve and also its impact on the jamming duration. Even though our adaptive detection method uses a small window of $L = 16$ compared to $L = 48$ for the default scheme, adapting ϵ based on finding V frame-start candidates increases the probability of precise frame detection even for the first candidate. This is shown in Figure 3.8, where each probability is calculated based on more than 25000 runs. By including additional $V - 1$ candidate start times, we further increase the probability of including the true start time in V candidates, specially under high noise levels. The chaining rule benefits from V start times because it equally likely considers all the candidate start times to construct the jamming signal.

The jamming duration depends on m_{V-1} and the amount of postfix extension. In Table 3.1, we report the average index-distance between the first and the last samples (m_{V-1}) in the set of V start times when an STS contains $L = 16$ samples. The table

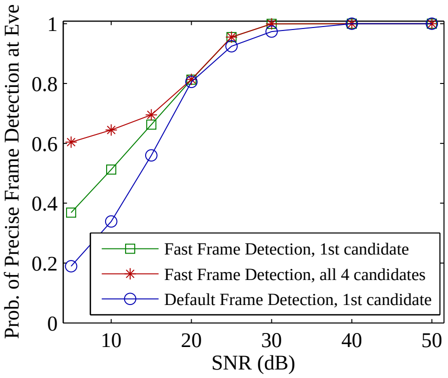
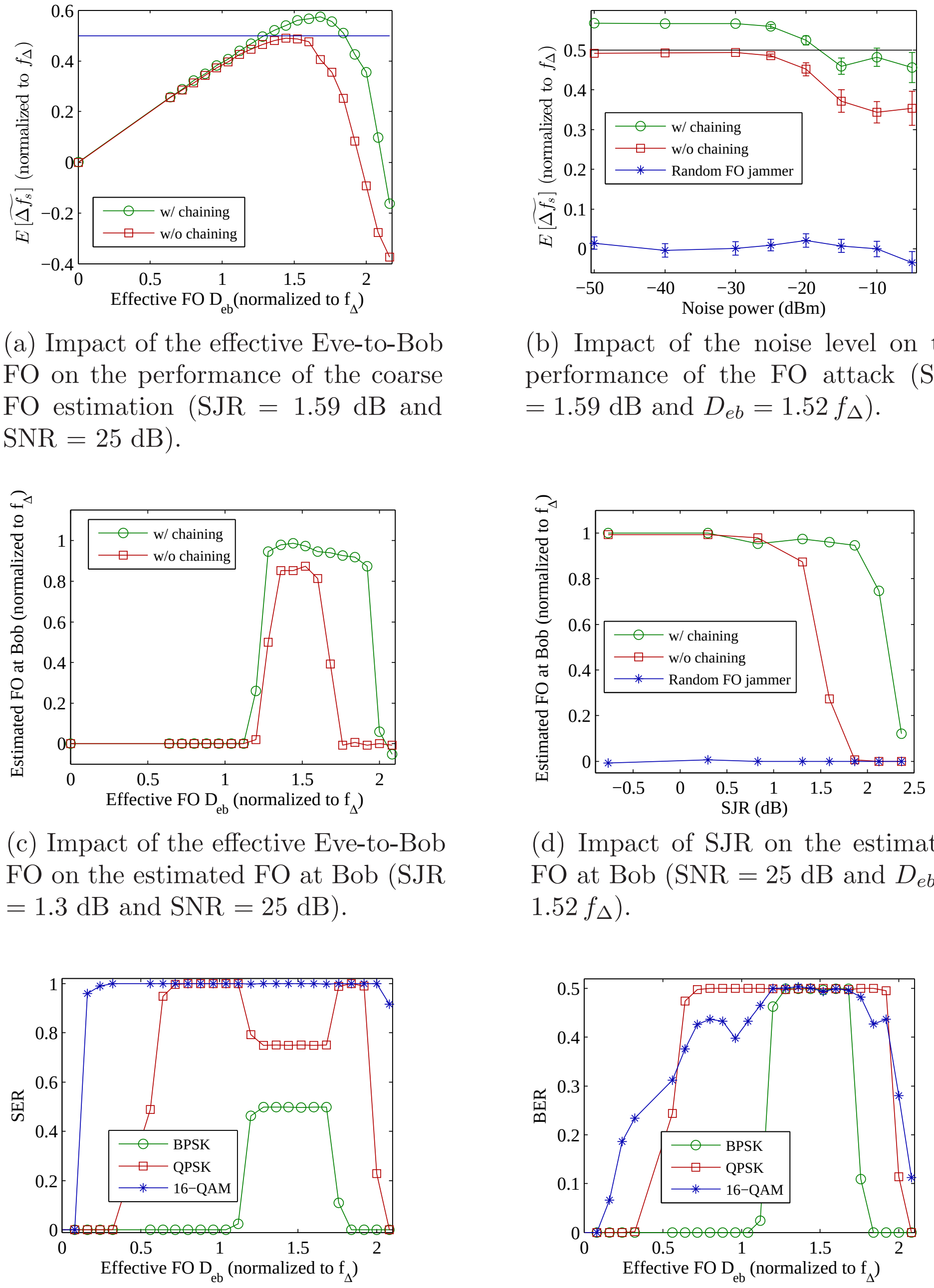


Figure 3.8: Performance of different variants of frame detection vs. SNR (simulations).

shows that even at low SNR, the amount of cyclic extension due to the chaining rule is often less than half an STS. In particular, in 99.88% of the cases, $m_{V-1} \leq 8$, which means the jamming duration will be less than $3.5\lambda_{STS}$ or, equivalently, 0.7 of an OFDM-symbol duration. A 1500-bit BPSK-modulated payload lasts for 32 OFDM symbols, equivalent to 160 λ_{STS}. The durations of 16-QAM-modulated and QPSK-modulated signals of the same payload will be 40 and 80 λ_{STS}, respectively. So the jamming effort in our simulations is upper bounded by 2.0%, 3.5%, and 5.9% for BPSK, QPSK, and 16-QAM-modulated payloads, respectively. In general, an 802.11a frame lasts for $20 \times 10^{-6} + \lceil(22 + \text{LENGTH})/\text{DATARATE}\rceil$ seconds [1], where LENGTH and DATARATE denote the encoded payload size (in bits) and the data rate, respectively. For a typical 802.11a frame [29], the jamming effort varies between 0.07% and 0.88%, depending on the code and data rates. This is 30% less than the effort of the OFDM symbol jamming attack in [29].

SNR (dB)	5	10	15	20	25	30
$E(m_{V-1})$	4.05	3.74	3.54	3.23	3.02	3.0

Table 3.1: Average value of m_{V-1} in the chaining rule for different SNR levels.



(a) Impact of the effective Eve-to-Bob FO on the performance of the coarse FO estimation (SJR = 1.59 dB and SNR = 25 dB).

(b) Impact of the noise level on the performance of the FO attack (SJR = 1.59 dB and $D_{eb} = 1.52\, f_\Delta$).

(c) Impact of the effective Eve-to-Bob FO on the estimated FO at Bob (SJR = 1.3 dB and SNR = 25 dB).

(d) Impact of SJR on the estimated FO at Bob (SNR = 25 dB and $D_{eb} = 1.52\, f_\Delta$).

(e) SER performance for different modulation schemes (SJR = 1.59 dB and SNR = 30 dB).

(f) BER performance for different modulation schemes (SJR= 1.59 dB and SNR = 30 dB).

Figure 3.9: Performance of different variants of the FO attack and of random FO jamming under different noise levels, D_{eb} and SJR values, and modulation schemes. The transmission power is 0 dBm. (simulation results)

FO Estimation

Figure 3.9(a) depicts the average $\widetilde{\Delta f_s}$, measured after the corrupted STSs of 150 frames, when SJR= 1.59 dB, transmission power is 0 dBm, and noise level is -25dBm. The horizontal line represents th_l, normalized to f_Δ. The chaining rule improves the jamming effectiveness and guarantees a range of effective FO values for which the attack is successful ($\widetilde{\Delta\varphi} > \Delta\varphi_l$). When the chaining rule is not applied, the jamming attack is optimal at the optimal effective FO derived in Section 3.4.4, but is still insufficient to pass the threshold because of frame detection errors. When the chaining rule with V candidates is applied, the maximum average $\widetilde{\Delta\varphi}$ occurs later than the maximum for the no-chaining case because of slightly higher power during postfix samples. Figure 3.9(b) shows the effect of noise on the STS-based estimated FO when SJR= 1.59 during the last three STSs and with $D_{eb} = 1.52\, f_\Delta$, a near-optimal value for this setup. The 90% confidence intervals are shown for each point. The increase in frame timing errors due to noise reduces the effectiveness of the attack, but this increase has less impact when the chaining rule is applied. When the noise level is higher than -20 dBm, the gap between the curves belonging to the two modes of the FO attack is wider, showing that the chaining rule is more robust in highly noisy channels .

When $|\widetilde{\Delta\varphi}| > \Delta\varphi_l$, the LTSs round the estimated FO to the nearest multiple of $2th_l$. Otherwise, LTSs try to round the FO to zero. In Figure 3.9(c), we plot the average final estimated FO at Bob when SJR= 1.3 dB during the last three STSs and the noise level is -25 dBm throughout the frame. The chaining rule improves Eve's ability to shift the subcarriers by one f_Δ. With respect to the SJR, we can observe in Figure 3.9(d) that when Eve'e D_{eb} is close to its optimal value, Eve is not able to guarantee a successful attack without the chaining rule even with the optimal SJR value of 1.5 dB.

Impact of the FO Attack on Modulation Performance

Under a relatively high SNR (e.g., 30 dB in our simulations) and without the FO attack, the SER is very close to zero. The FO attack impacts both the channel and FO estimations. We measure the overall impact for different modulation schemes by measuring the SER and BER. First, we consider the case when $\widetilde{\Delta\varphi} < \Delta\varphi_l$ and the LTSs are still able to correct the FO. In this case, Bob tries to minimize the error of estimating a channel phasor that is supposedly responsible for the phase shift accumulations over LTS samples. Because the phase shift $\Delta\varphi = 2\pi\Delta f t$ is linear in time, the best estimate is a phasor that equals to the average phase shifts. As long as $|\widetilde{\Delta\varphi}| \leq \Delta\varphi_l$ (i.e., the resulting FO is still less than th_l), the maximum phase offset between the first and last samples in an LTS is π, which implies that the error in phasor estimation is always less than $\pi/2$. On the constellation map, This error will cause an identical rotation of all the payload's modulated samples [36]. We select to apply channel estimation to one LTS to limit the amount of rotation. Figure 3.9(e) shows the SER for different modulation schemes. Clearly, BPSK is the most resilient scheme against channel phasor estimation error. Once $|\widetilde{\Delta\varphi}| > \Delta\varphi_l$ and the subcarriers are shifted, the sequence of modulated samples of any modulation scheme looks random relative to its original sequence, resulting in the highest possible SER, i.e., $(|M|-1)/|M|$, where $|M|$ is the modulation order.

The BER under higher-order modulation schemes, however, is less affected by the attack if the subcarriers are not shifted but the symbols are rotated to neighboring regions, as shown in Figure 3.9(f). With the increase of D_{eb}, first the BER of 16-QAM starts to increase due to symbol errors. However, once QPSK also experiences symbol errors, its BER will be larger than the one for 16-QAM. Because of the Gray code structure, higher-order modulations guarantee lower BER when one of the neighboring symbols is mistakenly demodulated instead of the true symbol. Nonetheless, as long as the FO attack shifts the subcarriers, the BER stays at its maximum (0.5), irrespective of the modulation scheme.

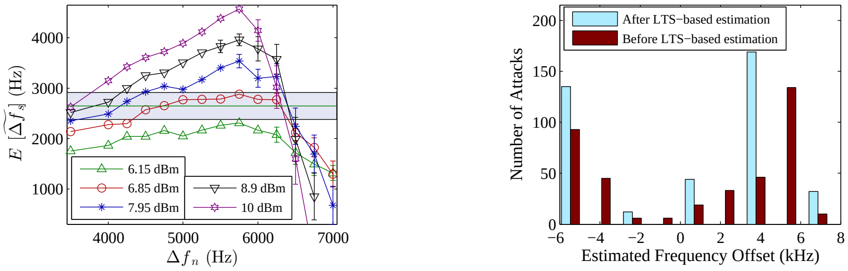
3.6.2 USRP Experiments

We experimentally evaluate the impact of the proposed FO attack using an NI-USRP 2922 testbed, operated in an indoor environment in the 2.4 GHz band and controlled by Windows-based hosts. Our setup consists of three USRPs, acting as Alice, Bob, and Eve. To estimate the FOs between the USRPs, we connected Alice and Eve devices to Bob through an SMA cable and conduct 4000 FO estimations. Δf_{ab} and Δf_{eb} were measured to be 1086 and 340 Hz with standard deviations of 270 and 230 Hz, respectively. Based on the estimated FOs, effective FO D_{eb} was approximately found to be $715.5 + \Delta f_n$ Hz with standard deviation of 355 Hz. In our experiments, we fix the locations of Alice and Bob and move Eve to create two scenarios: LOS and non-LOS (see Table 3.2). In the non-LOS case, a metal shelf is placed between Eve and the other two. At each location, Eve launches the attack with different jamming powers and different values of Δf_n. In the experiments, Alice's transmission power is set to 7.9 dBm.

Scenario	Alice-Bob	Eve-Alice	Eve-Bob
LOS	1.5 m	1.77 m	1.77 m
NLOS	1.5 m	4.74 m	5.15 m

Table 3.2: Distances between Alice, Bob, and Eve for two different scenarios.

The USRP-based implementation of our reactive attack faced two challenges. First, the internal buffer size of the USRPs, which is used to store the samples before forwarding them to the host PC, is not big enough to store the samples captured at the nominal rate of 20 MHz. So we had to reduce the symbol rate to 0.2 MSPS. As a consequence, λ_{STS} expanded to 80 μs and f_Δ dropped to 3125 Hz (i.e., the total bandwidth of 200 kHz). Second, the USRP's reaction time (which consists of the communication delay between a USRP and its host PC through an Ethernet cable, the host's processing delay, and the time to initialize for transmission) exceeded several milliseconds. So Eve will miss the rest of the frame before she starts her jamming[5]. To overcome these challenges, we made the following modification in the implementation. We let Alice send several back-to-back frames periodically with a known period of T ms. Upon being triggered by a received power increase, Eve captures 2 μs worth of the sequence. If a frame is detected, she assumes that the next frame starts exactly T ms after the start of the this one. The host PC at Eve then constructs a jamming signal based on the timing information of the first detected frame and sends it to the USRP. After initialization, the USRP's onboard timer, which has nanosecond 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 and other subsequent frames.



(a) Effect of Δf_n on STS-based FO estimation with different jamming powers.

(b) Effect of LTS-based FO estimation on $\widetilde{\Delta f_s}$ ($\Delta f_n = 7000$ Hz and jamming power = 8.9 dBm).

Figure 3.10: USRP results: Performance of the FO attack in LOS scenario.

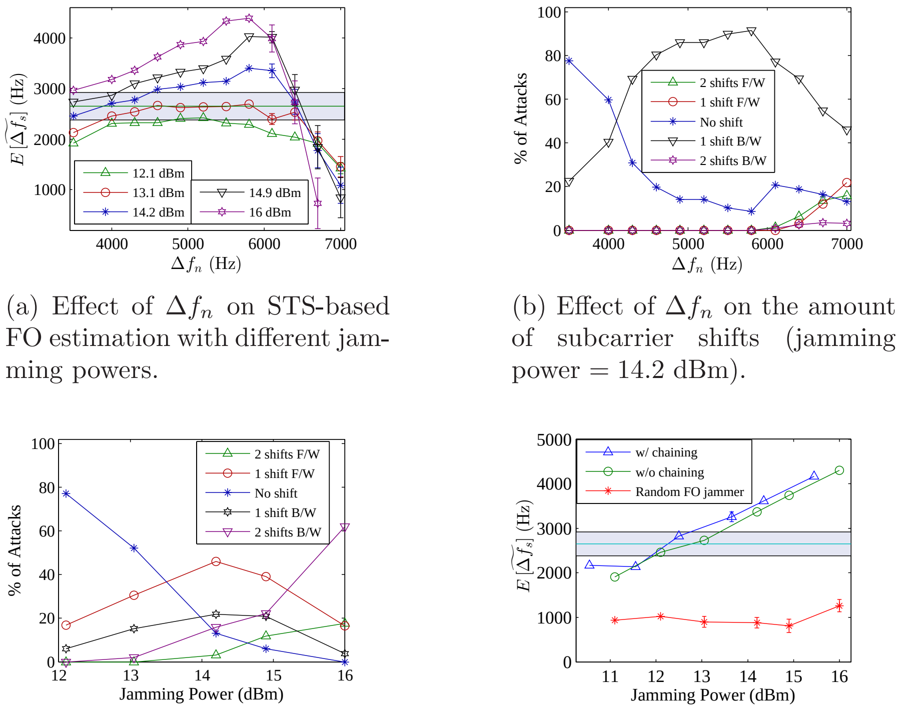
Figure 3.10(a) shows the average STS-based estimation of Δf_{ab} in the LOS scenario for different values of Δf_n and jamming powers. Because our USRPs do not have stable oscillators and hence Δf_{ab} varies with time, we represent the probable value of $\Delta f_{ab} + f_\Delta/2$ by a shaded area whose height is twice the standard deviation of Δf_{ab}. Eve is able to shift the subcarriers by pushing $\widetilde{\Delta f_s}$ beyond the actual value of $\Delta f_{ab} + f_\Delta/2$. The results show that even though Eve-Bob distance is larger than

[5] This is not the case for an off-the-shelf reactive jammer, which usually has an onboard processor and dedicated hardware. In addition, implementing a correlation-based reactive jammer on the USRP's FPGA can achieve a reaction time of 2.56 μs [40].

Alice-Bob distance, Eve can shift the subcarriers using almost the same power as Alice's power if Δf_n is optimally selected. In particular, when the jamming signal is 7.95 dBm, Eve is successful in shifting the subcarreirs in 84% of the attacks if $\Delta f_n = 5500$ Hz. This validates our optimal Δf_n selection scheme (see Section 3.4.4) since the "estimated" optimal Δf_n in our setup is $715.5 + 4687.5 = 5403$ Hz. After STSs, LTS-based estimation rounds the FO to the nearest multiple of f_Δ. In Figure 3.10(b), we depict a histogram to show/compare the number of jamming attacks that result in different ranges of FO estimates at Bob, before and after LTS-based estimation. It shows how LTSs can complacently exacerbate the FO estimation error. We show the results for the NLOS scenario in Figure 3.11(a). As seen in this figure, the lower the jamming power, the smaller is the optimal value of Δf_n, which is inline with Figure 3.7.

In the above results, we notice that the 95% confidence intervals at higher Δf_n values are noticeably larger than those at smaller Δf_n values. According to Figure 3.6, higher values of effective Δf_{eb} may result in estimating a negative FO (when $\measuredangle \mathcal{C} > \pi$) and thus the variance of $\widetilde{\Delta f_s}$ increases. A negative FO estimate results in forward subcarrier shifts, instead of backward shifts. To illustrate this behavior, in Figure 3.11(b) we plot the impact of Δf_n on the amount of subcarrier shift when the jamming power is 14.2 dBm. The attack achieves the highest success rate when $\Delta f_n = 5800$ Hz. As Δf_n increases further, the success rate slightly decreases, but Eve can impose various amounts of subcarrier shift, which can be leveraged to make it more difficult for Bob to guess the amount of subcarrier shift. Figure 3.11(c) shows the effect of jamming power on the amount of subcarrier shift when Δf_n is high. As the jamming power increases, Eve not only can achieve a higher success rate, but can also impose more than one subcarrier shift (forward or backward).

We compare the FO attack, with and without the chaining rule, against a random FO jammer in Figure 3.11(d). In this experiment, we configure Eve's USRP to start jamming zero, one, or two time indices before the estimated start of STSs. The



(a) Effect of Δf_n on STS-based FO estimation with different jamming powers.

(b) Effect of Δf_n on the amount of subcarrier shifts (jamming power = 14.2 dBm).

(c) Effect of jamming power on the amount of subcarrier shifts ($\Delta f_n = 7000$ Hz).

(d) Performance of different variants of the FO attack vs. jamming power ($\Delta f_n = 5700$Hz).

Figure 3.11: USRP results: Performance of the FO attack in the NLOS scenario. Alice's signal power is 7.9 dBm.

random FO jammer generates uniform white noise. The results confirm that the chaining rule strengthens the attack while random jamming cannot manipulate Δf_s and overcome the LTSs even with high jamming power.

Finally, we launch the FO attack during the transmission of a packetized image. Specifically, Eve attacks 24 packets in the middle of the transmission of 44 QPSK-modulated 720-byte-long packets that represent the image in Figure 3.12(a). In Figure 3.12(b), we show the received image. The parts that experience FO jamming are completely destroyed.

3.6.3 Comparison to Existing Jamming Attacks

Vulnerabilities of wireless protocols and Denial-of-service (DoS) attacks have been studied in the literature since the early 2000s. DoS attacks can be applied at either MAC or PHY-layer. MAC layer attacks usually take the form of malicious packet insertion, whereas RF jamming is a form of PHY-layer attack.

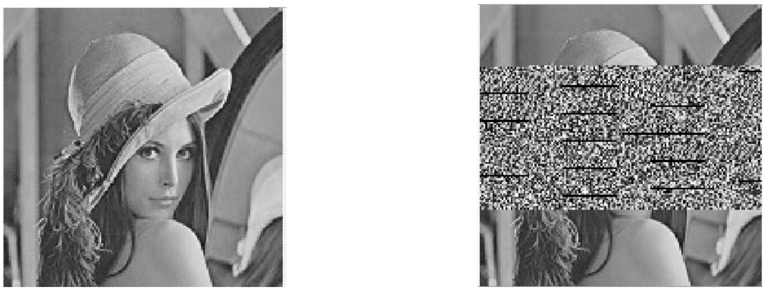
RF Jamming techniques are categorized into constant, deceptive, random, reactive, and short noise-based (narrow-band) intelligent jamming methods [30, 41]. Constant, deceptive, and random jamming models achieve a high level of DoS by excessively transmitting over the channel, but exhibit poor energy efficiency and high detection probability [41]. On the other hand, energy-efficient reactive jamming attacks select and target a (part of a) packet based on traffic analysis, protocol semantics, or publicity of some fields [20, 21, 29, 30, 33, 40]. These attacks may fail to significantly corrupt ongoing transmissions if, for example, channel hopping, randomization, and coding are used to hide the transmission features.

The efficiency of reactive jamming is assessed by the effort needed to drop a packet. In [29], jamming efficiency is defined as the ratio of communication effort to jamming effort. The authors demonstrated the jamming efficiency of $50 \sim 500$ in 802.11a by jamming an OFDM symbol. Using a high duty-cycle jammer, Gummadi *et al.* [33] could disrupt a link when the jamming power is 1000 weaker than the signal power by targeting timing recovery, dynamic range selection (AGC), and header processing. The authors in [19] observed that 22 μs of jamming is sufficient to make a frame undecodable. In comparison, our FO attack can achieve a jamming efficiency of $136 \sim 1400$ in 802.11a and defeat any ECC by jamming for only 2.8 μs.

Jamming OFDM systems is of particular interest due to their widespread use in modern systems. Simple barrage jamming targets the entire spectrum/tones and corrupts more bits than the more power efficient but less destructive partial band, single- and multi-tone jamming [22, 42]. Asynchronous off-tone jamming attacks exploit the uncompensated FO between Eve and Bob to transmit one or multiple subcarriers that will be received between some of the data subcarriers [22]. This creates significant ICI for those subcarriers. Though energy-efficient, these attacks cannot achieve 50% BER. Furthermore, because coding and interleaving are employed in the 802.11 systems for robustness against narrow-band interference, Bob may still be able to recover the frame. Several pilot jamming attacks were proposed in [22, 32] in order to distort channel estimation. In contrast, our proposed attack lasts for less than the duration of a pilot symbol jamming and corrupts the channel estimation without jamming pilots. Jamming against timing acquisition in OFDM systems and some countermeasures were discussed in [20, 34, 35]. However, OFDM systems are more sensitive to FO than timing errors. This vulnerability was first revealed in [21]. The structured but essentially random jamming scheme in [21], however, does not provide any performance guarantee and may have higher jamming effort than ours. Interested readers are referred to [39, 43] for more details about jamming attacks against OFDM systems.

3.7 Defense Strategies

Alice and Bob may work cooperatively or independently to mitigate the previously presented FO attack. For example, they may prevent accurate estimation of Δf_{ab} at Eve by transmitting Tx-based friendly jamming. This, however, requires additional antennas. Bob may also use the power-spectral density of the captured signal after LTSs to identify the missing subcarriers, and thus determine the overall subcarrier shift. This technique, however, fails if Eve transmits only one or two bogus subcarriers to replace missing subcarriers. Assuming that Bob is not equipped with additional antennas, in this chapter we propose three preliminary approaches for mitigating the FO attack. In Chapter 6, we further propose a mitigation technique by altering the default preamble signal in a way that Bob is still able to perform preamble functions as normal. However, Eve will no longer have the knowledge of the preamble samples u_i because the preamble is no longer public. Hence, she will



(a) Transmitted image

(b) Received image

Figure 3.12: Applying the FO attack on a sequence of frames belonging to an image.

not be able to design its jamming samples r_i based on u_i's and account for unknown channel coefficient and timing errors. Analysing and evaluating these strategies are beyond the scope of this dissertation, and will be addressed in future research.

1) Randomizing FO Sequences (Sequence Hopping): Because of the redundancy in the STSs, Bob can choose any pair of the ten consecutive STSs, to perform FO estimation. Furthermore, due to the maximum FO requirement for 802.11 devices (212 kHz = $1.3568\,th_s$ for devices operating in the 5 GHz band and 125 kHz = $0.8\,th_s$ for devices in the 2.4 GHz band [24]), the two autocorrelation windows do not necessarily need to be contiguous. In fact, the two windows can be two or four STSs apart (i.e., each sample is three or five STSs away from its dual) in the 5 GHz and 2.4 GHz bands, respectively. This means that Bob has the flexibility to randomly hop to any pair of STSs for FO estimation, given that the STSs in this pair are not more than two or four STSs apart, depending on the frequency band. Even if Bob selects an STS that is corrupted by a jamming signal together with a jamming-free one, he is still able to estimate the same FO as if two jamming-free sequences are selected [36]. To implement sequence hopping, Bob can record the received signal (the ten STSs) while he is in the process of detecting the start of the frame. Once the frame has been detected, Bob randomly chooses two STSs for FO estimation, while satisfying the maximum STS-distance constraint.

2) Preamble Obfuscation: Preamble obfuscation aims at making the timing or FO features hard to extract by Eve. We provide one simple example for timing extraction. Alice can obfuscate the preamble by adding artificial noise that is only known to Bob. He, on the other side, modifies the denominator in (2.2) to account for the power of the artificial noise during a certain section of the received signal. 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, Bob can still detect the frame by doubling the denominator in (2.2), but the increased power at Eve would decrease the value of $\mathcal{M}(n)$ through (2.2) for the first $L/2$ samples. Hence, $\hat{\mathcal{M}}$ will be in the second half, making the chaining rule fail because Eve does not include the actual start time in the $V = \log_2 L$ start times.

3) STS Bypassing: Bob can simply disable the STS-based FO estimation mechanism to dodge the FO attack. However, he still has to meet the requirement for coarse FO estimation, i.e., unambiguous phase estimation (see Section 3.2). Under BPSK modulation, which is typically used to transmit the PHY header, Bob can tolerate channel estimation errors due to an FO estimation error of up to 15 kHz [13]. Hence, Bob can divide the range of possible FO values into several equal-size frequency bins, each of 30 kHz bandwidth. He can then try each of the possible bins and compensate for its center frequency before applying LTS-based FO estimation. The center frequency that results in the minimum MSE in channel estimation can be considered as $\widetilde{\Delta f_s}$. Bob may also suppress both STS- and LTS-based FO estimation, and instead rely on pilot subcarriers for FO and channel estimation. This approach, however, often gives rise to ICI because adjacent subcarriers interfere with the pilot subcarriers (which even are not yet channel-equalized) and the FO estimation will be erroneous.

3.8 Summary

We demonstrated the vulnerability of OFDM systems against a highly disruptive but efficient-efficient DoS attack. This attack succeeds even when the PHY frame is shielded by interleaving and channel coding. The attack focuses on the frequency offset (FO) estimation process, and is channel-independent and robust to time-synchronization errors at Eve through applying the proposed pairing and chaining rules. Through this attack, a reactive jammer exploits and targets a small portion of the publicly known preamble used for FO estimation. The attack lasts for less than the duration of an OFDM symbol, i.e., less than 1% of a typical frame duration, and is at least 30% more efficient than previously reported attacks. Though short-lived, the attack results in a shift in subcarrier indices and the maximum possible BER (50%) even when the jamming signal at Bob is ~ 1.4 times weaker than Alice's signal. We verified via simulations and USRP experimentation. The simulation results show that different modulation schemes are equally susceptible if the FO attack can shift the subcarrier indices, and higher modulation orders are more affected when the attack impacts only the channel estimation. Finally, we sketched several possible mitigation approaches, whose detailed analysis and evaluation are left for future.

CHAPTER 4

Friendly Jamming in Multi-link Scenarios

4.1 Introduction

Information theoretic secrecy [44, 45] at the PHY layer is a lightweight approach that aims at preventing Eve from *decoding* a plaintext frame or extracting SCI from unencrypted headers. Alice and its legitimate receiver (Bob) are guaranteed secret communications if the Alice-Bob channel is better than Alice-Eve channel. In [44], the notion of *secrecy capacity* was introduced as the maximum rate at which Alice can securely transmit information to Bob. This rate is the difference between the mutual information between Alice and Bob, to that between Alice and Eve.

Non-zero secrecy capacity is not always possible. For example, if Eve is closer to Alice than Bob, then the Alice-Eve channel may be better than the Alice-Bob channel, resulting in zero secrecy capacity [44]. Friendly jamming (FJ), proposed in the pioneering work of Goel and Negi [23], can be used to degrade the Alice-Eve channel without harming Bob's reception. Essentially, a FJ signal is a randomly generated artificial noise. To nullify the FJ signal at Bob, the authors in [23] considered the case when Alice has multiple antennas. Alternatively, a bank of relay nodes can be utilized to transmit the artificial noise in the null space of the Alice-Bob channel.

4.1.1 Existing FJ-based Schemes for Multi-link Scenarios

Although FJ-based PHY-layer security has been extensively considered for single-link scenarios, only a few papers studied the problem in multi-link scenarios. Research efforts on secret communications in a multi-link network can be classified into two broad categories: Large-scale [46–48] and small-scale wireless networks [49, 50]. Considering a large-scale wireless network consisting of n nodes, the authors in [46] derived the per-node *asymptotic* secrecy capacity. They also proposed to use "Rx-based FJ", whereby legitimate full-duplex receivers are able to cancel the self-interference resulting from their generation of FJ signals. For the case of independent eavesdroppers, it was shown that a per-node secrecy capacity of $\Theta(\frac{1}{\sqrt{n}})$ is achievable, which is the same per-node capacity without secrecy considerations. These results imply that the per-node secrecy capacity is not affected by the presence of eavesdroppers. However, placing the FJ devices at the same locations of the communicating nodes may not be optimal from an energy consumption perspective. The interference of Rx-based FJ on other receivers was also not considered in [46].

The authors in [48] explored allowing a fraction of transmitters to cooperatively send their signals to their receivers through relay nodes, i.e., two-hop communications. The idea is based on the work in [51]; wherein, for each Alice-Bob pair, a relay node with "good" channels to Alice and Bob is selected. Relay nodes with "bad" channels to the selected relay or to Bob are used to produce FJ signals to confuse passive eavesdroppers. Instead of generating FJ signals, simultaneous transmissions are exploited in [48] to create high interference at the eavesdroppers. In this sense, the messages of other Alice-Bob pairs are utilized as FJ signals. Secrecy is guaranteed only as n tends to infinity. The results of large-scale wireless network, however, may not be always applicable to small-scale networks that can have irregular topologies.

Secure minimum-energy routing with the aid of FJ was investigated in [49,50] for a small-scale network. The objective is to compute a minimum-energy path subject to constraints on the end-to-end communication secrecy and the throughput over the path. The authors proposed a scheme to assign FJ power required to secure individual links. Each link was studied independently, assuming that it can be secured by its own set of FJ devices, and there is a discrete set of eavesdropping locations, each with a given probability of eavesdropping in that location. The secure routing problem was reduced to finding a path with the minimum total information and FJ power. These works, although applicable to small-scale networks, do not jointly consider the optimal placement and power allocation of the FJ devices. Moreover, they do not exploit the FJ devices associated with a given link to help in providing secrecy for another link, which can reduce the total jamming power. Finally, they assume that the FJ signals are nullified at legitimate receivers, but the conditions needed to ensure such nullification are not incorporated in the formulation. This undermines the applicability of their designs.

4.1.2 Main Contributions and Chapter Organization

The example in Figure 4.1 illustrates the importance of optimizing the placement of the FJ devices. Eve 1 and 2 are the most vulnerable eavesdroppers due to their proximity to Alice. If the FJ devices are placed near Bob (e.g., Rx-based jamming), they would need high power to deafen Eve 1 and 2. The FJ device would consume less power to deafen Eve 1 and 2 if they are moved to the potentially optimal location shown in Figure 4.1. Note that if the FJ devices are instead moved to the centroid of the eavesdropping locations or near Alice (e.g., Tx-based jamming), they will have a larger distances to Eve 1 and 2 and consume larger amount of power to deafen them.

In this chapter, we consider the problem of placing FJ devices in a small-scale multi-link wireless network, e.g., peer-to-peer (P2P) or multihop, and address the aforementioned limitations of existing PHY-layer solutions. Our contributions are summarized as follows:

- We first consider a per-link strategy and formulate an optimization problem that aims at jointly optimizing the power allocation and placement of the FJ devices for a given link under secrecy constraint. We show that our proposed scheme reduces power consumption by 55%–99% compared to the case in which

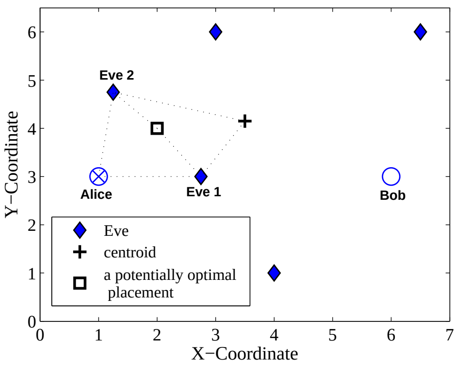


Figure 4.1: Possible placements of FJ devices (without nullification constraints.) Jamming power is a function of distances to and received information signals at each Eve.

the optimal placement of the FJ devices is not considered.

- We then consider the joint power allocation and placement of FJ devices under secrecy constraint for all links jointly (network-wide strategy). The exploitation of the FJ devices to simultaneously cover more than one link saves more energy and reduces the number of FJ devices relative to per-link case.

- We use distributed MIMO techniques to create a null region around all the legitimate receivers in network-wide scenario (the given receiver in per-link scenario) and accordingly establish sufficient conditions on the jamming powers and locations of the FJ devices. We incorporate these conditions as constraints in the formulation.

- We propose a novel link weight and a corresponding routing metric for the multihop scenario.

The rest of the chapter is organized as follows. In Section 4.2, we explain the underlying distributed MIMO mechanisms for generating FJ signals, and derive sufficient conditions on the powers and locations of FJ devices to ensure FJ nullification at legitimate receivers. In Section 4.3 we present the network model, formulate our problem, prove that it is an NP-hard signomial programming problem, and then approximate it by a geometric programming problem, which can further be convexified. In Section 4.4 we propose a novel link weight for the secure routing problem. Section 4.5 provides the simulation results. Finally, Section 4.6 concludes the chapter.

4.2 Distributed MIMO for FJ Nullification

If the FJ signals are to be generated by the same MIMO node (e.g., Alice), then the phases of these signals can be easily controlled to provide the desired nullification. A set of FJ signals can add destructively and nullify each other at an intended receiver if these signals, each of which traverses a different channel, are received out-of-phase and sum up to zero. To achieve this, techniques such as zero-forcing beamforming are employed to determine the phase and amplitude of each FJ signal at the transmit antennas. However, in general, FJ signals may be produced by different devices that do not share a reference clock and so are not synchronous. Hence, the signals transmitted from distributed FJ nodes may experience unknown random delays. In this section, we explain how we synchronize FJ devices, each equipped with a single antenna, and establish the sufficient conditions on the jamming signals to ensure nullification of FJ signals at all legitimate receivers.

4.2.1 Synchronization of FJ Devices

To enable synchronized FJ devices, we exploit SourceSync's synchronization protocol proposed and empirically demonstrated in [38] for OFDM systems. According to this method, a set of distributed cooperative transmitters choose a leader, who initiates the synchronization process by transmitting an OFDM-based sync header. Using the phase offsets measured across different subcarriers, each cooperating transmitter can accurately estimate the arrival time of the sync header. Based on the estimated

RTT between each transmitter and the leader, and the switching time from Rx mode to Tx, each transmitter synchronizes in time with the leader. Finally, considering the propagation delay of the transmitters to the receiver, each transmitter selects a transmission time so as to synchronize the arrival of all the transmissions at the receiver.

4.2.2 Nullification of FJ Signals

Assuming that the distributed FJ nodes have been synchronized, the amplitudes/phases of their signals must be adjusted to cancel out at the legitimate receivers. Consider M legitimate receivers and N FJ nodes. The channel is characterized by an $M \times N$ channel matrix $\mathbf{H} = [h_{ij}]$, where h_{ij} is the channel coefficient between receiver i and transmitter j. By setting $N = M + 1$, we can guarantee a nonempty null space for the channel matrix $\mathbf{H}$ [23]. Let $\mathbf{y}$ be an M-by-1 vector that represents only the received FJ signals at the M receivers, let $\mathbf{x}$ be an N-by-1 vector that represents the transmitted signals from the N FJ antennas, and let $\mathbf{F}$ represent the N-by-1 precoding vector (precoder) of the FJ signal.

At any time instant (time index is dropped for simplicity) and ignoring the effect of noise, we have:

$$\mathbf{y} = \mathbf{H}\mathbf{x} = \mathbf{H}\mathbf{F}m \tag{4.1}$$

where m denotes a random complex scalar at the current time and $||m||^2 = 1$. The Singular Value Decomposition (SVD) of $\mathbf{H}$ can be obtained as

$$\mathbf{H} = \mathbf{U_{M\times M}}\mathbf{\Sigma_{M\times N}}\mathbf{V}^{\dagger}_{\mathbf{N\times N}}. \tag{4.2}$$

Thus, $\mathbf{y}$ can be expressed as:

$$\mathbf{y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger}\mathbf{x}. \tag{4.3}$$

If the jamming precoder $\mathbf{F}$ lies in the null space of $\mathbf{H}$, then $\mathbf{y} = \mathbf{H}\mathbf{x} = \mathbf{0}$. In our design, we select $\mathbf{F}$ as the rightmost column of the matrix $\mathbf{V}$, i.e., the kernel of $\mathbf{H}$. For a given total budget on the jamming power and a given m, the precoder $\mathbf{F}$ determines the phase of each of the FJ signals and implies some dependencies between their jamming powers so that they add up destructively at the legitimate receivers. Let $P_j = ||\mathbf{x}_j||^2 = ||\mathbf{F}_j||^2$ be the jamming power of the jth FJ device. We explicitly derive these dependencies by solving $\sum_{j=1}^{N} h_{ij}x_j = 0$, $\forall i = 1, \ldots, M$. It turns out that each jamming power must be a linear function of P_1 as follows:

$$P_j = \omega_j P_1, \forall j = 2, \ldots, N \tag{4.4}$$

where ω_j is the scalar ratio between P_j and P_1.

4.3 PHY-layer Security for Multi-link Networks

4.3.1 Network Model and Problem Formulation

We consider a static multi-link network, consisting of an arbitrary number of legitimate nodes, each equipped with an omni-directional antenna. These nodes form a set of links $\mathcal{L}$. Each link l consists of a source (Alice) and a destination (Bob). This general multi-link network model accommodates both P2P and multihop scenarios. For the P2P scenario, $\mathcal{L}$ consists of several independent single-hop links, connecting different Alice-Bob pairs. In the multihop case, $\mathcal{L}$ contains specific links that form several paths between various pairs of nodes. Along with the set $\mathcal{L}$, there is a finite set $\mathcal{E}$ of eavesdropping locations and a set $\mathcal{J}$ of FJ nodes. We adopt a 2-D discrete model for the points in $\mathcal{E}$ [49, 50]. The probability that an eavesdropper is in location $e \in \mathcal{E}$ is denoted by p_e. Even though this model assumes some (probabilistic) knowledge of the eavesdroppers' locations, it can represent numerous scenarios by adjusting the number of locations and the probabilities assigned to them.

The number of FJ devices can be less than or greater than $|\mathcal{E}|$. In here, however, we only consider the case when $|\mathcal{E}| > |\mathcal{J}|$, since the solution for the other cases is trivial: Assign a FJ node to each of the possible eavesdropping locations. In contrast to previous research [49, 50], we assume that there can be more than one eavesdropping location per active link, i.e., $\sum_{e \in \mathcal{E}} p_e$ can be greater than one.

We formulate an optimal placement and power allocation problem for the FJ devices such that the *average* SINR at each location e is less than a threshold π. To nullify FJ interference, we consider the case of cooperative FJ whereby FJ devices cooperatively nullify their jamming signals at all $|\mathcal{L}|$ legitimate receivers, even if only a subset of these receivers are active. We employ the SourceSync protocol [38] to synchronize the FJ devices. SourceSync was initially designed to exploit sender diversity by synchronously transmitting the same packet from multiple senders. However, in our design, we leverage it to sync the FJ nodes. The leader will be an active data transmitter (Alice), who sends a sync header together with a random m before its main transmission. The power of the sync-header's transmission must be adjusted to reach all FJ nodes. Following the receipt of this header, FJ nodes calculate and adjust their transmission times to create a null region around all $|\mathcal{L}|$ receivers.

Henceforth, when we say Alice and Bob we mean the transmitter and respective receiver of a specific link $l \in \mathcal{L}$, respectively. Because the FJ signals are nullified at Bob, the transmission power of Alice is only a function of the SINR threshold at Bob and the length of link l, denoted by d_l (assuming a pathloss channel model). Therefore, to maintain the SINR at Bob $\geq$ some threshold β, the minimum transmission power at Alice of link l, denoted by $P_{t,l}$, will be:

$$P_{t,l} = \frac{N_o \beta}{d_l^{-\alpha}}. \tag{4.5}$$

where N_o is AWGN power and α is the pathloss exponent.

For the case of cooperative jamming, the SINR at Bob (SINR_b) is given by:

$$\text{SINR}_b = \frac{P_{t,l} d_l^{-\alpha}}{N_o}. \tag{4.6}$$

The optimization problem can now be stated as follows:

$$\begin{aligned} \textbf{P1:} \quad & \underset{\{x_j, y_j, P_j \, \forall j \in \mathcal{J}\}}{\text{minimize}} \quad && \sum_{j \in \mathcal{J}} P_j \\ & \text{subject to} && \textbf{C1:} \; p_e \text{SINR}_e \leq \pi, \forall e \in \mathcal{E}, \forall l \in \mathcal{L} \\ & && \textbf{C2:} \; \sum_{j=1}^{|\mathcal{J}|} h_{ij} x_j = 0, i = 1, \ldots, |\mathcal{L}| \end{aligned} \tag{4.7}$$

where (x_j, y_j) are the Cartesian coordinates of FJ node j. Constraints **C1** and **C2** represent the secrecy and nullification constraints, respectively. When link l is active, the SINR at eavesdropper e (SINR_e) is given by:

$$\text{SINR}_e = \frac{P_{t,l} d_{ae,l}^{-\alpha}}{N_o + \sum_{j \in \mathcal{J}} P_j d_{je}^{-\alpha}} \tag{4.8}$$

where $d_{ae,l}$ is the distance between Alice (of link l) and eavesdropping location e, and d_{je} is the distance between the FJ node j and eavesdropper e. Note that j, e, and hence d_{je} are not associated with a specific link l.

We propose two schemes based on formulation **P1**: per-link and network-wide schemes. For the per-link scheme, the problem is solved independently for each link. In this case, we have $|\mathcal{L}|$ independent problems. For each of these problems, the secrecy and nullification constraints are considered only for a specific link l, i.e., $|\mathcal{L}| = 1$. To ensure a nonempty nullspace for the channel matrix $\mathbf{H}$, $|\mathcal{J}|$ has to be greater than $|\mathcal{L}|$. This implies that in the per-link scheme, $\mathcal{J}$ in each problem must contain a minimum of two FJ nodes. Hence, we need at least $2|\mathcal{L}|$ FJ nodes to secure all links. For the network-wide scheme, all links and FJ devices are simultaneously considered in the secrecy and nullification constraints, i.e., we jointly optimize over all links in the set $\mathcal{L}$. One advantage of this scheme is that we only need $|\mathcal{L}|+1$ FJ nodes to ensure that the jamming signals are nullified at all $|\mathcal{L}|$ legitimate receivers.

Considering the network-wide scheme and assuming that the locations of legitimate nodes and $P_{t,l}$ $\forall l \in \mathcal{L}$ are known, the first constraint can be simplified to:

$$\textbf{C1:}\ p_e \frac{\max_{l\in\mathcal{L}} P_{t,l} d_{ae,l}^{-\alpha}}{N_o + \sum_{j\in\mathcal{J}} P_j d_{je}^{-\alpha}} \leq \pi, \forall e \in \mathcal{E}. \tag{4.9}$$

4.3.2 Solution Based on Condensation Techniques

A function f is said to be *monomial* if $f(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} x_i^{a_i}$, where $a_1, a_2, \cdots, a_n \in \mathbb{R}$. A function is said to be *posynomial* if it is a linear combination of monomials. *Signomial* programming is a class of non-convex, non-linear, and NP-hard optimization problems in which the posynomials in the constraints may be lower bounded by monomials [52].

Proposition 4.3.1. *Problem **P1** is a signomial programming problem with $|\mathcal{E}|+|\mathcal{L}|$ signomial constraints.*

Proof. The objective function is a summation of linear variables (i.e., $\sum_{j\in\mathcal{J}} P_j$). As for the secrecy constraint, we have $\forall e \in \mathcal{E}$:

$$p_e \text{SINR}_e \leq \pi, \tag{4.10}$$

$$\frac{p_e P_{t,l} d_{ae,l}^{-\alpha}}{\pi} \overset{(4.8)}{\leq} N_o + \frac{P_1}{d_{1e}^{\alpha}} + \frac{P_2}{d_{2e}^{\alpha}} + \cdots + \frac{P_{|\mathcal{J}|}}{d_{|\mathcal{J}|e}^{\alpha}}, \tag{4.11}$$

$$\frac{\left(\frac{p_e P_{t,l} d_{ae,l}^{-\alpha}}{\pi}\right) \prod_{j\in\mathcal{J}} d_{je}^{\alpha}}{N_o(\prod_{j\in\mathcal{J}} d_{je}^{\alpha}) + \sum_{j\in\mathcal{J}} P_j \prod_{\substack{i\in\mathcal{J}\\ i\neq j}} d_{ie}^{\alpha}} \leq 1 \tag{4.12}$$

which is in the form of:

$$\frac{Q(\mathbf{x})}{P(\mathbf{x})} \leq 1 \tag{4.13}$$

where $Q(\mathbf{x})$ and $P(\mathbf{x})$ are monomial and posynomial, respectively.

The same analysis can be applied to the nullification constraint to show that it also represents $|\mathcal{L}|$ signomial constraints. It follows that our formulation belongs to the category of signomial programming [52, 53]. ■

Signomial problems cannot be transformed to convex problems. However, by using *condensation techniques* [54], we can approximate any multi-term posynomial $P(\mathbf{x})$ by a monomial and transform **P1** into the standard geometric programming form (i.e., $Q(\mathbf{x}) \leq \widetilde{P}(\mathbf{x})$, where $\widetilde{P}(\mathbf{x})$ is the approximated monomial of the posynomial $P(\mathbf{x})$). It can be shown that optimal solution of the condensed problem is a feasible (but not necessarily optimal) solution for **P1** and so upper-bounds its optimal solution.

The original problem can then be heuristically solved by *iteratively* updating the parameters of the condensed problem. For each iteration, we use the optimal solution of the previous condensed problem to update the approximation parameters, and so on until we converge to the optimal solution of **P1**. Because the problem is non-convex, the algorithm may get stuck in a local minima, a case that is left for future work.

4.4 Secrecy-aware Routing Problem

Considering the per-link scheme to jointly optimize the transmission powers and locations of the FJ devices in the multihop scenario, we propose to use the total FJ power needed to secure each link as the link weight. Thus, for link $l \in \mathcal{L}$, its weight is:

$$w(l) = \sum_{j \in \mathcal{J}} P_j. \tag{4.14}$$

By securing each hop, end-to-end secrecy is achieved [47, 48]. At the same time, the quality of service is ensured by having the SINR at each end of a link lower-bounded by β. To find a secure path $\mathcal{P}$ with minimum total FJ power for a given source and destination, we run the shortest path algorithm with respect to the metric w. The cost of the resultant path $c(\mathcal{P})$ is calculated as follows: $c(\mathcal{P}) = \sum_{l \in \mathcal{P}} w(l)$.

4.5 Simulation Results

In this section, we provide the simulation results of the per-link optimization strategy (both P2P and multihop). We also compare the per-link P2P to the network-wide P2P scenarios. We set $\alpha = 2$ and $p_e = 0.5$ for each eavesdropping location, $\beta = 1$ and $\pi = 1$. The number of condensation iterations is set to 100. All power values are normalized with respect to N_o. FJ nodes are initially collocated with the data transmitters, but gradually re-positioned depending on the outcome of the optimization problem. The simulations are performed in MATLAB using the CVX package.

4.5.1 Per-link P2P Scenarios

we first study the performance of our proposed per-link scheme for P2P scenarios in terms of power consumption and interference at legitimate receivers. The network consists of 1 to 5 unidirectional links (see Figure 4.2). The set of eavesdropping locations is indicated by the diamonds in Figure 4.2. The network is deployed on a grid of dimensions $(2|\mathcal{L}|+1) \times 5$ (e.g., when $|\mathcal{L}| = 1$, we simulate a grid of dimension 3×5, with four potential eavesdropping locations).

Power Consumption

We compare our proposed per-link P2P scheme with the following schemes in which the locations of the FJ devices are fixed:

- *Tx-based FJ:* The FJ nodes are collocated with the transmitter, which could be a MIMO transmitter with some antennas dedicated to the FJ.

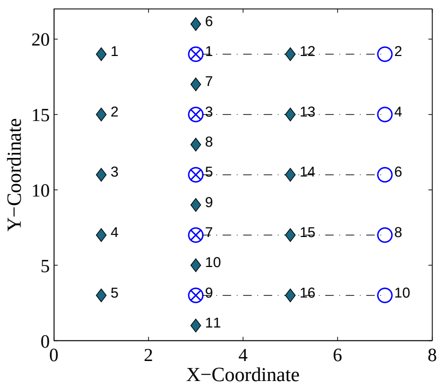


Figure 4.2: Network topology for the case of five P2P links. Hollow circles, crossed circles, and diamonds represent receivers, transmitters, and potential eavesdropping locations.

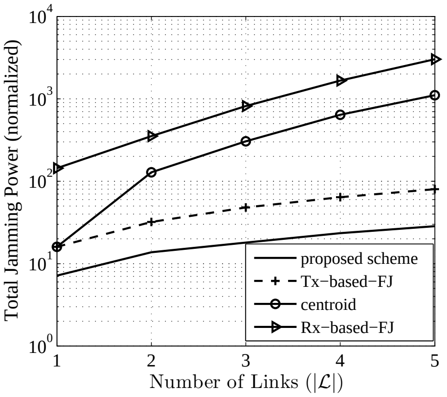


Figure 4.3: Total jamming power vs. number of links for the P2P scenario (per-link).

- *Rx-based FJ:* The FJ nodes are placed at full-duplex receivers with perfect self-interference cancellation.

- *Centroid:* The FJ nodes are placed at the centroid of all potential eavesdropping locations.

Collectively, we refer to the above three schemes by *fixed-placement* schemes. As shown in Figure 4.3, our proposed per-link P2P scheme outperforms the fixed-placement schemes achieving 55%–99% reduction in power consumption. The centroid scheme is the worst in terms of power consumption because FJ nodes are located far away from data transmitters and need to increase their powers to cover eavesdropping points around data transmitters.

To illustrate the outcome of the proposed per-link P2P case, we consider a network of one link (i.e., $|\mathcal{L}| = 1$), as shown in Figure 4.4. Numbers by each node in this figure represent the amount of interference caused by the FJ nodes on that node. In Figure 4.5, we show the change in the jamming power for each of the FJ nodes in Figure 4.4 along with their total jamming power as a function of the opti-

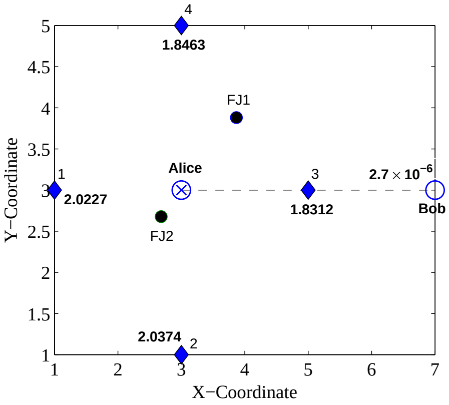


Figure 4.4: Outcome of the proposed per-link scheme for the one-link case. The hollow, crossed, and solid circles represent the locations of Bob, Alice, and the FJ nodes, respectively.

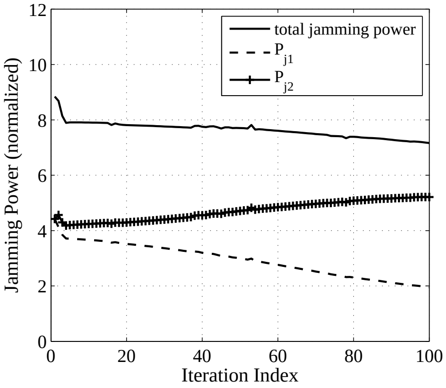


Figure 4.5: Jamming power of each FJ node along with total jamming power vs. approximation (iteration) index for the example in Figure 4.4.

mization iteration (see Section 4.3.2). The increase of P_{j2} may look counterintuitive because FJ2 is moving towards Eve1 and Eve2. FJ1 moves closer to Bob as it moves towards Eve3 and Eve4 to reduce P_{j1} required to suppress them. According to (4.4), $P_{j2} = P_{j1}(h_{11}/h_{12})^2$; so as FJ1 moves closer to Bob, P_{j2} increases quadratically with h_{11}/h_{12}. This prevents FJ2 from moving closer to Eve1 and Eve2 because the goal is to minimize the total jamming power.

In Figure 4.6, we vary $|\mathcal{E}|$ for the two-link P2P case and study the performance of our proposed per-link scheme in terms of power consumption. Our proposed scheme is shown to reduce power consumption by 54%–96% compared to fixed-placement schemes. It can be noted that the jamming power for the Tx-based FJ scheme does not scale with the number of eavesdroppers. This is because FJ nodes are located very close to data transmitters, thus jamming power will be a function of the transmit power (in **P1** substitute $d_{je} = d_{ae_l}, \forall j \in \mathcal{J}$). Note that the transmit power $P_{t,l}$ does not scale with the number of the eavesdroppers.

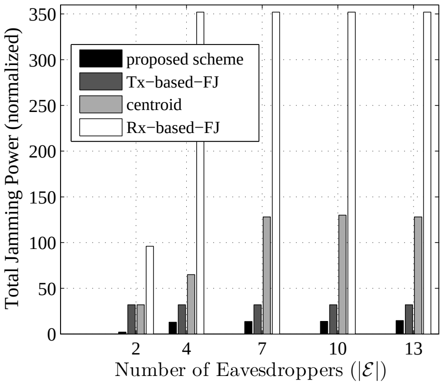


Figure 4.6: Power consumption vs. number of potential eavesdropping locations for the case of two P2P links (per-link).

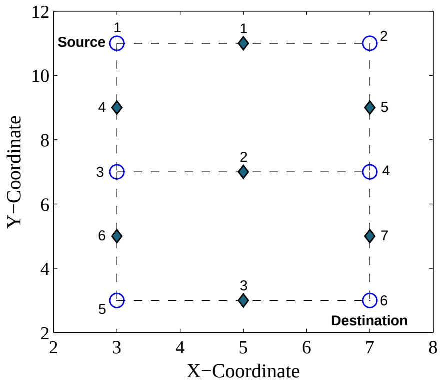


Figure 4.7: Network topology for routing simulations. Hollow circles represent the legitimate nodes and diamonds represent potential eavesdropping locations.

Interference at Legitimate Receivers

Because FJ nullification is incorporated in our formulation, the SINR at the receiver of any link should not be less than β. Considering the example in Figure 4.4, the SINR at Bob in the proposed per-link scheme is maintained at 1 dB with received jamming power of 2.7×10^{-6}. For the Tx-based and centroid schemes, however, FJ is hardly nullified and the SINR goes down to 0.5 dB, which means that Bob is unable to decode Alice's messages.

4.5.2 Per-link Multihop (Routing) Scenarios

We simulate a multihop network consisting of three interconnected and bidirectional links, as shown in Figure 4.7. We calculate the minimum energy route and its associated jamming power for a packet transmitted from node 1 to node 6 along a multipath route. We also calculate $\mathbb{E}[c(\mathcal{P})]$ for all possible paths $\mathcal{P}$ (i.e., all possible Tx-Rx pairs). A summary of the results is shown in Table 4.1.

Table 4.1: Comparison of the proposed per-link scheme and the fixed-placement schemes in terms of the cost of the minimum-energy path ($|\mathcal{L}| = 2$).

	Proposed	Tx-based FJ	Centroid	Rx-based FJ
Best path	1-2-4-6	1-2-4-6	1-3-4-6	1-2-4-6
Cost	7	48	128	304
$\mathbb{E}[c(\mathcal{P})]$	5.5	26.7	75.5	167.5

4.5.3 Network-wide Scenarios

To study the energy efficiency of our network-wide optimization strategy, we compare our proposed per-link P2P and network-wide P2P schemes in terms of the total jamming power required to cover the whole network when $|\mathcal{L}| = 2$. As shown in Figure 4.8, the network-wide scheme reduces power consumption by 14%–38% relative to the per-link scheme. Note also that the network-wide scheme allows for simultaneous operations of different links because it ensures the nullification of the FJ signals at all legitimate receivers.

4.6 Summary

In this paper, we exploited friendly jamming for PHY-layer security in small-scale multi-link wireless networks in the presence of eavesdroppers. We jointly optimized the powers and locations of the friendly jamming nodes so as to minimize the total jamming power required to secure legitimate transmissions. Distributed MIMO techniques are used to nullify the friendly jamming signals at legitimate receivers. A signomial programming problem was formulated and approximated as a convex geometric programming problem using condensation techniques. We then proposed two optimization strategies: per-link and network-wide (all links jointly). It was shown that our per-link scheme outperforms previous schemes in terms of energy efficiency (55–99 percent power saving). Moreover, the network-wide optimization was shown to be more energy-efficient than per-link scheme (14–38 percent addi-

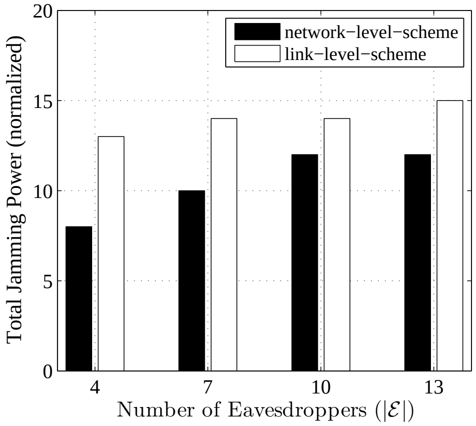


Figure 4.8: Power consumption of the network-wide and per-link schemes vs. the number of potential eavesdropping locations for the case of two P2P links.

tional power saving) and also requires about half the number of friendly jamming nodes than per-link optimization. For multihop scenarios, we proposed a routing metric that finds a secure path that requires minimal jamming power.

CHAPTER 5

Friendly CryptoJam

5.1 Introduction

Albeit using friendly jamming results in degrading Eve's channel (see Chapter 4), it does not completely immune wireless communications to eavesdropping. Eve can perform low-level RF analysis to obtain SCI even when PHY/MAC headers are encrypted or cannot be successfully decoded, a threat that has not been well-studied in the literature. Consider, for example, the detection of the payload's modulation scheme of an entirely encrypted PHY frame. Using an off-the-shelf device such as a signal analyzer or a dedicated device equipped with an FPGA [55], one can detect the modulation scheme, and accordingly estimate the payload's data rate. The same device can also measure the frame duration and determine the packet size based on the estimated data rate.

5.1.1 Existing Countermeasures for Hiding SCI

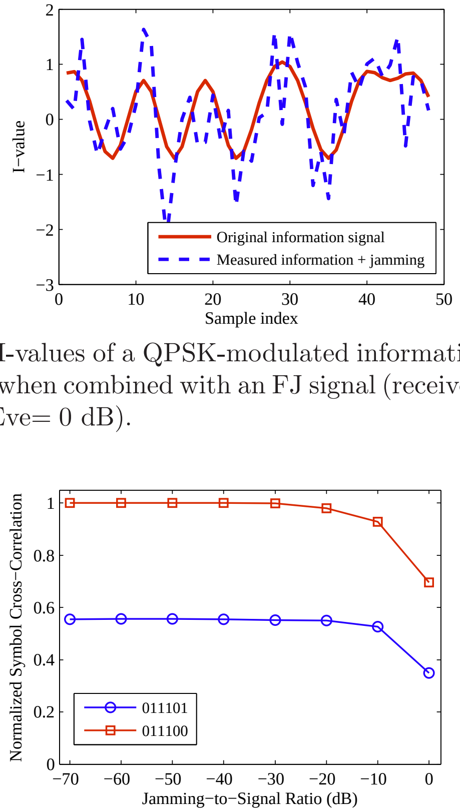
Before describing the limitations of various techniques that have been proposed to prevent Eve from obtaining SCI or intercepting PHY/MAC headers, we first explain why a naive approach based on encrypting such headers is not practical. To decrypt such headers, the intended receiver (Bob) needs to identify the sender at PHY layer among several potential senders and apply the right decryption key. When headers are fully encrypted, none of their fields (e.g., sender's MAC address) can be used for identification. Similarly, MAC address randomization that has recently been employed for hiding the true address in probe requests (e.g., in Apple iOS 8.1.3) is also not sufficiently helpful. Besides its other vulnerabilities [56], such hidden MAC identifier still cannot be used for decryption at the PHY layer. Likewise, Alice-Bob channel or Alice's radiometric features cannot be used as an identifier because of mobility and/or inaccuracy of low-end RF receivers [57].

Techniques to prevent SCI leakage can be divided into three categories: SCI obfuscation at upper layers, rate hiding techniques [58, 59], and eavesdropper deafening at the PHY layer. Upper-layer SCI obfuscation techniques aim at invalidating SCI, usually at the cost of traffic overhead. At the application layer, packet padding can be used to alter the traffic statistics. However, Dyer *et al.* [15] demonstrated that even if packet lengths are obfuscated, training a website-traffic classifier based only on the total bandwidth can result in a very high classification accuracy. They also proposed a countermeasure that obfuscates the total bandwidth, but with $100\%-400\%$ overhead. *Traffic reshaping* [60] is a MAC layer technique that involves configuring several virtual interfaces with different MAC addresses for the same device and dynamically distributing the traffic among them so as to create different traffic patterns on each MAC interface. This prevents Eve from associating all the packets with the same sender. Similarly, the sender and receiver can agree on a set of confidential time-rolling MAC addresses (e.g., [6, 7]. However, these identifier concealment techniques cannot hide certain attributes, including the modulation scheme and transmission rate.

More recently, trellis structure has been employed for providing security [58, 61–63]. To hide the payload's modulation scheme, Conceal and boost modulation (CBM) was proposed in [58], whereby convolutional codes based on a Generalization of Trellis Coded Modulation (GTCM) are used, combined with a cryptographic interleaving mechanism. In order to eliminate the dependency among successive symbols, the authors proposed securely interleaving blocks of p symbols, where p is a prime number. Large p is required to prevent exhaustive search and known-plaintext attacks on the interleaved blocks. GTCM directly encodes the symbols of any modulation scheme into the highest-order modulation scheme. A symmetric-key scheme was also proposed to encrypt the PHY-layer header. While CBM can achieve up to 8 dB asymptotic coding gain (in idealized simulation scenarios), it does not address the issue of sender identification and the decryption of the PHY-layer header. Moreover, the complexity of GTCM codes, interleaving, and expensive symmetric-key encryption result in a large decoding delay at Bob. CBM also suffers significantly from inaccurate frequency offset (FO) estimation ($>$ 4 dB loss) due to acute susceptibility of higher-order modulation schemes to phase offset.

PHY-layer eavesdropper deafening techniques include FJ, e.g., [64–66]. In this method, Eve's channel is degraded without impacting the channel quality at Bob. This is done using MIMO techniques or by having relay nodes transmit a jamming signal that is harmless to Bob. However, three fundamental issues limit the practicality of this approach. First, if Eve is equipped with multiple antennas, she can cancel out a transmitter-based FJ signal [67,68]. For example, Schulz *et al.* [68] exploited a known part of Alice's signal (e.g., frame preamble) to estimate the precoding matrix used in generating the FJ signal and then eliminate it from the received signal at Eve. This matrix is supposed to be secret and unique, as it depends on the channel state information (CSI) for the Alice-Bob channel, i.e., it represents a signature of the Alice-Bob channel. This known-plaintext attack can thwart any security scheme that relies on data prefiltering (precoding) at Alice. Other techniques such as beamforming and orthogonal blinding (e.g., [64]) are essentially based on precoding, and so are insufficient. Furthermore, the uniqueness of the Alice-Bob CSI has been shown not to be true in the presence of strong LOS component [69]. Specifically, a few adversaries located several ($\sim$ 18) wavelengths away from Bob can cooperatively reconstruct the signature for the Alice-Bob channel.

Second, FJ requires additional transmission power and antenna(s), which come at the expense of throughput reduction for the information signal. The jamming power may need to be even higher than the information signal power to achieve



(a) I-values of a QPSK-modulated information signal when combined with an FJ signal (received JSR at Eve= 0 dB).

(b) Cross-correlation between received (information + FJ) signal and one of two possible values for the information signal vs. JSR (011100 is the correct value).

Figure 5.1: Cross-correlation attack on a semi-static QPSK-modulated signal that is superposed on an FJ signal.

non-zero secrecy capacity [65]. Moreover, Alice may not have sufficient number of antennas (degrees of freedom) to apply friendly jamming.

Third, transmitter- and receiver-based FJ (e.g., [66]) are still vulnerable to *cross-correlation attacks* on (unencrypted) semi-static header fields. In these attacks, a header field can take one of a few possible values. Eve can detect the start of a frame, even if it is combined with a jamming signal [70]. By knowing the underlying header format (i.e., where each field is supposed to start), Eve can locate the start

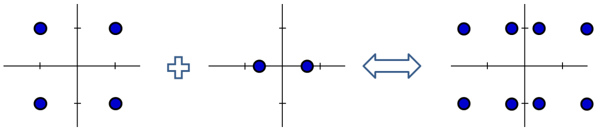


Figure 5.2: Combining QPSK-modulated and BPSK-modulated signals with different powers results in a 8-symbol constellation map.

time of the targeted field in the header. Figure 5.1(a) shows an example of the received in-phase (I) values of a sequence of modulated symbols containing a semi-static header field plus an FJ signal. Even though Eve may not be able to decode the received signal, she can correlate the modulated symbol of each possible value of this field with the received signal and guess the true field value. In general, this cross-correlation attack can be formulated as a composite hypothesis testing. We show a simple example in Figure 5.1(b), which depicts the cross-correlation between the combined received signal and one of two possible field values (011100 is the true value and 011101 is another possible value). The cross-correlation is shown as a function of the jamming-to-signal ratio (JSR). Each point is the average of 100 simulation trials. The plot shows that Eve can successfully determine the true value even when the FJ power is as high as the information signal power.

Furthermore, FJ cannot effectively hide the modulation scheme and frame duration. If the jamming signal is random, Eve can employ detection techniques for low SNR (e.g., [55, 71]) to detect the modulation scheme. Even if the FJ signal takes the form of a digitally modulated signal (as opposed to random noise), Eve may still detect the modulation scheme of the payload by analyzing the order and constellation map of the received superposition. The superposition of the I and Q components of the complex symbols that belong to the two signals results in a modulation scheme whose order and constellation depend on the original schemes and the respective received powers. For example, the constellation map resulting from the superposition of two signals, one modulated with QPSK and the other with BPSK, can disclose the constituent modulation schemes (see Figure 5.2).

5.1.2 Overview of Friendly CryptoJam

To address the aforementioned limitations, we propose *Friendly CryptoJam* (FCJ), a form of friendly jamming but with the information and jamming signals intermixed right after the digital modulation phase and before the frame is transmitted over the air. Our intermixing method makes FCJ a form of modulation-level encryption (for the whole frame) and also a form of modulation obfuscation (for the PHY-layer payload). To generate a secret FJ sequence, Alice exploits an unpredictable sender identifier as a seed, which is then embedded in the frame preamble (i.e., a PHY-layer identifier). This way, Bob can identify the sender for key lookup and synchronize with Alice in generating the same FJ sequence. Hereafter, we call this secret sequence as "FJ traffic". This identifier is independent of the link features and is robust to known plaintext attacks. The proposed modulation encryption preserves the Gray coding structure of the encrypted symbols on the original constellation map. In contrast to conventional (digital domain) encryption, the encryption in FCJ is modulation-aware.

Using parts of the same FJ traffic, encrypted symbols of the payload are then coded and mapped (upgraded) to the constellation map of the highest-order (target) modulation scheme supported by the system. Modulation upgrade is done in a way that prevents the original modulation scheme of the payload from being disclosed, i.e., it provides *indistinguishable modulation unification*. In contrast to the uncoded modulation unification in the initial design [59] and variable-rate coding for upgrading different modulation schemes to same target modulation scheme in CBM [58], the proposed mapping employs only two minimal trellis-coded modulation (TCM) codes with constraint length ≤ 2 (and constant rate for the same target modulation scheme). These codes are combined with the FJ traffic so as to obfuscate the original payload's modulation scheme through continuously moving the coded symbols on the constellation map while maintaining BER performance. Compared to [58], such mapping enjoys lower complexity, decoding delay, and susceptibility to FO estimation errors. We further provide an analytical study of the impact of these errors. In addition, FCJ hides the modulation scheme and the TCM code without symbol interleaving. In contrast to classic FJ techniques, a single antenna is sufficient to transmit both the information and FJ signals.

One important challenge in designing FCJ is how to modify the FJ traffic on a per-frame basis. Not changing the FJ traffic during a session opens the door for a dictionary attack against semi-static header fields. Furthermore, relying on a single (long) secret sequence for the FJ traffic makes the design prone to synchronization errors. To ensure consistency in the generation of FJ traffic at Alice and Bob, Alice conveys a frame-specific seed (e.g., frame and sender ID) whose modulated value is superposed onto the known frame preamble. Together with the session key, this seed is fed into an appropriately selected pseudo-random number generator (PRNG) to generate the secret FJ traffic. The seed is also used for sender identification at PHY layer. Superimposing the seed with the preamble, however, may degrade the preamble's crucial functions (including frame detection). To prevent that, we exploit the low cross-correlation property of cyclically rotated Barker sequences to construct a seed-bearing signal in 802.11b systems.

FCJ complements conventional data encryption and upper-layer traffic manipulation techniques by providing protection for the entire frame. The combination of modulation encryption and indistinguishable modulation unification prevents any SCI-based classification. It guards against any attack that is based on the payload's modulation scheme or unencrypted header fields. A high-level example of FCJ is given in Figure 5.3.

5.2 Background – PHY-layer Attributes

(1) PHY-layer header fields. Many standards, including 802.11 variants, specify the frame length and payload's transmission rate in the PHY header. The transmission rate is typically adjusted based on channel conditions, resulting in different

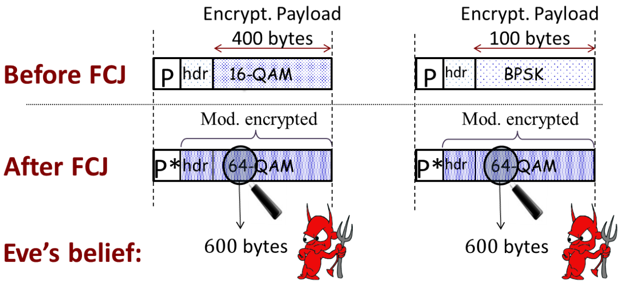


Figure 5.3: Example of using *Friendly CryptoJam* to hide the header fields and the modulation scheme (and payload size) of two frames. Headers and payload are modulated-encrypted and modulation-unified without changing the information rate and frame duration. Under FCJ, a seed ($\mathcal{ID}$) is overlaid on the original frame preamble (P), leading to a new preamble ($P^* = P + \mathcal{ID}$).

frame durations (in seconds) for the same payload. For example, in 802.11b/g, the data rate and the modulation scheme are specified in the 'Signal' and 'Service' fields, respectively. In 802.11a, the 'rate' field represents both the transmission rate and the modulation scheme (BPSK, QPSK, 16-QAM, or 64-QAM). The 'Modulation and Coding Scheme' field in 802.11n is similar to the rate field in 802.11a. All 802.11 variants specify a 'length' field, which represents the payload size in octets (for 11a/n) or in milliseconds (for 11b).

(2) Detection of lower-layer fields. Typically, the preamble and the PHY header are transmitted at the lowest supported rate.[1] Transmission rate for the frame payload (including MAC header) is adjusted adaptively. The 802.11i security amendment provides integrity only for the MAC header. The preamble, PHY, and MAC headers are all transmitted in the clear, allowing an adversary to intercept them and obtain the transmission rate information for the payload. This rate may also be determined by detecting the payload's modulation scheme and combining that with the frame length to compute the packet size. A modulation scheme is usually associated with two or three data rates of different code rates. For example,

[1] The only exception is the short header format of 802.11b/g, which uses DQPSK.

in 802.11a, 16-QAM is used for data rates 24 and 36 Mbps. Hence, by determining the modulation scheme, it is rather easy for the adversary to guess the data rate.

5.3 System Model

We consider a wireless network in which each link consists of single-antenna transmitter (Alice) and receiver (Bob). The link operates in the presence of one or more eavesdroppers (Eve). Alice and Bob first create a shared *pairwise transient key* (PTK) through the EAPOL 4-way handshake of 802.11i. PTK is used to encrypt the payloads, but as explained later we also use it to generate FJ traffic and frame IDs at the PHY layer. Each node maintains a table of PTKs and session IDs of all known neighbors in the network[2]. To customize FCJ, They exploit knowledge of the standard preamble and frame format without introducing a new preamble or header field (i.e., wasting the throughput). This way, customizing the design to other systems with a known Barker-based preamble structure and an arbitrary but known set of modulation schemes is straightforward. Without loss of generality, we consider a rate-adaptive system that uses the preamble of 802.11b. For simplicity, when presenting FCJ, we consider BPSK, QPSK, 16-QAM, and 64-QAM modulation schemes for the payload.

Eve knows the frame structure and protocol semantics. She can be a passive eavesdropper or a reactive jammer who selectively jams upon analyzing the early part of a frame. Eve's attacks may include cross-correlation attacks (e.g., Figure1(b)), rate-adaptation attack [19], device-based user-tracking attacks, dictionary attacks, known-plaintext attack [68], key-recovery attack, and any data-rate-based traffic classification attack. We allow Eve to be equipped with multiple antennas. She can also perform RF analysis, correlation, and (statistical) modulation detec-

[2] Because the (encrypted) MAC address is decoded after the PHY header, it cannot be used to retrieve the corresponding PTK at the PHY layer. Session ID is used instead to distinguish between different neighbors.

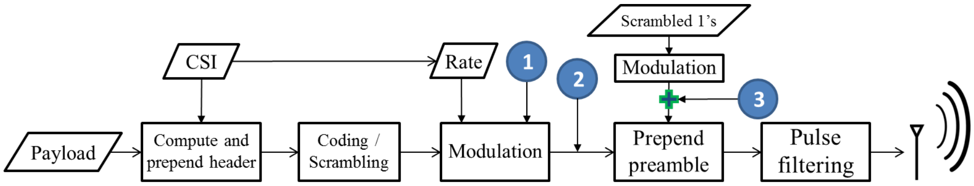


Figure 5.4: Transmission chain at Alice under FCJ. Insertion points (1), (2), and (3) refer to modulation encryption, TCM-based modulation unification, and message embedding within the preamble.

tion. Alice may employ any traffic classification mitigation technique (e.g., traffic morphing or random padding) at upper layers, but does not pad a packet to a fixed size (e.g., 'Maximum Transmission Unit'). Otherwise, if the PHY layer receives packets of the same size, the frame duration will reveal the actual modulation order.

Figure 5.4 shows a schematic view of Alice's transmit chain and the insertion points of FCJ's components, which include modulation encryption (point 1), modulation unification (point 2) and frame/session ID embedding (point 3). Starting with the MAC header, once the frame payload arrives at Alice's PHY layer, Alice computes the PHY-header fields, including the modulation scheme for the payload that is calculated based on same rate-adaptation algorithm. The preamble, PHY header, and payload are then scrambled and modulated. They are then passed to the pulse shaping filters and transmitted over the air. Bob, on the other hand, detects the preamble and extracts the frame ID embedded in it to regenerate the FJ traffic and estimates the CSI. Subsequently, Bob recovers and decrypts the header to extract the payload's modulation scheme, which is used to recover the rest of the frame.

5.4 Modulation Unification

In this section, we introduce a method for indistinguishably unifying different modulation schemes using FJ traffic. For now, we assume that the FJ sequence is already available at both Alice and Bob. Furthermore, we assume that Bob can decrypt the headers and obtain the true modulation scheme. Confidential generation and synchronization of the FJ sequence will be explained in Section 5.5 along with the modulation encryption scheme.

5.4.1 Uncoded Modulation Unification

To prevent any rate-based SCI classification, the modulation scheme used for different frame payloads should always look the same to Eve. We achieve that by embedding the payload's original modulation symbols in the constellation map of the highest-order modulation scheme supported by the underlying system (denoted by $\mathcal{M}_M$). However, when upgrading the modulation scheme, we want to preserve the same demodulation performance at Bob.

To unify various payload modulation schemes, denoted by $\mathcal{M}_i$, $i = 1, 2, \ldots, M$, each modulated symbol of Alice's payload is combined with one modulated FJ traffic, producing one point in the target constellation map. The distribution of these points in the target constellation map must be uniform, similar to the distribution of the points of a $\mathcal{M}_M$-modulated signal. As long as the generated symbols are equally probable and a given symbol is independent of the previous and next symbols (from Eve's perspective), Eve cannot determine if $\mathcal{M}_i$ has been obfuscated.

In general, a higher-order modulation scheme is more susceptible to demodulation errors. To illustrate, let the FJ traffic sequence be $\mathbf{j}$ and let $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$ be a mapping, known to both Alice and Bob, that is used to embed the symbols of $\mathcal{M}_i$, (where $\mathcal{M}_1$ and $\mathcal{M}_M$ are the lowest- and highest-order modulation schemes, respectively) into the constellation map of $\mathcal{M}_M$. The minimum Euclidean distance between the symbols in the constellation of $\mathcal{M}_i$, denoted by $d_{min,i}$, specifies the probability of a demodulation error at a given SNR value. This $d_{min,i}$ generally decreases with i. Tables 5.1 and 5.2 depicts $d_{min,i}$ for the 802.11a system after taking into account a modulation-dependent normalization factor K_{MOD} [1]. K_{MOD} is a coefficient that is multiplied by the (I,Q) values to achieve the same average symbol-power across different modulation schemes. To maintain the same $d_{min,i}$ after upgrading $\mathcal{M}_i$ to $\mathcal{M}_M$, i.e., achieve the same BER, any two neighboring points in the constellation of $\mathcal{M}_i$ should be mapped to two points in $\mathcal{M}_M$ whose distance is no smaller than their distance in $\mathcal{M}_i$. At the same time, all the resulting constellation points of $\mathcal{M}_M$ as observed by Eve must be equally probable (as when a random information sequence is modulated using $\mathcal{M}_M$). Otherwise, Eve may guess $\mathcal{M}_i$ by performing statistical analysis. In the following, we explain an uncoded mapping $\mathcal{F}_{\mathbf{j}}$, first proposed in [59], which fulfils both of the above design requirements. Note that modulation unification is not applied when $i = M$.

For a given $\mathcal{M}_i$, our scheme defines $\frac{|\mathcal{M}_M|}{|\mathcal{M}_i|}$ equal-size and nonoverlapping sets of constellation points in $\mathcal{M}_M$, where $|\mathcal{M}_i|$ is the number of constellation points in $\mathcal{M}_i$. The constellation points of $\mathcal{M}_i$ can be mapped to any of these sets, but the selection of a set depends on $\mathbf{j}$ and hence can be different from one symbol to another. For any symbol s that is modulated using $\mathcal{M}_i$, Alice needs $(\log_2 |\mathcal{M}_M| - \log_2 |\mathcal{M}_i|)$ FJ bits to select one of the $|\mathcal{M}_M|/|\mathcal{M}_i|$ sets. The value of s determines one of the points inside the selected set. So Alice picks the first $\log_2 \frac{|\mathcal{M}_M|}{|\mathcal{M}_i|}$ bits in $\mathbf{j}$ for the first symbol to be transmitted, the next $\log_2 \frac{|\mathcal{M}_M|}{|\mathcal{M}_i|}$ bits for the second symbol, and so on. The same bits in $\mathbf{j}$ always point to the same target set, i.e., $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$ is static. This ensures that the transmitted symbols are equally probable, considering that the bits in $\mathbf{j}$ are randomly distributed (so as Alice's symbols). As explained in Section 5.5, we rely on a cryptographic hash function like SHA-3 to generate $\mathbf{j}$.

Let j, $j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$, be the decimal representation of the bits in $\mathbf{j}$ that are associated with s, and let $\mathcal{U}_j = \{u_j^0, \ldots, u_j^{|\mathcal{M}_i|-1}\}$ be the corresponding target set on the constellation map of $\mathcal{M}_M$. During the decoding process, Bob knows $\mathbf{j}$ and $\mathcal{M}_i$. To obtain the original symbol s, Bob first removes all the constellation points of $\mathcal{M}_M$ except those belonging to $\mathcal{U}_j$. He then follows a standard demodulation process to determine the most likely symbol in $\mathcal{U}_j$, given the observed symbol.

i	$\mathcal{M}_i$	K_{MOD}[1]	$d_{min,i}$	$d_{min}\big(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\big)$	$\gamma_i(3)$	$\gamma_i^{(2)}(3)$	$\gamma_i^{(4)}(3)$
1	BPSK	1	2	$4/\sqrt{5}$	$4/5 \simeq$ -0.969 dB	$0.9 \simeq$ -0.46 dB	$6.8/4 \simeq$ 2.3 dB
2	QPSK	$1/\sqrt{2}$	$2/\sqrt{2}$	$4/\sqrt{10}$	$4/5 =$ -0.969 dB	$1 = 0$ dB	$1.6 \simeq$ 2.04 dB
3	16-QAM	$1/\sqrt{10}$	$2/\sqrt{10}$	$2/\sqrt{10}$	$1 = 0$ dB	N/A	N/A

Table 5.1: Parameters of the optimal mapping from BPSK and QPSK to 16-QAM.

i	$\mathcal{M}_i$	K_{MOD}[1]	$d_{min,i}$	$d_{min}\big(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\big)$	$\gamma_i(4)$	$\gamma_i^{(2)}(4)$	$\gamma_i^{(4)}(4)$
1	BPSK	1	2	$8/\sqrt{21}$	$16/21 =$ -1.181 dB	$66/84 \simeq$ -1.05 dB	$130/84 \simeq$ 1.9 dB
2	QPSK	$1/\sqrt{2}$	$2/\sqrt{2}$	$8/\sqrt{42}$	$16/21 =$ -1.181 dB	$68/84 \simeq$ -0.92 dB	$128/84 \simeq$ 1.83 dB
3	16-QAM	$1/\sqrt{10}$	$2/\sqrt{10}$	$4/\sqrt{42}$	$2/2.1 \simeq$ -0.21 dB	$5/4.2 \simeq$ 0.76 dB	$4/2.1 \simeq$ 2.8 dB
4	64-QAM	$1/\sqrt{42}$	$2/\sqrt{42}$	$2/\sqrt{42}$	$1 = 0$ dB	N/A	N/A

Table 5.2: Parameters of the optimal mapping from 802.11a modulation schemes to 64-QAM.

Next, we discuss an optimal strategy for constructing the sets $\mathcal{U}_j$ by clustering the $|\mathcal{M}_M|$ symbols in the constellation map of $\mathcal{M}_M$ into $|\mathcal{M}_i|$ equal-size disjoint partitions. In here, optimality is taken w.r.t. maximizing the minimum distance between all pairs that belong to the same set. Let $d_{min}\big(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\big)$ be the minimum distance between any two elements in $\mathcal{U}_j$ over all possible j, $j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$. For an arbitrary modulation scheme $\mathcal{M}_M$, the problem of optimizing $\mathcal{U}_j$ to maximize $d_{min}\big(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\big)$ (a max-min problem) can be easily mapped (converted) via a sign change to the *min-max clustering* problem [72], which can be solved in $\mathcal{O}(|\mathcal{M}_M|^2\,|\mathcal{M}_i|\,\log(|\mathcal{M}_M|))$ time. For the special cases of $\mathcal{M}_M$ = 8-PSK and 16-QAM, an optimal solution to this problem was obtained in [73] by applying the "mapping by set partitioning" rule, which successively partitions the sets into subsets with increasing minimum distances.

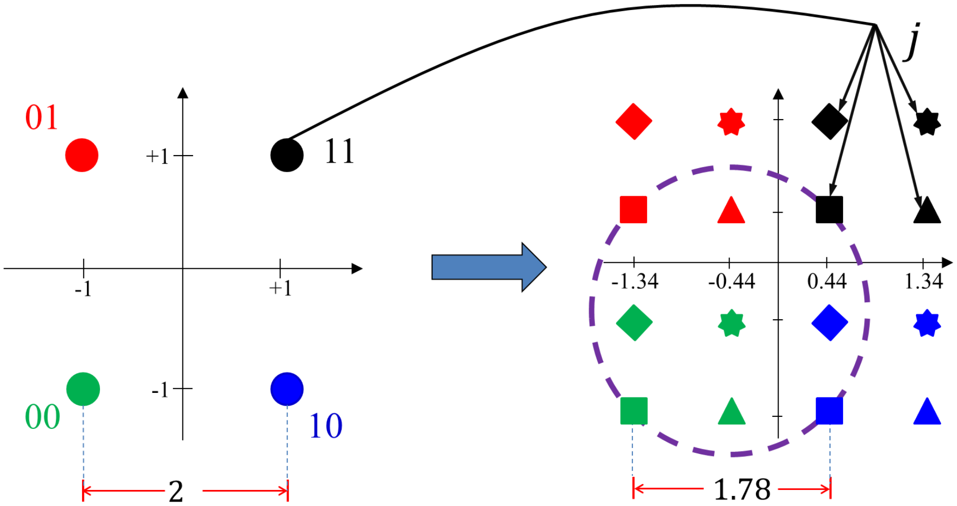


Figure 5.5: Optimal mapping from QPSK to 16-QAM. The points that belong to the same set $\mathcal{U}_j, j = 0, \ldots, 3$, are shown using the same shape. For example, the squares on the dashed circle constitute $\mathcal{U}_0$.

Figure 5.5 illustrates an optimal mapping from QPSK to 16-QAM. On the 16-QAM constellation, the points that belong to a given $\mathcal{U}_j$ are shown using the same shape. In this case, The quadrant of any given constellation point in $\mathcal{M}_M$ is specified by the user payload bits; the $\log_2 |\mathcal{M}_M|/|\mathcal{M}_i|$ FJ bits specify the symbol position within that quadrant. In Figure 5.6, we show an optimal partitioning of 64-QAM symbols into four sets, which are used to embed 16-QAM-modulated symbols. The optimality is followed by the property that the elements in every set are uniformly distributed across the constellation map and the minimum distance within each set is the same for all sets. Through successive partitioning, we can obtain the optimal partitioning for embedding BPSK and QPSK into 64-QAM. Depending on $\mathcal{M}_i$ and $\mathcal{M}_M$, the number of FJ bits required for modulation unification can be different. We will explain how Alice secretly conveys $\mathcal{M}_i$ to Bob in Section 5.5.

The above mapping may not maintain the initial $d_{min,i}$ for all $\mathcal{M}_i$'s. Let $\gamma_i(M)$

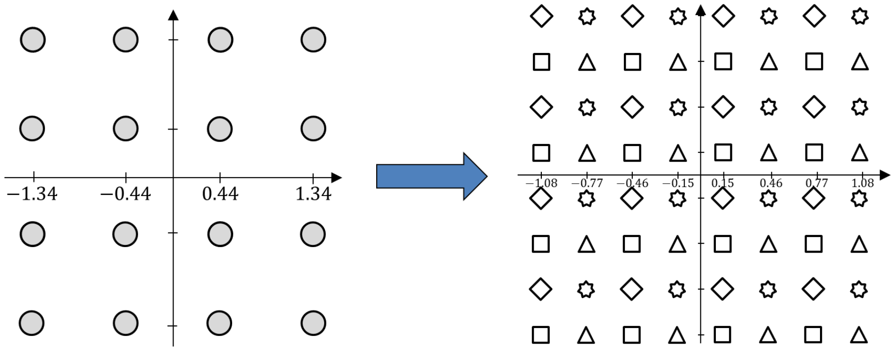


Figure 5.6: Optimal mapping from 16-QAM to 64-QAM. The points that belong to the same set $\mathcal{U}_j, j = 0, \dots, 3$, are shown using the same shape.

be the demodulation performance gain of mapping $\mathcal{M}_i$ into $\mathcal{M}_M$:

$$\gamma_i(M) = \frac{d^2_{min}\big(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\big)}{d^2_{min,i}}, i = 1, \dots, M-1. \tag{5.1}$$

For optimal mapping to 16-QAM and 64-QAM, BPSK (QPSK) will have a gain of -1.181 dB (1.181 dB loss) and -0.969 dB (0.969 dB loss), respectively, as shown in Tables 5.1 and 5.2. We compensate for this loss by applying a novel untraceable modulation coding technique.

5.4.2 Residual FO Estimation Error

Besides the SNR and $d_{min,i}$, the demodulation performance at Bob depends on how accurate he estimates Δ_f. Residual error in estimating Δ_f manifests itself as a time-varying phase error that increases linearly with the symbol index and may eventually displace a received symbol out of its expected region in the constellation map (see Chapter 2). Depending on the frame duration, the residual FO may move a received symbol to a wrong point on the constellation map, causing a demodulation error. Such error . Therefore, for the same residual FO and SNR level, longer frames experience more symbol/bit error rate towards the end of the frame.

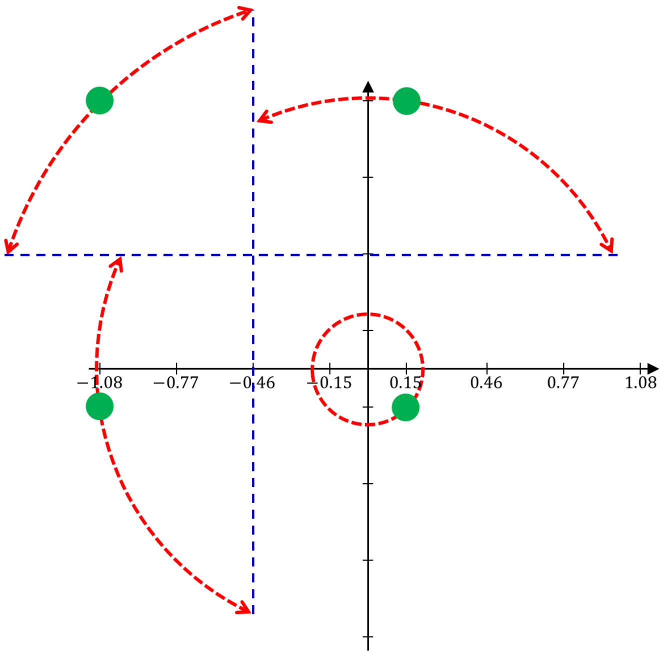
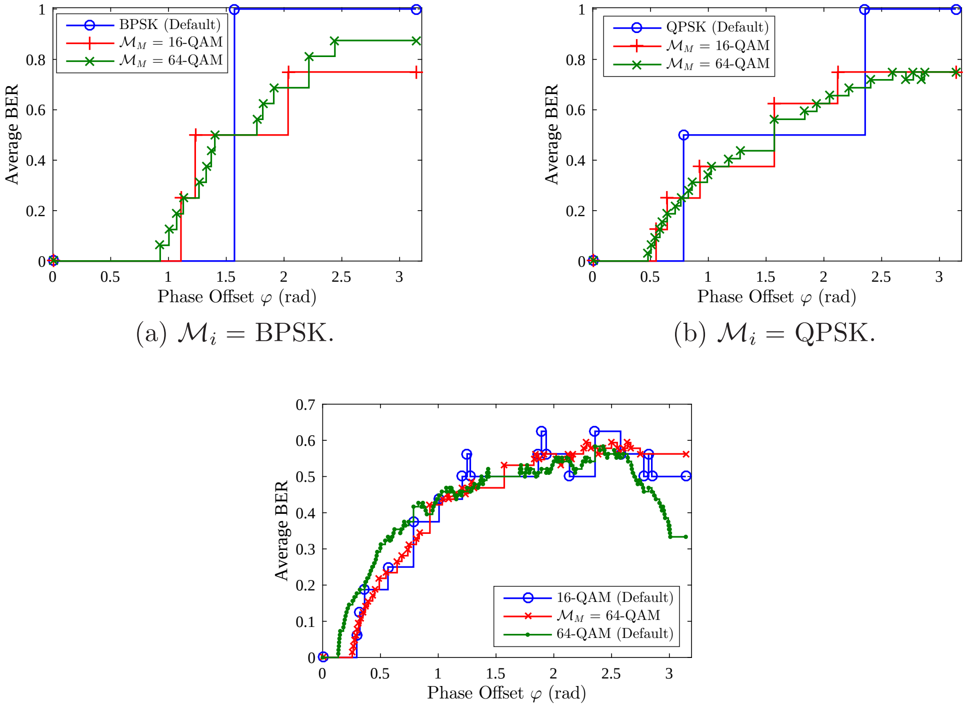


Figure 5.7: Example of uneven impact of phase offset on symbols with different amplitudes when QPSK symbols are mapped to four 64-QAM symbols.

The higher density of the symbols in higher-order modulation schemes contributes to higher susceptibility of these schemes to FO estimation errors. Although applying Gray coding in these schemes alleviates the consequences of symbols displacement on BER, higher-order modulation schemes experience demodulation errors by smaller phase offsets. Therefore, any unification scheme that maps the $\mathcal{M}_i$-modulated symbols to a denser set of symbols in $\mathcal{M}_M$ incurs a performance loss, especially if the symbols are correlated via coding and a demodulation error may propagate to the subsequent symbols (e.g., in [58, 73]). One advantage of the uncoded modulation unification in FCJ over [58] is that the target sets $\mathcal{U}_j$ have the same density as $\mathcal{M}_i$, i.e., $|\mathcal{U}_j| = |\mathcal{M}_i|$.

However, the symbols in a target set $\mathcal{U}_j$, which is used to embed in $\mathcal{M}_M$ the symbols of symmetric constellation map of $\mathcal{M}_i$, are asymmetrically distributed. Therefore, with the same phase offset φ, different symbols experience uneven dis-



(a) $\mathcal{M}_i$ = BPSK.

(b) $\mathcal{M}_i$ = QPSK.

(c) $\mathcal{M}_i$ = 16-QAM or 64-QAM.

Figure 5.8: Impact of phase offset on different $\mathcal{M}_i$ and $\mathcal{M}_M$ combinations in modulation unification.

placements. That means some of the symbols are more robust to phase offset and other symbols are more vulnerable. We illustrate this in Figure 5.7 with an example of one of the $\mathcal{U}_j$s when hiding QPSK in 64-QAM. The coordinates shown with dashed lines are the ones used for demodulating the four 64-QAM-modulated symbols at Bob. These lines are the boundaries for the optimal demodulation regions in AWGN channel. However, unequal amplitudes of the symbols results in different displacement lengths for the same phase offset φ. The dashed arcs represent the amount of displacement that can move a symbol to its neighboring region. The symbol with the smallest amplitude never leaves its expected region, while the symbol with the highest amplitude may easily leave its region when the FO estimation is not accurate.

In Figure 5.8, We compare the average BER of different $\mathcal{M}_i$s embedded in $\mathcal{M}_M =$ 16-QAM and 64-QAM to the BER of original $\mathcal{M}_i$s for different φ values. When $\mathcal{M}_i =$ BPSK or QPSK (Figure 5.8(a) and Figure 5.8(b)), applying modulation unification makes the demodulation more vulnerable to small φ values. For example, as long as $\varphi < \pi/4$, QPSK-modulated symbols will not experience any bit error. However, when these symbols are mapped to 16-QAM or 64-QAM symbols, the error-free phase offset range is shrunk to $\varphi < \pi/6$. That means more symbols in a frame will be demodulated in error. On the other hand, depending on the phase offset value, modulation unification can make the demodulation more robust to residual errors (e.g., when $\pi/4 \leq \varphi < \pi/2$).

When $\mathcal{M}_i =$ 16-QAM, modulation unification has little impact on the average BER, as can be seen in Figure 5.8(c). In this figure we also plot the BER for 64-QAM. Comparing this plot versus the ones of BPSK and QPSK, we observe high susceptibility of any unification scheme that directly maps low-order modulation schemes to 16-QAM and 64-QAM (e.g., CBM [58]).

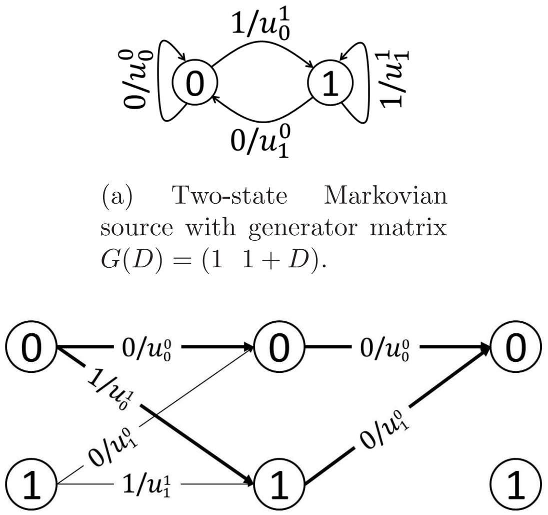
5.4.3 Untraceable Trellis-Coded Modulation Unification

Assuming accurate FO estimation, FCJ can maintain the same demodulation performance of the original modulation scheme by coding the $\mathcal{M}_M$-modulated symbols. Coding creates dependency among successive symbols, which can be exploited at Bob to guess the possible symbols sequences; hence, reducing the demodulation error. To construct these possible sequences at Bob, trellis diagram is employed, based on which the most probable path (i.e., sequence) is identified. Trellis-coded modulation [73, 74] is a generic technique to encode the modulated symbols of a given modulation scheme by introducing dependency among modulation symbols of a higher-order modulation scheme and transmitting them. A set of "states" is defined to impose the dependency where the state transitions (and hence the transmitted symbols) are specified by information bits. This combines coding and modulation operations instead of performing them separately [73, 74]. Figure 5.9 shows an example of a 2-state TCM source and its corresponding trellis diagram that encodes BPSK symbols using a 4-symbol modulation scheme, which is a way of upgrading BPSK to QPSK. Introducing the dependency in FCJ is possible because a given symbol s may correspond to multiple symbols of $\mathcal{M}_M$ (for different j values). For the time being, however, let the choice of set $\mathcal{U}_j$ be based on only the current source state. The asymptotic coding gain of TCM is defined as

$$\gamma_i^{(n)}(M) = \frac{d_{free}^2}{d_{min,i}^2} \tag{5.2}$$

where d_{free} is the minimum total Euclidean distance between the symbols along any two distinct paths in the trellis diagram (free distance) and n is the number of states. To get the benefit of TCM, d_{free} should be at least as large as $d_{min,i}$. While, in general, complex TCM codes of rate $\log_2|\mathcal{M}_i|/\log_2|\mathcal{M}_M|$ can be designed to significantly improve the gain [58], in here we exploit two simple yet efficient codes of rate $\log_2|\mathcal{M}_i|/(\log_2|\mathcal{M}_i|+1)$ which facilitate our *indistinguishable* modulation unification. These codes are based on either two-state (Figure 5.9) or four-state (Figure 5.10) TCMs presented in [73] with constraint lengths of 1 and 2, respectively. They embed $|\mathcal{M}_i|$ symbols into $2|\mathcal{M}_i|$ symbols of $\mathcal{M}_M$. (When $\mathcal{M}_i \neq$ BPSK, the same structures are used but multiple parallel edges need to be defined for each state transition.) The $2|\mathcal{M}_i|$ symbols consist of the symbols of two sets $\mathcal{U}_{j_1}$ and $\mathcal{U}_{j_2}$, $j_1, j_2 \in 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$. (In Figure 5.9 and 5.10, $\mathcal{U}_0$ and $\mathcal{U}_1$ are used.)

Using Ungerboeck's symbol assignment rules (as shown in Figure 5.9 and Figure 5.10), we maximize d_{free} for each TCM scheme and eliminate the performance loss of the uncoded modulation unification, without incurring significant decoding complexity. (Note that, for example, the least-complex code in [58] for mapping from BPSK to 64-QAM has the constraint length of 5.) The coding gains of the proposed TCM schemes are shown in Tables 5.1 and 5.2. The two-state TCM main-



(a) Two-state Markovian source with generator matrix $G(D) = (1 \;\; 1 + D)$.

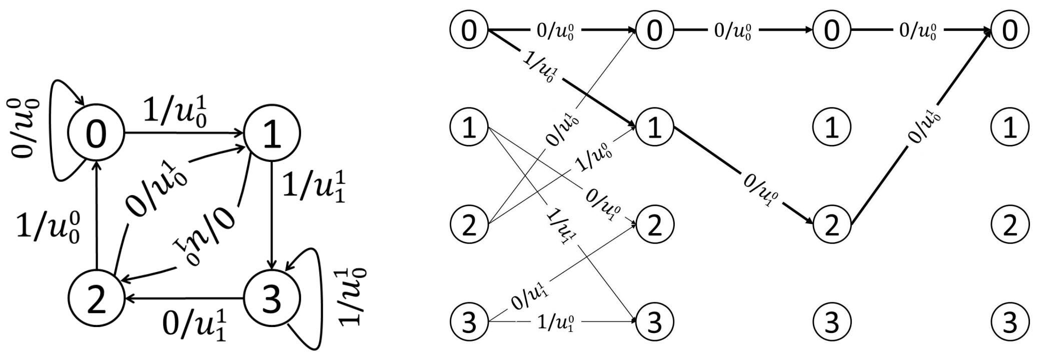
(b) Two-state trellis diagram and two paths with the minimum total Euclidean distances across different possible pairs of paths.

Figure 5.9: Minimal two-state TCM scheme. The edge label I/O denotes the transmission of symbol O if the input is I.

tains the performance of the system only in some of the cases (e.g., $\gamma_2^{(2)}(3) = 0$ dB), but the four-state TCM provides gain over $\mathcal{M}_i$-modulated transmissions in all the cases. One advantage of having a low constraint length is that when the Viterbi algorithm is employed at Bob to identify the true symbols, small delay and memory overheads are incurred for tracking and storing the most probable paths and retrieving the original symbols.

The TCM codes in FCJ take advantage of only $2|\mathcal{M}_i|$ symbols out of $|\mathcal{M}_M|$, in contrast to the codes in [58], which use all $|\mathcal{M}_M|$ symbols. Such selection exhibits from lower constellation density than $\mathcal{M}_M$, and so is less susceptible to inaccurate FO estimation than CBM [58]. By not using all $|\mathcal{M}_M|$ symbols, Alice and Bob also have the freedom of changing the edge labels from one state transition to another, which is exploited in FCJ for providing indistinguishable modulation unification.

The known code rate and the dependency among coded symbols in these TCM



(a) Four-state Markovian source with generator matrix $G(D) = \begin{pmatrix} D & 1 + D^2 \end{pmatrix}$.

(b) Four-state trellis diagram and two paths with the minimum total Euclidean distances across different possible pairs of paths.

Figure 5.10: Minimal four-state TCM scheme. The edge label I/O denotes the transmission of symbol O if the input is I.

codes may leak the original modulation scheme. Because they do not utilize all possible $\mathcal{M}_M$-modulated symbols, the number of distinct generated symbols may disclose $|\mathcal{M}_i|$; hence the original modulation scheme. Furthermore, the dependency among the symbols along the trellis diagram can reveal the original modulation order. Eve can employ different techniques to discern $\mathcal{M}_i$ using observed sequence of $\mathcal{M}_M$-modulated symbols. For example, by applying hidden Markov model techniques, she can first obtain the number and the sequence of states of a TCM scheme, and then, project the observed symbols on the trellis structure to find out the original modulation scheme.

To prevent this leakage of information, we propose exploiting the FJ bits j to randomly change the sets $\mathcal{U}_{j_1}$ and $\mathcal{U}_{j_2}$ to $\mathcal{U}_j$ and $\mathcal{U}_{(j+1) \bmod |\mathcal{M}_i|}$, respectively, at each state transition and generate all $\mathcal{M}_M$ symbols. From a security perspective, randomly replacing $\mathcal{U}_j$'s allows us to use all $|\mathcal{M}_M|$ symbols with equal probability. Because Bob knows j, he can shrink the set of possible symbols to the ones mandated by the FJ bits, and keep track of transitions and compute the distance/error between the received symbol sequence and the possible sequences. So the coding gain, obtained above, remains valid. From Eve's perspective, however, any two successive symbols are completely independent of each other, i.e., the dependency among symbols is destroyed, because for a given symbol, the next symbol is selected completely randomly based on the random information bits together with the random FJ bits. Therefore, the TCM codes become untraceable and $\mathcal{M}_i$ is not disclosed to Eve.

The untraceable-TCM-based modulation unification scheme proposed above hides the true modulation scheme of the frame's payloads, if this payload contains random bits. This scheme also makes it hard for Eve to guess the unencrypted parts of the MAC (and PHY-, if unified) layer headers. For example, when $\mathcal{M}_i =$ BPSK and four-state TCM is used, any of the $|\mathcal{M}_M|$ symbols can be used according to $\mathbf{j}$ to modulate an input bit, depending on the current state (see Figure 5.10(b)). As long as the current TCM state is unknown to Eve, she cannot guess this input bit. However, because the *static* mapping $\mathcal{F}_{\mathbf{j}}(.)$ and the labeling used in the underlying TCM structure are not necessarily secret, a transmission is still vulnerable to the following attacks. First, if the initial state is not secret and if $\mathcal{M}_i$ is known (e.g., the modulation scheme of the PHY header), Eve will be able to track the time-evolution of the states and eventually, guess the input bits. To illustrate, the mapping $\mathcal{F}_{\mathbf{j}}(.)$ partitions the $|\mathcal{M}_M|$ symbols into $|\mathcal{M}_M|/|\mathcal{M}_i|$ disjoint sets, one for each original symbol s and different FJ bits j. From the inverse function ${\mathcal{F}_{\mathbf{j}}}^{-1}$ and the current state, Eve can determine the symbols s from their $\mathcal{M}_M$-modulated counterparts, revealing the true content of that field. This is especially the case if Eve exhibits a high SNR and can reliably detect the $\mathcal{M}_M$-modulated symbol. Note that TCM does not impact $\mathcal{F}_{\mathbf{j}}(.)$.

Another attack occurs on a semi-static field if $\mathcal{M}_i$ is known. In this case, Eve can guess the field value by trying different possible TCM states in the beginning of the field and comparing the sequence of received symbols with the few possible symbol sequences for each initial state. In these two cases, Eve is able to extract unencrypted fields in the PHY and (if the rate field is disclosed) MAC headers by detecting the frame preamble and obtaining the $\mathcal{M}_M$-modulated symbols of the target header field.

A third attack involves an unknown $\mathcal{M}_i$ but some parts of the payload, e.g., MAC header, can take one of a few possible values (or even if these semi-static parts are encrypted but with a time-invariant cipher). Eve may again apply $\mathcal{F}_{\mathbf{j}}^{-1}$ of different $\mathcal{M}_i$'s on the $\mathcal{M}_M$-modulated symbols and check which one produces one of the known values. This reveals not only the content (if unencrypted), but also the true modulation scheme of the payload.

To remedy the above vulnerabilities, we also need to encrypt PHY/MAC headers using a time-varying cipher, e.g., a one-time pad. Such encryption, however, is not trivial. It creates challenges and prevents some of the essential functions of the header. For example, the decryption operation at Bob requires knowledge of the shared key dedicated to the Alice-Bob session. This key is different for a different session (e.g., Charlie-Bob session). In order to retrieve the right key, Bob needs to know the sender's identity of the incoming frame, which is typically done using the MAC address. But with the MAC address being encrypted, Bob cannot identify Alice and retrieve the right key. An unencrypted MAC address also cannot be used to sender identification at the lower layer. Moreover, generating one-time pads requires a good pseudo-random number generator that is robust to plaintext (e.g., semi-static fields) attacks and also time-varying seeds that are common between Alice and Bob. In the following section, we propose a novel approach for providing time-varying identification based on embedding a message in the preamble (i.e., before the to-be encrypted headers). This approach provides synchrony between Alice and Bob for using the same seed. Note that a sender identifier based on the Alice-Bob channel characteristics will not work when nodes are mobile. It can also be spoofed if the channel is estimated by the attacker [69].

5.5 Preamble-based PHY-Layer Identifier

Alice and Bob need to establish an identification method that is robust to mobility/channel variations and can be used as a means to synchronously generate the FJ traffic at PHY-layer. Such an identifier should also be different from one frame to another; otherwise, a semi-static header field that can take a few possible values (e.g., the 8-bit Signal field in 802.11b takes four possible values) would produce a fixed set of constellation points in $\mathcal{M}_M$. After eavesdropping on several frame transmissions that may have different values for that field, Eve may estimate the part of $\mathbf{j}$ used to protect that field. This can facilitate a dictionary attack and disclose the field values. Moreover, in the case of a packet loss and retransmission, applying the same $\mathbf{j}$ results in the same sequence of $\mathcal{M}_M$-modulated symbols. Eve may then correlate successive transmissions, detect retransmissions, and then exclude them from the statistics used to fingerprint the session (e.g., packet size histogram). Furthermore, if Alice and Bob instead synchronously use different parts of a common $\mathbf{j}$ for different frames, the loss of an ACK would make Alice and Bob out-of-sync. For these reasons, we require $\mathbf{j}$ to be generated via an identifier that varies from one frame to another. In this section, we explain how a secret frame-specific $\mathbf{j}$ is generated based on the PTK and frame/sender identifier.

We exploit a PRNG that is constructed based on a cryptographic hash function from the standardized family of SHA-3 algorithms (e.g., [75, 76]) to generate $\mathbf{j}$ based on a seed value and frame IDs. These algorithms enjoy several attractive properties. First, a single-bit change in the seed results in a completely different hash value (equivalently, $\mathbf{j}$ generated by the PRNG). Therefore, as long as the seed is not repeated, $\mathbf{j}$ will not be repeated (similar to an ideal one-time pad), thus preventing dictionary attacks. The randomness of the seed values in our method will be discussed in the next section. Second, if Eve captures the hash value, she cannot use it to recover the key or the seed value used to generate $\mathbf{j}$, i.e., it is one-way hash and robust against *chosen-plaintext* attacks [76], which are stronger than known-plaintext attacks. Third, if Eve captures some part of $\mathbf{j}$ (or the frame ID, defined later), she cannot predict subsequent values of $\mathbf{j}$ or ID (i.e., robustness to generic state recovery attacks) [75, 76]. Fourth, the fact that such PRNG can be built in a very compact core and can be implemented using bitwise Boolean instructions and rotations within bytes only [75] makes it very resource-efficient and suitable for embedded devices with low overhead/delay requirements. Fifth, the security of $\mathbf{j}$ generated by such PRNG can be compared to the security of an ideal random number sequence that does not have any generic flaw, i.e., indifferentiability property [75]. Altogether, the sequence $\mathbf{j}$ that is used for encrypting headers containing semi-static fields and for unifying the modulation schemes provide confidentiality for the headers and unpredictability for the modulation unification approach.

If the seed contains nothing but the secret PTK, the stream cipher $\mathbf{j}$ will always remain the same. To vary $\mathbf{j}$ from one frame to another, we concatenate a non-secret frame-specific ID, denoted by $\mathcal{ID}$, to the PTK and compose a partially secret seed for the given frame (similar to the method in [76]). $\mathcal{ID}$ is also used to simultaneously identify and authenticate the sender/session, allowing Bob to distinguish Alice's transmission from other transmissions (e.g., Charlie's) destined to Bob. FCJ embeds $\mathcal{ID}$ in the frame preamble and transmits it in the clear.

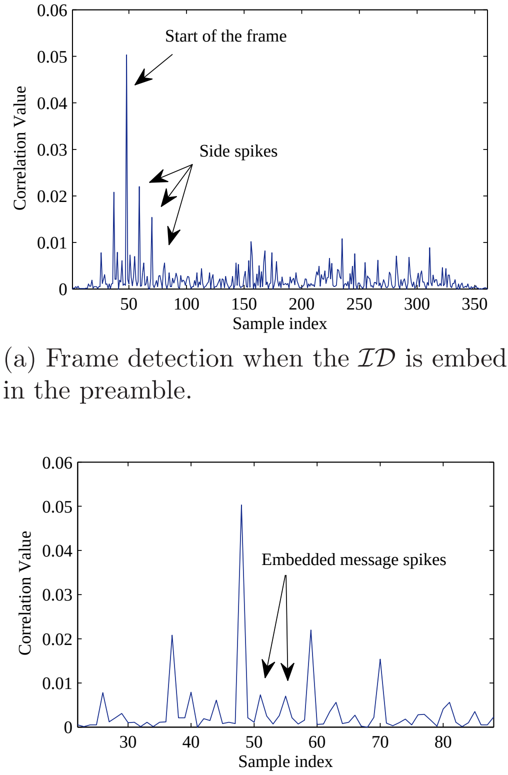
With frame-specific and time-rolling $\mathcal{ID}$s during a session, Eve will not be able to identify and track the user or correlate different frames that belong to the same session. However, Bob must be able to associate different $\mathcal{ID}$s to the same sender (e.g., Alice). We adopt an $\mathcal{ID}$ generation method similar to [77] and create a *chain* of confidential $\mathcal{ID}$s using SHA-3 hash algorithm and PTK. For the first frame, Alice and Bob agree on an initial $\mathcal{ID}$ (e.g., during the 4-way handshake). The $\mathcal{ID}$ for subsequent frames will be the hash of the previous frame using PTK. To account for possible frame losses and retransmissions, Bob maintains a short chain of subsequent $\mathcal{ID}$s for each active neighbor and checks whether or not the received $\mathcal{ID}$ exists in the chain. It is followed by the properties of SHA-3 that the chance of collision between the $\mathcal{ID}$s of different senders will be low. Moreover, by observing one $\mathcal{ID}$, Eve cannot predict the next one.

5.5.1 Embedding the $\mathcal{ID}$

To embed the non-secret $\mathcal{ID}$, one may introduce a new field between the preamble and the standard PHY header. However, to keep the standard PHY frame format intact for interpretability purposes and also to avoid increasing the frame size, we embed encoded $\mathcal{ID}$ onto the known preamble via analog-signal superposition. (Note that we cannot use any reserved bits in the header(s) because the entire header is supposed to be encrypted.) The design below is specific to the 802.11b preamble, but the idea can be extended to other preamble structures.

Extracting $\mathcal{ID}$ from the superposition is critical for Bob. At the same time, Bob does not want to lose the important functions of the preamble as a result of this superposition. To satisfy both requirements, we propose using cyclically rotated Barker sequences (Section 5.2) to encode Alice's $\mathcal{ID}$. When a Barker sequence is aligned with the original preamble, the function $\mathcal{R}(\mathbf{b}, n)$ (defined in (2.8)) spikes, indicating the start of a frame. To preserve this spike, we utilize cyclically shifted versions of the reference 11-chip Barker sequence. Every k-shifted sequence, $k = 1, \ldots, 10$, can be a message ($\mathcal{ID}$). Because of the orthogonality of Barker sequences, this overlaid $\mathcal{ID}$ is easily detectable with RF correlation. Moreover, unless the power of the superposition is normalized, the frame detection process will be negligibly affected because the encoded message will have little contribution to the correlation with the reference sequence, when aligned properly. To maintain the original preamble power, Alice can multiply the preamble by the normalization coefficient of $\sqrt{11/20}$ (i.e., 2.6 dB reduction in the power of original preamble). The peak-to-average-power ratio (PAPR) of the preamble is also increased by 3.42 dB.

Figure 5.11(a) is an example drawn from our experiments (Section 5.6) that



(a) Frame detection when the $\mathcal{ID}$ is embedded in the preamble.

(b) Small but detectable spikes during the preamble due to the embedded $\mathcal{ID}$.

Figure 5.11: $\mathcal{R}(.,n)$ computed over a frame.

shows the value of $\mathcal{R}(.,n)$ when applied over a frame with two embedded rotated Barker sequences, repeated in each half of the preamble. The preamble in this example consists of four Barker sequences, which create a few side spikes when the correlator is moved a multiples of 11 indices away from the beginning of the preamble. Figure 5.11(b) zooms into the preamble and shows the two *messages spikes* (i.e., spikes corresponding to the cyclicly rotated Barker sequences) between every two successive preamble (side) spikes.

For each frame (including retransmissions), Alice generates a new $\mathcal{ID}$, which is conveyed by concatenating several k-shifted versions of the Barker sequence and is superimposed on the original preamble in the analog domain. Specifically, let

Preamble ($\mathcal{P}$): 0 0	+1 −1 +1 +1 −1 +1 +1 +1 −1 −1 −1 +1 −1 +1 +1 −1 +1 +1 +1 −1 −1 −1
$\mathcal{ID}$: 3 7	+1 −1 +1 +1 +1 −1 −1 −1 +1 −1 +1 +1 −1 −1 −1 +1 −1 +1 +1 −1 +1 +1
$\mathcal{P}^* = \mathcal{P} + \mathcal{ID}$	+2 −2 +2 +2 0 0 0 0 0 −2 0 +2 −2 0 0 0 0 +2 +2 −2 0 0
$\mathcal{R}(\mathcal{P}, 0)$	22^2

Table 5.3: Example of the concatenation of two Barker sequences to embed $\mathcal{ID}$ value (3 7) in the preamble.

$(k_1 k_2 \ldots k_l)_{10}$ be the decimal representation of the value of $\mathcal{ID}$, where k_i, $i = 1, \ldots, l$, is the i-th most-significant digit. Then, the value of k_i is conveyed in a cyclically shifted Barker sequence with $k_i + 1$ shift. Concatenation of the l shifted Barker sequences produces $\mathcal{ID}$ (see the example in Table 5.3). Bob is still able to detect the preamble and the $\mathcal{ID}$, as shown in Figure 5.11. The steps taken by Bob to extract $\mathcal{ID}$ and perform the preamble functions are summarized as follows:

1. Detect frame, estimate FO, and compensate for it.
2. Extract frame $\mathcal{ID}$.
3. Construct a new reference preamble using the original preamble and the embedded $\mathcal{ID}$.
4. Perform channel estimation using the new preamble.
5. Look up the PTK associated with the session ID and start generating $\mathbf{j}$.

5.5.2 Implication on PHY-layer Functions and Practical Issues

Embedding $\mathcal{ID}$ in the preamble may affect some of the preamble's common functions. We discuss how our message embedding mechanism can maintain these functions.

(1) Frame detection. A typical receiver performs sliding-window correlations using different time offsets (parameter n in (2.8)). In the case of FCJ, the ratio between the height of the side spikes and the main spike remains the same, but the superposed $\mathcal{ID}$ will cause a few spikes when Bob correlates the reference preamble with the received signal at time offsets $k_1, \ldots, k_l$, from the start of the preamble. To avoid creating an alias of the actual start of the preamble, Alice makes sure that she uses different rotation values over successive preamble bits. Let the number of such successive rotations be $l < 11$ ($l \neq 6$). Excluding the noise and multipath channel effect, the message spikes cannot be larger than $\frac{(6-l)^2}{(5l)^2}$ of the highest spike, because in every sequence of l rotations, at most one of them will perfectly align with the correlating sequence, i.e., the original preamble. Note that the correlation value of two Barker sequences with the same (different) rotation value(s) is $|11|^2$ ($|-1|^2$).

(2) FO estimation. As explained in Section 5.2, FO estimation requires two identical repetitions of an arbitrary sequence. We satisfy this requirement by repeating the $\mathcal{ID}$-bearing signal at least twice. Specifically, if Bob uses K repetitions of the Barker sequence (preamble bits) for FO estimation, Alice places the $\mathcal{ID}$-bearing signal in the first $K/2$ sequences and then repeats it over the other $K/2$ sequences. (If $K > 2l$, Alice uses the last l bits in each half to superimpose the $\mathcal{ID}$.) If Alice does not know K a priori, she only exploits the portion of the preamble that will likely be detected by Bob. Bob can then find the start of the $\mathcal{ID}$ signal either by an energy-increase detection, or by iteratively running (on each preamble bit) a series of threshold-based correlations with nonzero rotations of the Barker sequence. Once a correlation value exceeds the threshold, this indicates the start of the $\mathcal{ID}$ signal.

(3) Message capacity and error correction. There are 10 distinct rotations of an 11-chip Barker sequence (one preamble bit). In DBPSK, this translates to 10 different $\mathcal{ID}$s per preamble bit. So in every nine out of 128 bits of the preamble, 10! different $\mathcal{ID}$s of the decimal form $(k_1 k_2 \ldots k_9)_{10}$ can be embedded. Using DQPSK, we can double the number of possible $\mathcal{ID}$s. Given this large number, FCJ can ensure the randomness required by the PRNG even when Alice introduces a coding scheme over the set of $\mathcal{ID}$s to reduce the $\mathcal{ID}$ detection errors (e.g., using $\mathcal{ID}$s with large Hamming distances).

(4) Channel estimation. A known sequence, such as the preamble, is often used for channel estimation. Upon capturing $\mathcal{ID}$, Bob constructs a new "temporary" preamble by superposing the same $\mathcal{ID}$-bearing signal over the original preamble, and uses the new preamble for channel estimation.

5.5.3 Encryption of Header Fields

We apply a modulation-level stream encryption $\mathcal{E}_{\mathbf{j}}(\mathcal{M}_i)$ to the $\mathcal{M}_i$-modulated symbols of the frame (payload + header)[3] to randomize the location of the original symbols in the constellation map of $\mathcal{M}_i$ (or equivalently, *dynamically* change the mapping between a symbol s and one of the disjoint sets determined by $\mathcal{F}_{\mathbf{j}}(.)$). This way, sole knowledge of $\mathcal{F}_{\mathbf{j}}$ is not sufficient to disclose the symbol s that corresponds to an observed $\mathcal{M}_M$-modulated symbol. $\mathcal{E}_{\mathbf{j}}(.)$ can be applied before $\mathcal{F}_{\mathbf{j}}$ or jointly with $\mathcal{F}_{\mathbf{j}}$. Note that if we alternatively upgrade the modulation scheme first and then apply encryption, Bob may not reliably decode an $\mathcal{M}_M$-modulated symbol.

The encryption function $\mathcal{E}_{\mathbf{j}}(\mathcal{M}_i)$ is performed by bit-wise XORing of the information and FJ bits. Consider $\log_2|\mathcal{M}_i|$ information bits, corresponding to one symbol of the modulation scheme $\mathcal{M}_i$. We select $\log_2|\mathcal{M}_i|$ successive bits from $\mathbf{j}$ and XOR them with the information bits. In the symbols domain, a lookup table can be used to map the decimal value of the FJ bits, denoted by x, and the index of information symbols on the constellation map to the symbol index corresponding to the XOR of the underlying information and FJ bits. According to Gray coding, adjacent points in the constellation map of $\mathcal{M}_i$ have a 1-bit difference. Equivalently, the encryption can be merged with TCM by changing edge labels u_j^k with $u_j^{k \bigoplus x \bmod |\mathcal{M}_i|}$ per each transition. One advantage of using an XOR operation is that adjacent constellation points before the symbols relocation by $\mathcal{E}_{\mathbf{j}}(\mathcal{M}_i)$ remain adjacent after the relocation because they are bit-wise XORed with the same FJ bits and thus the Gray coding

[3] We do not encrypt the preamble, since otherwise Bob cannot detect the start of the frame without knowing in advance the sender's identity.

property is preserved (in contrast to the encryption scheme in [59]). Therefore, the BER due to demodulation errors is not impacted by modulation encryption. As long as the FJ traffic is robust against various attacks (e.g., plaintext attack), the encryption $\mathcal{E}_{\mathbf{j}}(\mathcal{M}_i)$ is secure. As we discussed earlier, SHA-3 family of hash functions can provide us with such PRNG.

Altogether, Alice applies the composite mapping $\mathcal{F}_{\mathbf{j}}\big(\mathcal{E}_{\mathbf{j}}(\mathcal{M}_i)\big)$ to the payload symbols. For each $\mathcal{M}_i$-modulated symbol, Alice (Bob) sequentially picks a block of $\log_2|\mathcal{M}_i| + \log_2\frac{|\mathcal{M}_M|}{|\mathcal{M}_i|}$ bits from $\mathbf{j}$ to first encrypt (recover) the symbol and then upgrade (decrypt) it.

If the PHY header symbols are upgraded, Bob treats the modulation-encrypted header and payload the same way, except that the true modulation order for the PHY header is known a priori. So Bob knows in advance how many bits from $\mathbf{j}$ are needed to decrypt and recover the header. The modulation scheme for the payload is determined after the PHY header has been decoded and the rate field recovered. Eve, on the other hand, cannot correctly decode the header because it is modulation-encrypted by the secret $\mathbf{j}$. As long as the rate field in the header is unknown, Eve cannot determine $\mathcal{M}_i$ of the payload, and hence does not know how many information bits are associated with an observed symbol.

5.6 Performance Evaluation

We implement *Friendly CryptoJam* in NI LabVIEW programming environment. Our LabVIEW PHY-layer libraries include the transmitter components in Figure 5.4, as well as frame timing and detection, channel and FO estimation modules at the receiver. Using the same LabVIEW code, we emulate wireless transmissions with all the transmitter/receiver components in an AWGN channel and then empirically evaluate FCJ on an NI-2922 USRP testbed controlled by the LabVIEW USRP driver.

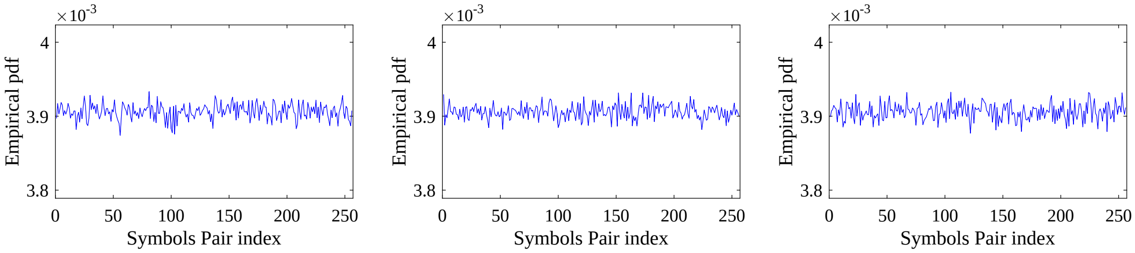
(a) Metrics. We evaluate the BER performance and preamble-related operations, such as frame detection and FO estimation, for different SNR (transmission power in the experiments) values and modulation schemes. The $\mathcal{ID}$ extraction success rate is another important metric of interest.

(b) FJ traffic. To generate $\mathbf{j}$ and evaluate the communication metrics (e.g., BER), or to generate $\mathcal{ID}$ and evaluate the detection rate, we do not implement SHA-3, which is beyond the scope of this dissertation. Instead and without loss of generality, we exploit the LRSR-based PRNG available in LabVIEW with Galois implementation and polynomial degree of 12 (or 14). IEEE 802.11a systems use the same type of PRNG. For each frame, we generate a random sequence (or an $\mathcal{ID}$, depending on the metric of interest) and share it between Alice and Bob. With respect to the security of our scheme against plaintext and key-recovery attacks, we rely on the theoretical and reported properties of SHA-3.

(c) Modulation. We use four basic modulation schemes, BPSK, QPSK, 16-QAM, and 64-QAM. The modulation mappings follow set-partitioning rule (e.g., Figure 5.5 and Figure 5.6) and Figure 5.9 and Figure 5.10 for TCM-based modulation unification. The parameters of such upgrades are shown in Tables 5.1 and 5.2.

(d) Physical frame. Unless specified otherwise, each frame consists of a 66-bit Barker code DBPSK-modulated preamble (six 11-chip Barker sequences) with a random three-digit embedded $\mathcal{ID} = (k_1 k_2 k_3)_{10}$ followed by a random payload. The frame is transmitted over a 2.4 GHz frequency band at a symbol rate of 1 Msamples/s in the simulations and 83.3 Ksamples/s in the USRP experiments.

(e) Viterbi decoder. The receiver implements the Viterbi algorithm to decode the TCM-based symbols. We studied the performance of the decoder for different path truncation depths. It turned out that when $\mathcal{M}_M =$ 16-QAM, the depths of 5 and 10 for the two-state and four-state TCM schemes, respectively, are large enough to achieve the desired performance. When $\mathcal{M}_M =$ 64-QAM, the depths of 17 and 30 are sufficient. Higher depths did not produce noticeably better results. Therefore,



(a) $\mathcal{M}_i$ = BPSK (b) $\mathcal{M}_i$ = QPSK (c) $\mathcal{M}_i$ = 16-QAM

Figure 5.12: Empirical probability density functions of pairs of successive modulated symbols using $\mathcal{M}_M$ = 16-QAM and different $\mathcal{M}_i$'s. The input bit sequence is generated using uniform distribution.

the maximum decoding delay imposed by FCJ is bounded by 10-30 symbol times, depending on $\mathcal{M}_M$.

5.6.1 Modulation Scheme Indistinguishability

It may be argued that the dependency (correlation) that is introduced by TCM among successive $\mathcal{M}_M$-modulated symbols could be used by Eve to distinguish between $\mathcal{M}_i$-modulated symbols embedded in $\mathcal{M}_M$, $\mathcal{M}_i \neq \mathcal{M}_M$, from true $\mathcal{M}_M$-modulated symbols. To verify the indistinguishability property of our modulation schemes unified by the proposed untraceable TCM, we employ Kolmogorov-Smirnov (KS) statistical test to compare sequences of $\mathcal{M}_i$-modulated symbols embedded in $\mathcal{M}_M$ constellation to the sequences of true $\mathcal{M}_M$-modulated symbols. In particular, we consider the empirical probability distributions (pdfs) of transmitted symbols as well as pairs of successive symbols. The latter one is important because If Eve detects any correlation between two successive symbols (provided that the information bits are random), she may conclude that $\mathcal{M}_i \neq \mathcal{M}_M$ and may also be able to discern $\mathcal{M}_i$.

Without loss of generality, we consider $\mathcal{M}_M$ = 16-QAM; hence, 256 pairs of symbols. In Figure 5.12, we plot the empirical probability distributions of successive-symbols pairs in a pool of 2×10^6 transmitted symbols when all bits in information

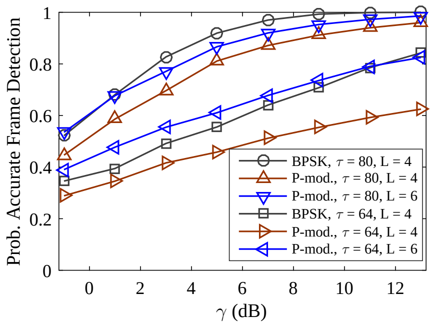


Figure 5.13: Impact of embedded $\mathcal{ID}$ on frame detection (emulations).

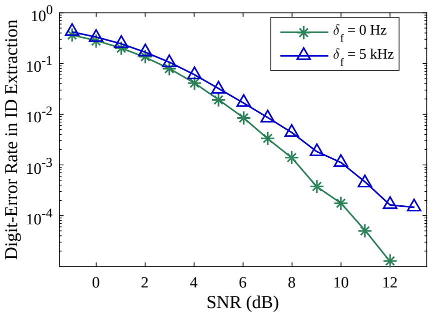


Figure 5.14: Digit-error rate in $\mathcal{ID}$ detection vs. SNR (emulations).

and FJ sequences are randomly selected from a uniform distribution. At a confidence level of 97.5%, the KS test approves that the three empirical pdfs are drawn from the same (uniform) probability distribution function and so are indistinguishable. Because Alice uses only $\mathcal{M}_M$ for transmission, Eve will think that $\mathcal{M}_M$ is the underlying modulation scheme. By applying $\mathcal{M}_M$ to demodulate the symbols that were originally modulated using $\mathcal{M}_i \neq \mathcal{M}_M$, Eve's estimate of the payload size will be incorrect and further, BER will be maximum.

5.6.2 System Emulations

To assess the performance of individual components of FCJ, we decouple the unification/encryption schemes from the message embedding approach. AWGN channel model is considered to emulate frame transmission and reception. In the emulations, FO is a controllable parameter, whereas in the experiments, it is a feature of the USRP radio oscillator.

Identifier embedding

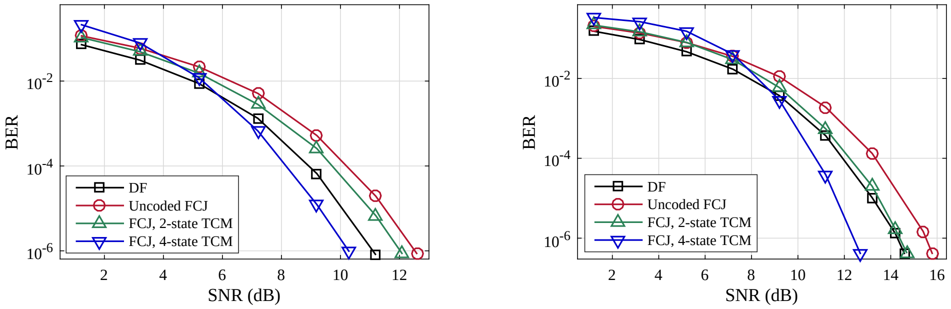
First, we consider the $\mathcal{ID}$ embedding scheme and study how much the superposition of $\mathcal{ID}$ onto the preamble affects frame detection and FO estimation accuracy.

Once the frame is detected, Bob uses the two identical halves of the preamble to estimate Δ_f and compensates for it before $\mathcal{ID}$ extraction. We also measure the performance of the (uncoded) $\mathcal{ID}$ detection method at Bob in the presence of residual FO estimation errors.

Frame detection is the first step in the decoding process. It starts by a threshold-based energy detection, followed by the cross-correlation of the received samples r against a series of the known Barker sequences. We assume that the average total transmission power with and without an superposed $\mathcal{ID}$ onto the preamble is expected to be the same. Figure 5.13 shows that the power reduction for the original preamble in our embedding scheme results in about 2 dB loss in frame detection; irrespective of Δ_f. Although three distinctly shifted Barker sequences (repeated twice) generate additional message spikes and also Bob is still agnostic to the embedded identifier, the highest of these spikes in the absence of noise and random payload will not be more than 4% of the spike corresponding to the beginning of the preamble (see Section 5.5.2).

Bob then moves on to the next phase; FO estimation. Even though the $\mathcal{ID}$ superposition in FCJ results in variable amplitudes for different symbols (in fact, some of the symbols will have zero amplitude), the results (not shown here) show that the symmetry between two parts of the $\mathcal{ID}$-bearing signal helps Bob in maintaining the same FO estimation performance without FCJ. The reason is that for estimating $\varphi(T)$, the amplitude of the identical pairs is usually taken into account. Therefore, the noise cannot dominate the FO estimation process in FCJ more than default scheme.

While in current 802.11 systems Bob needs to successfully decode the sender's 32-bit MAC address to decrypt an encrypted payload, in FCJ Bob needs error-free extraction of the $\mathcal{ID}$ to generate $\mathbf{j}$. In Figure 5.14 we show the digit-error rate performance of our preamble-based identification scheme. Assuming that the $\mathcal{ID}$s are uncoded, Bob needs to successfully detect all the l digits of an $\mathcal{ID}$. The results



(a) $\mathcal{M}_i$ = BPSK (b) $\mathcal{M}_i$ = QPSK

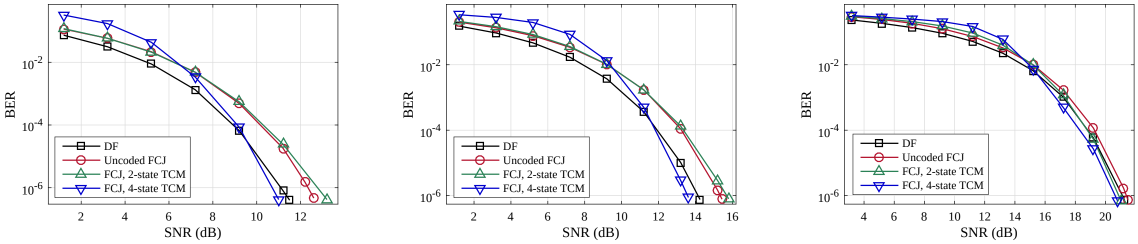
Figure 5.15: BER versus received SNR of modulation unification at Bob when $\mathcal{M}_M$ = 16-QAM (emulations).

confirm that FCJ can correctly convey the sender identifier with high probability. For example, When SNR= 8 dB, the uncoded identifier embedding has a digit-error rate of 1.5×10^{-3}. So for a concatenation of l of such identifiers, the success rate will be 0.9985^l (e.g., $0.9985^{10} = \%98.5$, which is equivalent to correct decoding of a 21-bit binary sequence, which has comparable capacity, when BER $= 7 \times 10^{-4}$). If channel coding is employed for encoding the $\mathcal{ID}$, FCJ can deliver even higher identifier detection rate. In Figure 5.14 we also the digit-error rate performance when the residual Δ_f estimation error is very high. It can be seen that even with significantly high FO estimation error, the detection rate is high. (When $\Delta_f < 1$ kHz, the performance is the same as when $\Delta_f = 0$. Those results are not shown in the figure.)

At this point, Bob constructs the new preamble for CSI estimation, which essentially includes estimating the constant phase offset.

TCM-based modulation unification

Now we study the performance of the employed TCM schemes compared to the uncoded unification scheme [59] and the default operation of 802.11 without FCJ (referred to as DF), as our benchmark. BER in digital communications not only



(a) $\mathcal{M}_i$ = BPSK (b) $\mathcal{M}_i$ = QPSK (c) $\mathcal{M}_i$ = 16-QAM

Figure 5.16: BER versus received SNR of modulation unification at Bob when $\mathcal{M}_M$ = 64-QAM (emulations).

depends on SNR and $d_{min,i}$, but also on FO estimation accuracy. In order to focus only on the impact of modulation encryption/unification, in this subsection, we assume $\Delta_f = 0$ but Bob still have to correctly detect the frame and estimate the CSI.

Figure 5.15 and Figure 5.16 depict the BER performance of FCJ as a function of the SNR at Bob for different modulation schemes $\mathcal{M}_i$ when $\mathcal{M}_M$ = 16-QAM and 64-QAM, respectively. When BPSK is embedded into 16-QAM (Figure 5.15(a)), the two-state TCM scheme can alleviate to some extent the performance loss due to the (uncoded) unification. However, using the four-state TCM scheme, Bob approaches the asymptotic coding gain without leaking the original modulation scheme. Note that if the underlying bit sequence belongs to the PHY-layer header with a known modulation scheme (e.g., BPSK), Eve may be able to obtain the original (encrypted) symbols but she still is not able to decrypt them because of our robust modulation encryption. For the QPSK case in Figure 5.15(b), it can be observed that the two-state TCM scheme can be sufficient for maintaining the performance of the default operation with the minimum delay and complexity. This figure also verifies the asymptotic gains calculated in Table 5.1.

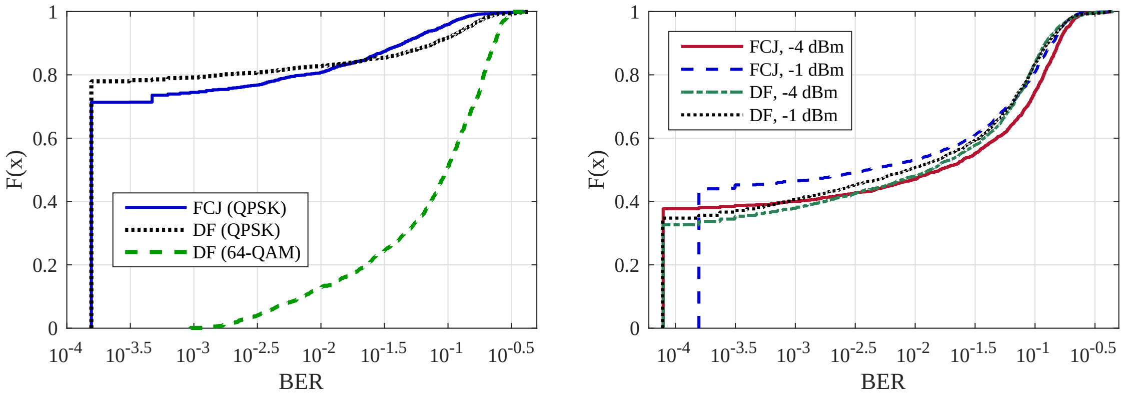
The constellation of 64-QAM is denser than the one of 16-QAM. Therefore, when $\mathcal{M}_M$ = 64-QAM, the coding gain in general will be less than the case of $\mathcal{M}_M$ =

16-QAM, as can be seen in Figure 5.16. For example, using the two-state TCM for $\mathcal{M}_i$ = QPSK is no longer sufficient in this case (Figure 5.16(b)). However, the two-state TCM is good enough when $\mathcal{M}_i$ = 16-QAM (Figure 5.16(c)). As a general rule, the higher the order of $\mathcal{M}_i$ is, less complex TCM codes can be sufficient. Alice and Bob can agree upon the least complex but sufficient TCM codes for each $\mathcal{M}_i$ to reduce the overall complexity.

5.6.3 USRP Experiments

We now exploit our USRPs, one acting as Alice and another as Bob, to evaluate real transmissions in an indoor environment. The distance between Alice and Bob is 2.2 m and we vary the transmission power. Alice and Bob each are equipped with a 3 dB antenna. The noise level at the receiver is about −87 dBm. We assume that the payload consists of 3200 symbols. This selection is to mimic a situation in which Alice hides the true size of different frames by transmitting the frames with same frame duration. Hence, when $\mathcal{M}_i$ = BPSK, QPSK, 16-QAM, and 64QAM, Alice transmits 400, 800, 1600, and 2400 bytes, respectively. Using the same number of symbols also makes the amount of phase offset errors comparable for different $\mathcal{M}_i$'s.

In our empirical evaluations, we encountered a few challenges. First, the USRPs truncates peaks of a signal with high PAPR (to avoid overflow) when the average signal power necessities transmitting the peak at a power higher than the one set by the user. In 16-QAM and 64-QAM, certain symbols (e.g., corners of the constellation map, which result in high PAPR) are often truncated; resulting in several bit errors. Compared to QPSK/BPSK, 64-QAM has 3.7 dB higher PAPR. To remedy this issue, we scaled down the average power of generated samples at the transmitter to a level that the USRPs can transmit the peak values without truncation. Furthermore, we scale the sample sequences in all experiments to the same normalized average Tx power. This solution is more reliable for comparison purposes than a solution in which the peak value is always transmitted at the maximum power but the average



(a) $\mathcal{M}_i$ = QPSK and Tx power = -8 dBm (-1 dBm for 64-QAM)

(b) $\mathcal{M}_i$ = 16-QAM

Figure 5.17: Empirical cumulative distribution function of BER (USRP results).

power varies from one frame to another.

The second challenge is inaccurate FO estimation. As reported in [58, 59] and discussed in Section 5.4.2, the FO estimation is often inaccurate in hardware experiments, which results in high BER for large frames. Denser modulation schemes and asymmetric constellation maps are often more sensitive to FO estimation errors. In FCJ, $|\mathcal{M}_i|$ symbols of a symmetric constellation are encoded to $2|\mathcal{M}_i|$ symbols and are asymmetrically distributed in the constellation of $\mathcal{M}_M$. To reduce the estimation error, we (1) increase the length of the preamble to better average out the noise, and (2) maintained a coarse estimate of Δ_f based on previous transmissions and compensating for it before performing the normal FO estimation in each run (similar to [58]). However, the estimate may still be inaccurate and result in high BER. When averaging the BER of several transmissions, a (small) subset of BERs with high values (e.g., 10^{-1}) dominates the rest of lower BER values. Therefore, we used CDF curves for comparing the performance of different schemes in order to separate the BER values due to inaccurate FO estimation from the rest. Each CDF represents the BERs of 2000 transmissions.

The third challenge is inaccurate frame detection when Δ_f is high. The Δ_f between the two USRPs at 2.4 GHz carrier frequency varies between 0.6 kHz to 1.1 kHz. At Bob, the summation of the terms with time-varying phase offsets during the preamble may reduce the value of (2.8). To address this problem, we first try to detect the frame and then calculate an *initial* FO estimate using the two correlation values with highest amplitude spikes in (2.8) before taking the absolute value (similar to the method in [28]). The phase offset between these two values is an estimate of the phase offset between two samples that are 11 samples away from each other. After compensating for this initial estimate, Bob again performs frame detection. Note that the embedded $\mathcal{ID}$ does not impact the phase offset between the two values and so the performance of this method.

In analyzing the measured payload BER, we distinguish between cases based on whether or not the frame or $\mathcal{ID}$ is correctly detected. Basically, any frame or $\mathcal{ID}$ detection error will result in a packet drop and so we exclude these cases when measuring the BER. Nevertheless, the single-digit detection rate in our experiments is $> 99\%$ even if the transmission power is set to the minimum in our setup (-8 dBm).

In Figure 5.17, we compare the performance of FCJ with the four-state TCM to the one of the default scheme. In our measurements, the SNR was high and the decoding errors were mainly due to inaccurate FO estimation. Figure 5.17(a) depicts the BER distribution when $\mathcal{M}_i =$ QPSK and $\mathcal{M}_M =$ 64-QAM. In this case, we set the Alice's transmission power to its minimum. Even though the median BER in both cases is the same ($10^{-3.8}$), erroneous FO estimation and accumulation of phase error in this case results in slightly worse performance compared to the default scheme (as explained in Section 5.4.2). In the same figure, we also show the performance of 64-QAM, which is significantly impacted by erroneous FO estimation. However, our scheme performs better than the default scheme when $\mathcal{M}_i =$ 16-QAM (see Figure 5.17(b)). The reason is that the impact of residual FO estimation errors on our scheme in this case is similar to its impact on 16-QAM. Moreover, when the residual error is low, the TCM code helps Bob in FCJ to correct few bits in error and achieve higher error-free transmissions. This can be seen in Figure 5.17(b) for two different transmission powers. The median BER values for FCJ and the default scheme are $10^{-1.79}$ and $10^{-1.879}$, respectively, when the transmission power is -4 dBm. These values when the transmission power is -1 dBm are $10^{-2.388}$ and $10^{-2.06}$, respectively.

5.7 Summary

Preventing the leakage of transmission attributes, including unencrypted PHY/MAC header fields and the payload's modulation scheme, is challenging. In this paper, we proposed *Friendly CryptoJam* (FCJ), a combination of friendly jamming and low-level encryption, to effectively protect the confidentiality of lower-layer fields and prevent SCI-based traffic classification, rate-adaptation, plaintext, dictionary, modulation detection, and device-based tracking attacks. FCJ employs three main techniques. First, modulation-aware encryption is used to perfectly secure plaintext headers and readily encrypted payload. Second, an energy-efficient and indistinguishable modulation unification technique based on trellis-coded modulation (TCM) is used to obfuscate the payload's modulation scheme and partially decorrelate the modulated-frame duration from the payload size. Third, a message embedding technique is applied to overlay a frame-specific PHY-layer sender identifier on the frame preamble, obviating the need for MAC address and facilitating session-key lookup at PHY layer. We showed theoretically and experimentally that such an identifier that is constructed using a series of shifted Barker sequences and is superposed it on the 802.11b preamble can be reliably detected at the receiver without considerably affecting typical preamble functions. The simulation and experimental results also verify that modulation unification and encryption are successful in hiding the true packet size, modulation scheme, and frame content without degrading the BER performance.

CHAPTER 6

Exploiting Frame Preamble to Modulate User-Information Bits in OFDM-based 802.11 Systems

6.1 Introduction

Wireless systems continue to boost their transmission speeds and enhance their security. Among others, PHY layer techniques, including multi-user MIMO and beamforming (e.g., [78,79]) for higher payload's transmission rate and artificial noise (e.g., [80]) for improved transmission security, have shown great promise. However, in these wireless systems, the preamble is never utilized for conveying information bits at the PHY layer; a feature that can facilitate new functionalities related to enhanced performance and/or security. In a wireless system, the frame preamble is a special signal prepended to the (modulated) PHY-layer header at the transmitter (Tx) and is used by the receiver (Rx) to perform several PHY-layer functions. These functions include frame detection, FO estimation, CSI estimation, dynamic range estimation (used for automatic gain control (AGC) convergence), etc. In this chapter, we consider the frame preamble in OFDM-based WiFi systems. The duration of this preamble in 802.11a [1] is 16 μs and can be up to 52 μs in 802.11n/ac MIMO systems [24, 79], irrespective of the payload size and transmission rate. So the preamble can take 0.5-10% of a frame duration depending on the number of bits and the transmission rate of the payload. Because it is not designed to carry any user-payload bits, the frame preamble in current systems is under-utilized.

In Section 5.5, we considered the preamble in non-OFDM (single carrier) systems for message embedding. In this chapter, however, we focus on the preamble of widespread OFDM-based IEEE 802.11 systems (e.g., .11n/ac) and study the feasibility of carrying a sequence of random bits using this preamble. For the first time, we exploit the potential of this preamble in modulating a bit sequence, while preserving its basic properties (legacy receivers are still operate as normal). In OFDM-based 802.11 systems, the Rx needs to know the preamble "structure" but does not need to know the exact signal of the whole preamble. We exploit this feature to construct several new but legitimate preamble waveforms. Each waveform can represent the modulated version of a bit sequence.

To generate the legitimate preamble waveforms, we design a novel modulation technique called *preamble modulation* (*P-modulation*). *P-modulation* is a combination of two independent signal processing techniques, time shift in the time domain and phase rotation in the frequency domain. These techniques preserve the key characteristics of a standard preamble. At the Rx and after applying channel equalization to the received preamble signal, *P-modulation* exploits the particular dependency pattern among the subcarriers of legitimate preamble waveforms as well as the mandatory repetitions in the preamble to efficiently distinguish between different waveforms and accordingly demodulate the embedded information bits. Furthermore, *P-modulation* employs a fine-scale synchronization technique to account for the sensitivity of its demodulator to timing errors. For the same level of robustness to channel/device impairments as BPSK modulation, *P-modulation* can embed up to 8 and 19 bits in the preamble for channel bandwidth of 20 MHz and 80 MHz (e.g., 802.11ac), respectively. More bits can be embedded if *P-modulation* is contrasted to higher-order modulation schemes.

The rest of the chapter is organized as follows. In Section 6.2, we motivate the significance of *P-modulation* by discussing various potential applications. In Section 6.3, we provide background on preamble functions and the requirements that must be met in a legitimate OFDM-based WiFi systems preamble. We then introduce *P-modulation* and its related modulation/demodulation techniques in Section 6.4. The robustness of this scheme to channel/device impairments and its possible extensions to MIMO systems are discussed in Sections 6.5 and 6.6, respectively. Finally, we present the results of extensive simulations and USRP experiments in Section 6.7. The chapter is concluded in Section 6.8.

6.2 Applications of P-modulation

Embedding/modulating information in the frame preamble opens the door for several important applications. We divide these applications into two categories: Security and PHY-layer signaling.

1) *Security–* The bit sequence embedded in the preamble can be used to represent a (time-varying) PHY-layer sender identifier, hence, facilitating full PHY-level frame encryption and preventing MAC spoofing attacks. It can also represent the sender's time-varying digital signature (used to authenticate a link and prevent copycat/replay attacks [81]) or the initialization vector for generating secret pilot subcarriers locations (to mitigate pilot tone jamming attacks [82, 83]). As an example of a use case, we now discuss using the embedded information for PHY-layer sender identification.

Sender identification is a key functionality in any wireless network. It allows the Rx to distinguish between different transmitting nodes. Moreover, if nodes employ full-frame encryption for data confidentiality, sender identification is required at the Rx before the decryption can take place so as to look up the right decryption key. In this case, the Rx needs to receive a plaintext sender identifier at some header in the protocol stack before it can start the decryption process. For example, in IEEE WLAN standards, the globally unique MAC address at the link layer acts as the sender identifier. However, a link-layer identifier is extracted only after decoding the PHY-layer header. Thus, if the full frame is to be encrypted (including the PHY header), the MAC address cannot be used to look up the decryption key. In other words, the PHY-layer header cannot be decrypted until the MAC address has been obtained, creating a deadlock at the Rx. As a result, relying on a link-layer identifier necessitates transmitting the PHY-layer header in the clear. However, it has been shown that an adversary can fingerprint a device/user [10] using PHY-layer header and can apply traffic analysis to disclose several private information, even if the frame payload is encrypted [10, 16, 84]. Transmitting the PHY-layer header in the clear can also be exploited to launch various selective jamming attacks (e.g., [19]).

Furthermore, a plaintext and predictable MAC identifier opens the door for MAC spoofing and/or unauthorized user tracking attacks. The case of trash cans in London a couple of years ago is an example of such attacks [4]. The trash can suppliers had installed a device in the cans to collect information from smartphones of people walking in London's Square Mall, mostly based on PHY-layer header fields. The intention was to study customers' shopping habits and generate targeted advertisements. The seriousness of these tracking attacks has been recently acknowledged by IEEE and IETF, and accordingly, a new study group was formed to assess the privacy implications of visible MAC addresses and other link-layer privacy issues [85]. To provide a more secure link-layer sender identification approach and prevent user tracking, this group suggested using random MAC addresses, generated based on a chain of unpredictable but unencrypted time-rolling identifiers (e.g., [6–8, 86]). As discussed earlier, however, link-layer identification prevents full-frame encryption. In addition, MAC address randomization implementations on commercial devices (e.g., for hiding the true address in probe requests in Apple iOS 8.1.3) have been shown to exhibit several vulnerabilities [56]. By employing *P-modulation*, such random identifiers can instead be used at the PHY layer, facilitating full-frame encryption. Such a PHY-layer identification can replace existing inefficient PHY-layer sender identification methods (or sender authentication to defend against identity-based spoofing).

Besides sender identification, time-varying preamble waveforms that change from one frame to another can be used to counter against FO estimation attacks (e.g., [36]). These attacks exploit the publicly known preamble signal to craft a jamming signal that efficiently disrupts the FO estimation process at the Rx.

2) *PHY-layer Signaling– P-modulation* can be used as a signaling mechanism for certain PHY-layer operations, which otherwise requires modifying existing header structures and introducing new fields. For example, the embedded bit sequence can be used to convey the operation mode of the Tx in full-duplex communications (e.g., transmit/receive vs. transmit/sense), the (time-varying) pattern of *traveling pilots* proposed for the upcoming IEEE 802.11ah standard to improve channel estimation under high Doppler scenarios [87], the modulation scheme of the PHY-layer header to enable the Tx to use a higher transmission rate for this header, or any PHY-layer field required for future applications that cannot be conveyed in the standard PHY-layer header. Alternatively, the embedded bits can be a part of the PHY header, merged into the preamble and removed from the frame (to reduce the frame duration). Such an approach increases the utilization of the frame by communicating a part of it through the preamble.

6.3 Preliminaries – STF Requirements

Before introducing *P-modulation*, we first explain the key operations and special characteristics based on which the preamble of an OFDM-based IEEE 802.11 frame is designed. The preamble consists of two fields [1,24]: a *short training field* (STF), which consists of 10 repetitions of some periodic *short training signal* (STS), and a *long training field* (LTF), which includes two repetitions of another periodic long training signal (see Figure 6.1 for the preamble of a 20 MHz channel). The signal in the STF is constructed by transmitting only on subcarriers that are four subcarrier-spacings apart from each other, resulting in a larger greatest common divisor (gcd) of the subcarrier frequencies than the gcd of the LTF subcarrier frequencies and so a shorter time period. The preambles in 802.11n and 802.11ac (MIMO) standards

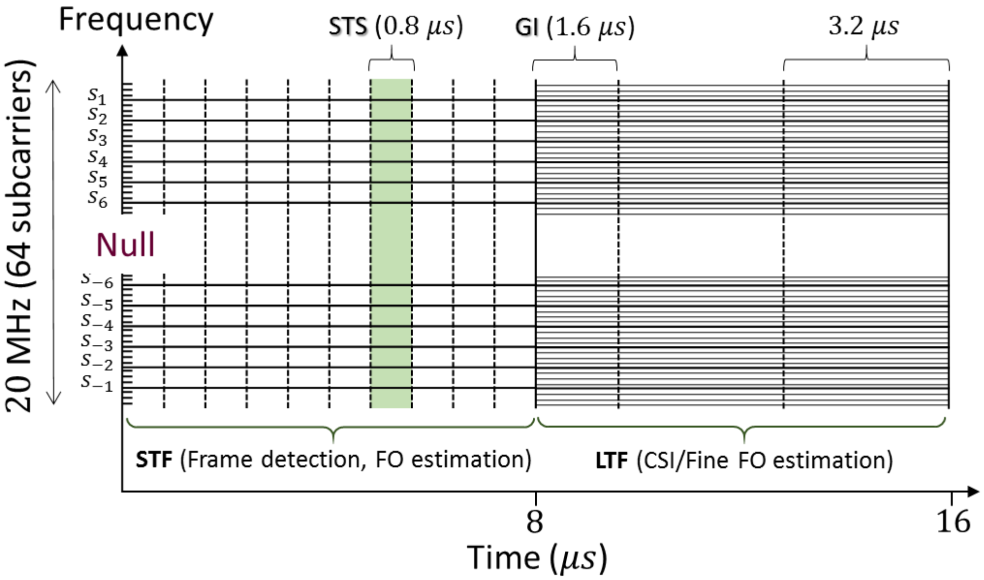


Figure 6.1: Preamble structure and subcarriers in 802.11a. s_i refers to the ith subcarrier in the STF, $i \in \{-6, \ldots, 6\}/\{0\}$.

are similar to 802.11a but are transmitted over a wider bandwidth (up to 160 MHz). They may also include an additional STF for better AGC and multiple LTFs for channel sounding and backward compatibility. After performing STF functions (described below), LTS is used for CSI and fine FO estimation; hence, it must be fully known to the Rx. On the other hand, the STF does not need to be fully known to the Rx, as will be discussed next.

6.3.1 STF Functions

The STF is used for frame detection, coarse FO estimation, AGC, diversity selection, and other functions. Accurate frame detection and FO estimation are two key operations that require two identical signals in the STF (e.g., two STSs). Having identical signals enables the Rx to detect the frame without knowing the channel.

Let $h(t)$ represent the channel between the Tx and the Rx, and let $x(t)$ be the transmitted signal at time t. (The noise term is excluded in our discussion to simplify the presentation.) In autocorrelation-based frame detection, the Rx needs only to know the period, denoted by λ_S, of the STF signal to perform frame detection [26]. It searches for identical STSs by correlating two parts of the same received signal $h(t)x(t)$ that are separated by a time lag that equals to a multiple of λ_S. Let $\mathcal{A}_\tau(t)$ be the autocorrelation value at time t and time lag τ:

$$\mathcal{A}_\tau(t) = \sum_{n=0}^{\tau-1} \frac{h^*(t+n)\, h(t+\tau+n)\, x^*(t+n)\, x(t+\tau+n)}{\sum_{q=0}^{\tau-1} |h(t+\tau+q)\, x(t+\tau+q)|^2} \tag{6.1}$$

where the time unit of the integer values λ_S, τ, and n is the sampling period at the Rx. The time at which $\mathcal{A}_\tau$ peaks is taken as the start of the frame.[1] Assuming the channel does not change during the STF, this method is channel-independent. It is adopted in practical systems.

The other key function of the STF is FO estimation. FO refers to the inherent difference in the operating frequencies of two oscillators. It creates a time-varying phase offset in the received signal, and can significantly harm the performance of an OFDM system [26]. To estimate the FO after determining the start of the STF, the Rx considers samples that belong to two consecutive STSs and measures the phase difference between the corresponding samples in these two STSs. The average of the phase differences captures the FO between the two devices.

6.3.2 STF Requirements

The STF in current IEEE standards (as defined in [1]) is designed to satisfy certain requirements related to the preamble functions. Any modification in this design should take these requirements into account. In the following, we discuss these requirements and explain how the current STF design satisfies them.

[1] To account for noise, in practical systems the first time that $\mathcal{A}_\tau$ exceeds a near-to-maximum threshold is taken as the frame start time. If this instance is a few samples before the true start time, the Rx can use the known LTF signal to fine-tune the estimation.

Periodicity

In 802.11a, an OFDM symbol consists of 64 orthogonal frequency subcarriers (up to 256 subcarriers in 802.11n and 802.11ac). Without considering the guard interval, the symbol lasts for 3.2 μs. Twelve subcarriers are not used (see the null subcarriers in Figure 6.1). The period λ_S of an STS is set to $3.2/4 = 0.8\,\mu$s by using only 12 equally-spaced subcarriers out of 52 active ones. This enables the Rx to cover a wide range of FOs, up to 625 kHz [88]. In addition, autocorrelation-based frame detection and FO estimation rely on identical repetitions of the same STS. So all the ten STSs should be identical.

Let $\mathcal{S} = [s_{-6}, \ldots, s_{-1}, s_1, \ldots, s_6]$ be the symbol sequence carried in these 12 subcarriers. Let $P(t)$ be the value of the STF signal at time t. Then,

$$P(t) = \sum_{k=-6, k\neq 0}^{6} s_k e^{2\pi j (4k) \Delta_f t}, \text{ for } t \in [0, 8\,\mu s] \tag{6.2}$$

where Δ_f is the frequency spacing between any two subcarriers in the original 64 subcarriers.

s_i	s_{-6}	s_{-5}	s_{-4}	s_{-3}	s_{-2}	s_{-1}	s_1	s_2	s_3	s_4	s_5	s_6
Value	$1+j$	$-1-j$	$1+j$	$-1-j$	$-1-j$	$1+j$	$-1-j$	$-1-j$	$1+j$	$1+j$	$1+j$	$1+j$

Table 6.1: Sequence of QPSK-modulated symbols used to generate STF in 802.11a/g [1]. $|s_{-6}| = |s_{-5}| = \ldots = |s_6| = \sqrt{2}$. This sequence is then multiplied by $\sqrt{13/6}$ to normalize the average power of the resulting symbols.

Peak-to-average Power Ratio (PAPR)

Due to the nonlinearity of power amplifier at the Tx, the PAPR of the STF, denoted by R_{PAP}, should be as small as possible to avoid poor transmission. IEEE 802.11a/-n/ac standards use a set of 12 QPSK-modulated symbols for the sequence $\mathcal{S}$ (as shown in Table 6.1), which achieves $R_{\text{PAP}} = 2.24$ dB in 802.11a/g [88]. These symbols are multiplied by a factor of $\sqrt{13/6}$ to normalize the average power of the STF with respect to the rest of the frame [1].

Dynamic range

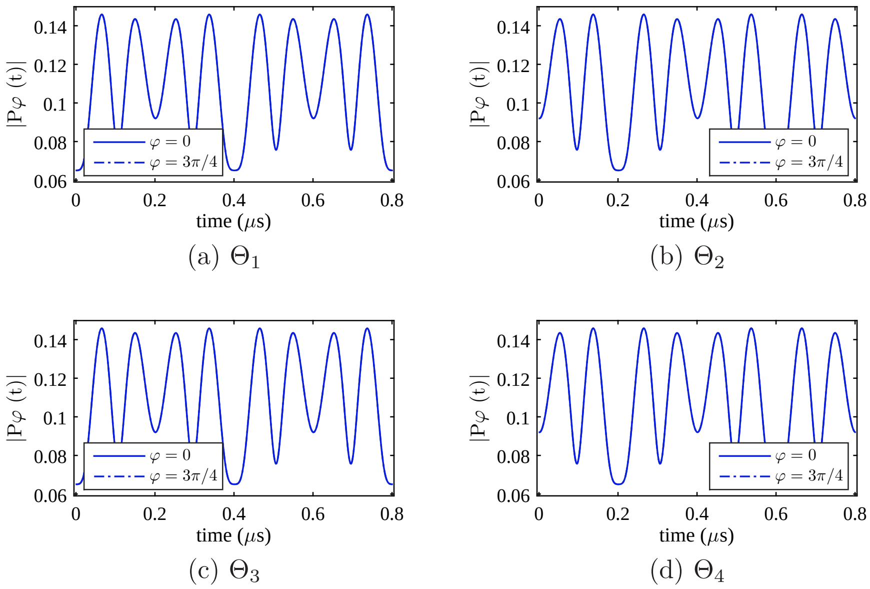
The STF is also used for AGC convergence. In order to accelerate AGC locking and adjusting the reference signal value for the A/D converter at the Rx, the whole dynamic range of the STF, denoted by R_{DR}, should be covered by the A/D converter resolution and without any overflow/underflow [88]. The R_{DR} for the symbol sequence shown in Table 6.1 is 7.01 dB, one of the lowest possible dynamic values among the ones of all symbol sequences $\mathcal{S}$ that have low R_{PAP}. In HT-mixed format of 802.11n, an additional STF is used to improve AGC estimation in MIMO systems [24].

Θ_i	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	θ_{11}	b_2b_1
$i=1$	0	0	0	π	0	π	π	0	π	π	π	00
$i=2$	$\pi/2$	$\pi/2$	$\pi/2$	$-\pi/2$	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$	01
$i=3$	π	π	π	0	π	π	0	π	0	0	0	11
$i=4$	$-\pi/2$	$-\pi/2$	$-\pi/2$	$\pi/2$	$-\pi/2$	0	$\pi/2$	$-\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	10

Table 6.2: Dependency patterns among all possible combinations of QPSK-modulated symbols in $\mathcal{S}$ that satisfy the STF requirements. The IEEE standard uses the dependency pattern Θ_3 and $\varphi = 0$ when $s_{-6} = 1 + j$.

6.4 Proposed Preamble-Modulation Scheme

We now introduce *P-modulation* and show how it modulates (embeds) a bit sequence in the STF of the preamble at the Tx and then demodulates (extracts) this sequence at the Rx with low complexity and very little impact on normal preamble functions (for backward compatibility with legacy receivers).



(a) Θ_1

(b) Θ_2

(c) Θ_3

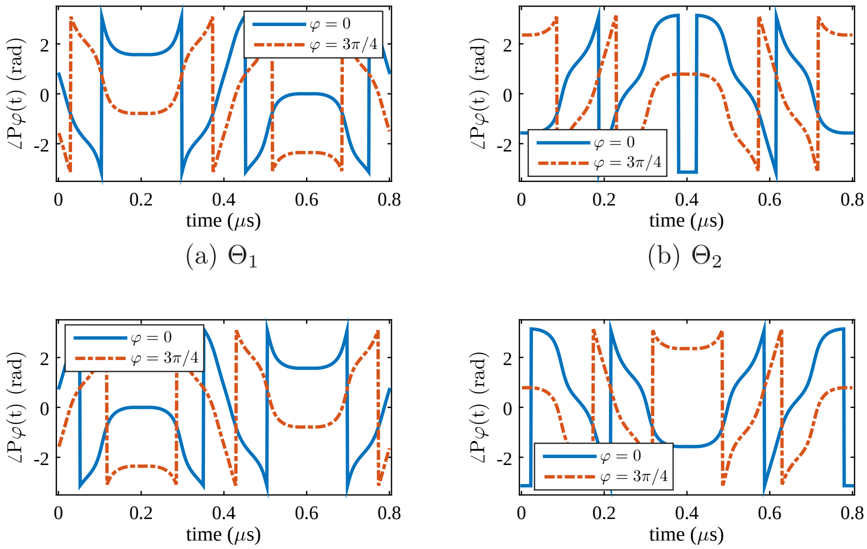
(d) Θ_4

Figure 6.2: Amplitudes of the four STFs generated using the patterns in Table 6.2. (Only one STS is shown.)

6.4.1 Sequence Modulation

In our design, modulating a bit sequence is different from conventional digital modulation, where every element in the sequence of modulated symbols is determined independently by one or multiple input bits. Instead, we use a particular “chain” of correlated symbols to modulate a bit sequence. The correlation among symbols in $\mathcal{S}$ is imposed by the R_{PAP} and R_{DR} requirements stated above. In the following, we say a signal is STF-compliant if it satisfies all the requirements in Section 4.2. The number of distinct STF-compliant signals determines the number of different bit sequences that can be embedded in the STF.

To construct a set of STF-compliant sequences, we first identify different dependency patterns among the symbols in $\mathcal{S}$ of known compliant signals. Using those patterns, we then employ two signal processing techniques to generate as many compliant signals as possible. In here, a dependency pattern is defined as the sequence



(a) Θ_1 (b) Θ_2

(c) Θ_3 (d) Θ_4

Figure 6.3: Phases of the four STFs generated using the patterns in Table 6.2. (Only one STS is shown.)

of (wrapped) phase differences between the successive symbols in $\mathcal{S}$. Let θ_i represent the ith phase difference and let $\Theta = (\theta_1, \ldots, \theta_{11})$ represent a dependency pattern in $\mathcal{S}$ starting from s_{-6}. For example, $\theta_1 = \measuredangle(s_{-5}) - \measuredangle(s_{-6})$, where $\measuredangle(.)$ indicates the phase of a complex symbol. Therefore, a set $\mathcal{S}$ can be alternatively represented using its s_{-6} and associated Θ, as follows:

$$s_i = e^{j\theta_{i+6}} \times s_{i-1}, i = -5, \ldots, 6, i \neq 0 \tag{6.3}$$

Through exhaustive search among all 4^{12} sequences that only consist of QPSK symbols, we identified 16 STF-compliant signals. Each of the QPSK symbols appears as the value of s_{-6} in four of these signals. We further recognize four distinct dependency patterns for each s_{-6} value: $\Theta_1, \ldots, \Theta_4$, each corresponding to one of the four possible values for θ_1 (see Table 6.2). For example, the dependency pattern of the sequence in Table 6.1 is Θ_3. (As will be discussed shortly, θ_i's also depend on

θ_1.)

To design more compliant signals using the same dependency patterns Θ_i, $i = 1, \ldots, 4$, we exploit the fact that if the symbols transmitted on the subcarriers of an OFDM symbol are all shifted in phase by the *same* amount, then the period, the R_{PAP}, and the R_{DR} of that OFDM symbol do not change. Hence, we can use higher-order PSK symbols as s_{-6}. To illustrate, let φ be the phase shift of the elements in $\mathcal{S}$ and $P_{\varphi}(t)$ be the new STF after this shift. Then,

$$P_{\varphi}(t) = e^{j\varphi} P(t). \tag{6.4}$$

Multiplying a signal by a constant coefficient does not change the ratio of the maximum and minimum amplitude of the signal (i.e., the R_{DR}) or the ratio of the maximum and the root-mean-square of the signal (i.e., the R_{PAP}). So the Tx can select any phase for s_{-6} and the same amplitude of $\sqrt{2}$, follow one of the patterns Θ_i, $i = 1, \ldots, 4$, to define the rest of the symbols in $\mathcal{S}$, and generate a STF-compliant signal.

In Figure 6.2, we show the amplitudes of the STF-compliant signals, constructed by using two different values for φ and one of the four dependency patterns. These figures also show that $P_{\varphi}(t)$ and $P(t)$ with the same dependency pattern have the same envelope. Hence, amplitude-based STF functions (e.g., frame detection and FO estimation) will not be impacted by the phase shift. The specific selection of the dependency pattern Θ_3 in the IEEE 802.11 standards is with respect to cross-correlation-based detection issues (e.g., matched filter performance in the boundary region between the STF and the LTF [89]). However, by using the autocorrelation method for frame detection at the Rx, those issues will not be binding for our scheme.

The coefficient $e^{j\varphi}$ rotates the constellation map of the symbols in $\mathcal{S}$ by φ degrees. Therefore, the set of s_{-6} values that can be used to generate STF-compliant signals consists of the symbols of a PSK modulation scheme. The order of this modulation scheme, denoted by M, depends on the performance of the PSK demodulation operation and the accuracy of pattern detection (discussed in Section 6.4.2), as well as the accuracy of CSI and FO estimation (discussed in Section 6.5). The order specifies how many bits can be modulated using different constellation rotations when using the same pattern. We refer to these $\log_2 M$ bits as *rotation bits*. The Rx can exploit the correlation among the symbols in $\mathcal{S}$ and use all of them to improve the demodulation accuracy. However, incorrect detection of the underlying pattern may significantly change the expected sequence of phases. This can be seen in Figure 6.3, where we plot the phases of the signals in Figure 6.2.

A closer look at the patterns in Table 6.2 reveals that they are indeed not independent and a pattern Θ_i can be calculated by adding a multiple of $\pi/2$ to Θ_1 (for θ_6 a multiple of π should be added). For example, $\Theta_2 = \Theta_1 + \pi/2$. In other words, when the phase differences between successive symbols of the frequency subcarriers are changed by a constant, the resulting signal remains a STF-compliant signal. This property in the frequency domain has an interpretation in the time domain: In OFDM systems, a time shift in the signal results in a *linear* phase shift in the symbols along the ordered subcarriers [90]. Because the elements in $\mathcal{S}$ are ordered by their frequencies, such a linear shift brings about the same change in phase differences θ_i, $i = 1, \ldots, 11$. This amount of change is indeed the slope of a line that defines a symbol's phase shift with respect to its frequency. Let N be the number of samples in a symbol and ν be the line slope. The corresponding time shift t_s (in terms of the number of samples) is $t_s = N \times \nu/2\pi$. For example, the dependency patterns in Table 6.2 represent different amounts of time shift of the same signal, as shown in Figure 6.2. Note that a time shift in a periodic signal does not change its dynamic range and PAPR values. Therefore, we can generate other sets of STF-compliant signals by shifting one compliant signal in time, or equivalently, using different dependency patterns $\Theta^{(\nu)} = \Theta_1 + \nu, \nu \in [-\pi, \pi]$. For example,

Θ_2 in Table 6.2 can be represented by $\Theta^{(\pi/2)}$. A time shift, however, impacts the frame detection accuracy because the last few samples in the STF can have higher amplitudes than the ones in the standard STF. This will inflate the noise samples located before the true start of the frame during the frame detection when the autocorrelation window includes a few of them. We exploit the LTF to account for such errors.

We note that using different dependency patterns with the same φ is equivalent to a form of frequency-domain differential PSK (FD-DPSK), which is robust to channel phasor and FO estimation errors (i.e., a non-coherent scheme). The number of different dependency patterns in our scheme, denoted by Q, depends on the target performance of the demodulator (discussed in Section 6.4.2) and the frame detection accuracy (discussed in Section 6.5). With the same φ, we can encode $\log_2 Q$ bits. We show an example in Table 6.2 for $Q = 4$. In *P-modulation*, the assignment of bits to patterns and phase shifts follows the Gray Coding rule.

Altogether, the Tx can embed the total of $\log_2 M + \log_2 Q$ bits in the STF by using the proposed time shift and phase rotation techniques simultaneously. So, the bit sequence will be as follows:

$$\big[\underbrace{b_{\log MQ}, \ldots, b_{1+\log Q}}_{\varphi}\ \underbrace{b_{\log Q}, \ldots\ b_1}_{\nu}\big] \tag{6.5}$$

6.4.2 Sequence Demodulation

The process of message extraction involves detecting the most probable symbols sequence carried on the 12 subcarriers of the STF. In contrast to digital demodulation, in which the most probable symbols are independently identified to obtain the bit sequence, in here a sequence of symbols should be detected simultaneously.

To demodulate and extract the embedded bit sequence, the Rx exploits the correlation among the symbols that belong to the same STS as well as the repetition of the STSs. It needs a small memory to first store the 10×16 received STF samples.

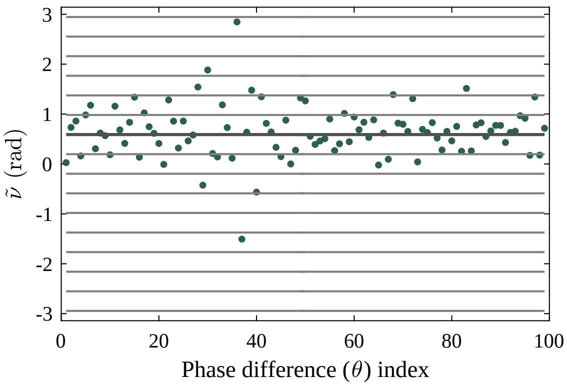


Figure 6.4: 99 phase differences among $12 \times 9 = 108$ symbols extracted from 9 STSs. The Tx uses 16-DPSK and the Rx detects $\tilde{\nu} = 3\pi/16$ using MMSE estimator ($\gamma = 3\,\text{dB}$).

After compensating for the FO and equalizing the effect of the LTF-based estimated CSI on these samples, it applies the 16-point discrete Fourier transform (DFT) on each of the STSs and obtain the symbols on each subcarrier. Then, it estimates ν. Let $\tilde{\mathcal{S}}_l = [\tilde{s}_{l,-6}, \ldots, \tilde{s}_{l,-1}, \tilde{s}_{l,1}, \ldots, \tilde{s}_{l,6}], l \in \{1, \ldots, 9\}$[2] be the sequence received on the lth STS and $\tilde{\nu}$ be the most probable estimate of ν based on $\tilde{\mathcal{S}}$. Because the elements in $\mathcal{S}$ are correlated, we use minimum mean-square error (MMSE) to estimate $\tilde{\nu}$ with respect to Θ_1. An example drawn from our experiments is shown in Figure 6.4. Next, to find an estimate of φ, denoted by $\tilde{\varphi}$, the Rx subtracts the phases of the reference elements constructed according to $\Theta^{(\tilde{\nu})}$ from the corresponding elements in $\tilde{\mathcal{S}}_l$, $l = 1, \ldots, 9$. To account for noise when measuring the phase, the Rx calculates the sum of all $12l$ elements. The estimation of ν requires accurate frame detection while the estimation of φ needs accurate FO and CSI estimation. After FO correction and channel equalization, the bit sequence is fully demodulated based on $\tilde{\nu}$ and $\tilde{\varphi}$.

To study the limit on the order of the PSK modulation used for constellation rotation (the minimum phase by which a sequence is shifted) and the number of

[2] We cannot rely on all ten STSs because the last one may not be the same as other STSs due to pulse shaping in its boundary with the LTF.

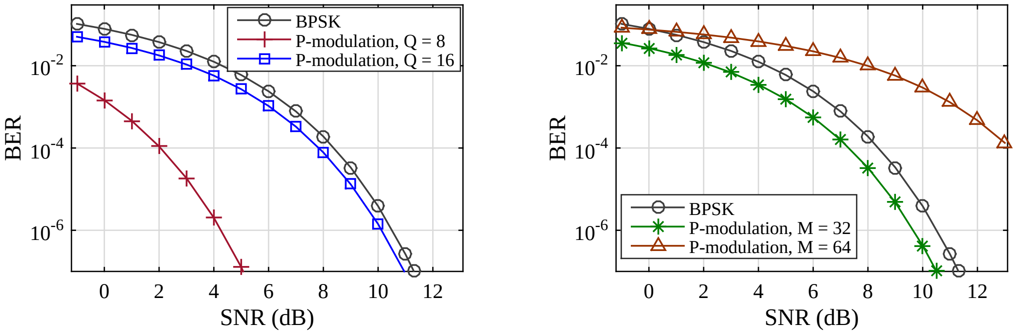
different time shifts Q (equivalently, the number of different Q-DPSK patterns), we target a performance for our scheme that is comparable to the performance of the most reliable modulation scheme supported by the system, which is often used for transmitting the PHY header. This will guarantee that whenever the PHY header can be successfully decoded, the embedded bit sequence can also be extracted successfully. In 802.11, BPSK is the most reliable modulation scheme.

The performance of demodulating a sequences of modulated symbols depends on the signal-to-noise ratio (SNR), FO and CSI estimation accuracy, and frame detection accuracy. In this section, we assume an AWGN channel model as well as accurate FO and frame detection to obtain the M and Q that result in the same BER as BPSK. Let $B_2(\gamma)$ be the BER of 2-PSK (BPSK). Then, $B_2(\gamma) = \mathcal{Q}\left(\sqrt{2\gamma}\right)$, where $\mathcal{Q}(.)$ is the Q-function and γ denotes the SNR. The Rx first estimates $\Theta^{(\nu)}$ followed by estimating φ.

Pattern Detection

The pattern $\Theta^{(\nu)}$, modulated using FD-DPSK, is the base for the estimation of the other parameter φ. Let the BER of frequency-domain Q-DPSK modulation be $B_Q^D(\gamma)$. When the channel is flat-flat fading, the BER performance of DPSK is the same, irrespective of whether it is time-domain or frequency domain [91]. However, FD-DPSK performs worse than the time-domain DPSK when the channel is frequency-selective/time-dispersive [91]. The expression of the BER under an AWGN channel is provided in Equation 8.86 of [92], but it does not have a closed-form. Instead, we use the MATLAB function *berawgn*, which uses a very close expression to the one in [92], to numerically find the maximum Q that satisfies $B_Q^D(\gamma) \leq B_2(\gamma)$. (DPSK modulation scheme loses 3 dB in gain compared to PSK due to the subtraction of two random variables.)

The amplitude of a symbol in the STF is $\sqrt{13/3}$ times the one of a BPSK-modulated symbol in the frame payload. In addition, $l = 9$ STFs, which corresponds



(a) Q-DPSK

(b) M-PSK

Figure 6.5: BER of BPSK and *P-modulation* with different Q-DPSK and M-PSK schemes vs. γ.

to 2.25 times the duration of an OFDM symbol in the frame payload. However, because every two adjacent θ_i's are negatively correlated (an increase in one results in a comparable decrease in the other, see Figure 6.4), using 11 DPSK symbols gives only 5.5 SNR gain. Overall, the SNR is improved by $5.5 \times 2.25 \times 13/3 = 275.5/3$. In Figure 6.5(a), we depict $B_8^D(.)$ and $B_{16}^D(.)$ compared to $B_2(.)$. Even though $B_{16}^D(\gamma)$ is slightly less than $B_2(\gamma)$, the DFT error contributes to more BER when $Q = 16$, as will be shown in Section 6.7. Therefore, we set $Q = 8$ for an AWGN channel. In the experiments, however, we use $Q = 16$ because of the robustness of DPSK to FO and CSI estimation errors.

Phase Detection

After $\Theta^{(\tilde{\nu})}$ estimation, FO correction, and channel equalization, the Rx estimates $\tilde{\varphi}$ using $12l$ symbols. Although one can arbitrarily increase the modulation order and reduce the distance between the PSK symbols without violating the STF requirements, we intend to maintain the BER less than that of BPSK. Let the BER of an M-PSK modulation scheme be $B_M(\gamma)$. We want to find the maximum M such that $B_M(\gamma) \leq B_2(\gamma)$.

Similar to the previous case, the amplitude of a PSK-modulated symbol in the STF is $\sqrt{13/3}$ times the one of a BPSK-modulated symbol. The summation of 12 i.i.d random variables with Gaussian noise improves the SNR by 12. The noise variance is further decreased by using $l = 9$ repetitions of the same STS. Because the duration (power) of nine STSs is 2.25 times the duration (power) of a BPSK-modulated symbol in the preamble, the overall SNR improvement will be $12 \times 2.25 \times 13/3 = 351/3$. An approximation of the BER for M-PSK is given by:

$$B_M(\gamma) = \frac{2}{\log_2 M} \mathcal{Q}\left(\sqrt{2\gamma} \sin \pi/M\right). \tag{6.6}$$

Numerically solving for the maximum M that satisfies

$$\frac{2}{\log_2 M} \mathcal{Q}\left(\sqrt{2 \times 351 \times \gamma/3} \sin \pi/M\right) \leq \mathcal{Q}\left(\sqrt{2\gamma}\right), \tag{6.7}$$

we obtain $M = 2^5 = 32 \forall \gamma \in \mathbb{R}$, and $M = 2^6 = 64 \forall \gamma \leq 0.2\,\mathrm{dB}$ (see Figure 6.5(b)).

6.4.3 Noncompliant but Possible STF Waveforms

In the scheme above, all the STF requirements, including $R_{\text{PAP}} \leq 2.24\,\mathrm{dB}$ and $R_{DR} \leq 7.01\,\mathrm{dB}$, are met. However, modern wireless devices are capable of processing signals with higher R_{PAP} and R_{DR} values. For example, COTS wireless routers usually support $R_{DR} > 100\,\mathrm{dB}$. This paves the way to expand the set Θ and include several new patterns by identifying STF waveforms whose R_{PAP} and R_{DR} values are close to the ideal values, i.e., almost-compliant sequences. For example, if we allow R_{PAP} to increase to 2.95 dB, two new independent sets of patterns will become available in addition to the set $\Theta^{(\nu)}$ defined above.

Moreover, by expanding the search space for s_i's of STF-compliant signals beyond QPSK, one may find other types of dependency patterns. Considering 8-PSK, for example, we identified at least one new pattern (in addition to the patterns that define the sequences with only QPSK symbols). For this pattern, $R_{\text{PAP}} \leq 2.27\,\mathrm{dB}$

and $R_{DR} \leq 12\,\text{dB}$, which are close to the target ideal values.

6.5 Effects of Channel/Device Impairments on P-modulation

The sequence demodulation process explained above is impacted by the complex channel coefficient h, the residual FO estimation, and the frame detection errors. In this section, we study the effect of each of these parameters and discuss the robustness of time shift and constellation rotation techniques against them. Let δ_f and t_E represent the FO (normalized to Δ_f) and the timing error (in number of samples), respectively.

6.5.1 Tx-Rx Channel Coefficients

First, we consider the effect of the channel on the dependency pattern $\Theta^{(\nu)}$. Let $h_{l,k}, k = -6, \dots, 6 (k \neq 0), l = 2, \dots, 9$ be the channel coefficient on each STF subcarrier during the lth STS. Assuming $\delta_f = 0$ and $t_E = 0$, the symbol s_k on the lth STS at the Rx, denoted by $\tilde{s}_{l,k}$, will be:

$$\tilde{s}_{l,k} = h_{l,k} s_k + n_k. \tag{6.8}$$

where n_k is the noise component.

To detect the dependency pattern, the Rx calculates the phase difference between each pair of adjacent subcarriers. If the channel's coherence bandwidth is larger than the signal's bandwidth, hence the $h_{l,k}$'s will have the same phase value, the channel does not impact the phase differences θ_i. On the other hand, if the coherence bandwidth is very small and the phases of $h_{l,k}$'s are uncorrelated, the variations will be averaged out when MMSE is used. In the case of neither small nor large coherence bandwidth, the Rx treats each period of coherence bandwidth independently. For each set of subcarriers with the same $h_{l,k}$'s, the Rx computes θ_i's separately. The same argument can be made for multi-path channels. Furthermore, because the proposed demodulator treats STSs independently, θ_i's are not affected if the channel is fast fading/time-selective (provided that the coherence time is larger than λ_S).

Similarly, when $\delta_f \neq 0$, the subcarriers that belong to the same STS will experience the same amount of phase offset and so residual FOs do not impact the decoding performance. However, FD-DPSK is susceptible to timing errors. We remedy that by taking advantage of the LTF (see below).

Although the channel has little impact on $\tilde{\nu}$, it may significantly change the phase of the PSK symbols in $\mathcal{S}$ by a constant, i.e., the channel phasor. So the Rx needs to estimate the channel and equalize it before estimating φ.

6.5.2 Frame Detection Accuracy

A time offset in the time domain manifests itself as a linear phase shift in the frequency domain [38, 90]. This phase shift varies linearly with the subcarrier index and may significantly change the phase differences between successive symbols in $\mathcal{S}$. The timing error t_E will create phase shift ν_k in the kth subcarrier:

$$\nu_k = \frac{2\pi k t_E}{N}, k = -N/2, \ldots, N/2. \tag{6.9}$$

Figure 6.6 depicts an example (drawn from our experiments) of the linear change of (wrapped) estimated channel phases with the subcarrier frequency when frame detection is not accurate. A linear phase shift in the frequency domain will itself result in a constant phase change $2\pi t_E/N$ in the pattern Θ_i. For example, if $t_E = 4$, $N = 16$, and Θ_1 in Table 6.2 are used for each STS, the Rx will detect Θ_2 because $\Theta_2 = \Theta_1 + \pi/2$. Hence, correct detection of the pattern depends on accurate detection of the STF.

In *P-modulation*, we take advantage of time-domain LTF-based CSI estimation to fine-tune the frame detection. The Rx usually considers a channel estimation period of length, say L, to estimate the maximum of L multi-path channel components.

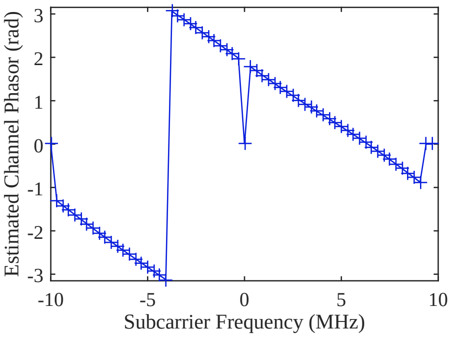


Figure 6.6: Linear channel phase changes across OFDM subcarriers ($t_E = -4$, $N = 64$, and Tx power $= -25\,\text{dBm}$).

Once the STF has been detected, the Rx constructs an L-row *Toeplitz* matrix whose first row corresponds to the known LTF and the be the remaining $L - 1$ rows are filled with shifted versions of the LTF. Using an MMSE estimator and the Toeplitz matrix, the Rx then estimates the channel coefficient of each path. Assuming that the signal coming along the first path is the strongest signal (e.g., LOS), the Rx estimates t_E by finding the highest channel coefficient amplitude among the channel coefficients.

The above method is more reliable than the fine-synchronization method in the SourceSync protocol [38], which is performed before CSI estimation. In SourceSync, the Rx uses the phase of the estimated channel coefficients $h_{l,k}$ to find $2\pi t_E/N$, i.e., the slope of the phase values ν_k with respect to the subcarrier index k. However, because *P-modulation* shifts the STF in time as part of its design, applying the synchronization method of SourceSync deteriorates both the frame detection accuracy and DPSK demodulation. In spite of that, SourceSync can be applied to further fine-tune the frame estimation after LTF-based frame detection. In this case, any (residual) timing error less than $\lambda_S/(NQ/4)$ can be estimated if the frame detection is applied on an oversampled version of the signal. Note that the minimum time shift t_s caused by the Q patterns is equal to $2\lambda_S/(NQ/4)$. When LTF-based CSI estimation uses a downsampled version of the signal, this method takes advantage of the estimated pattern $\Theta^{(\tilde{\nu})}$ and estimates the amount of extra phase shift in the STSs with respect to θ_i's of $\Theta^{(\tilde{\nu})}$. Assuming that the oversampling parameter is known, this method can unambiguously estimate the timing error using (6.9) as long as the extra phase shift is less than $\frac{2\pi t_E}{2Q}$. The Rx will compensate for this error before estimating φ.

6.5.3 Frequency Offset Estimation Error

FO at the Rx moves all the subcarriers in the frequency domain by δ_f. Moving away from the expected frequency locations of the subcarriers changes the phase and amplitude of a symbol and also creates inter-carrier interference as explained below (assuming $h = 1$ and $t_E = 0$) [93]:

$$\tilde{s}_{l,k} = s_{l,k}T_0 + \sum_{i=-6, i\neq 0,k}^{6} T_{i-k}s_{l,i} + n_k, k = -6, \cdots, 6 \tag{6.10}$$

where

$$T_k \overset{\text{def}}{=} \frac{\sin \pi(k+\delta_f)}{N \sin \frac{\pi}{N}(k+\delta_f)} \exp\left[j\pi\left(1-\frac{1}{N}\right)(k+\delta_f)\right]. \tag{6.11}$$

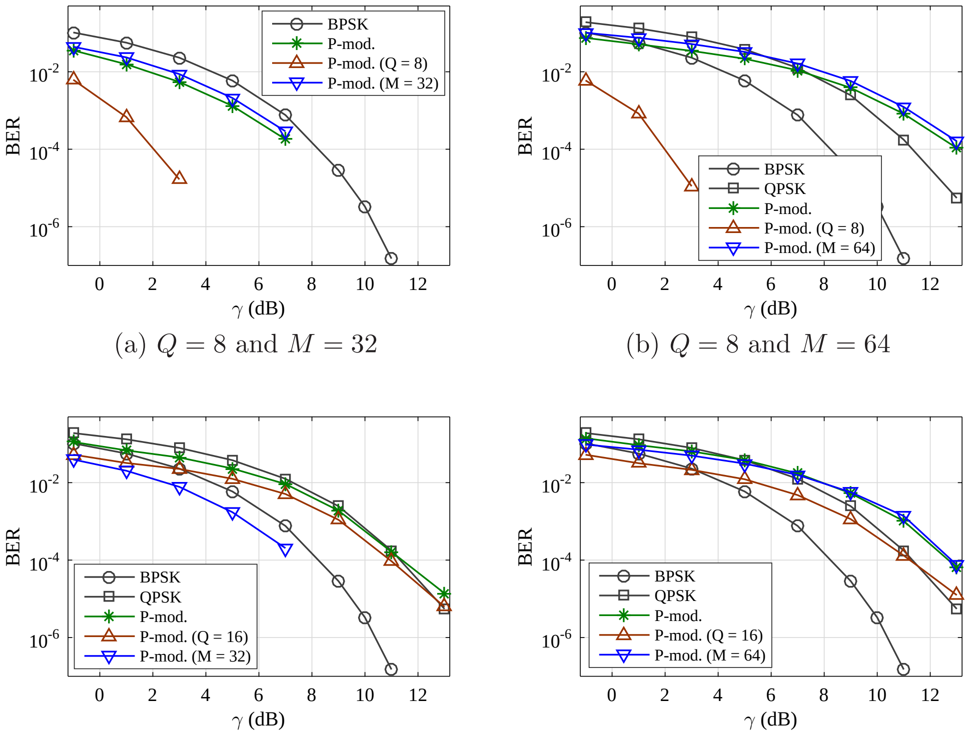
We can observe from the equations above that: (1) the phase of $s_{l,k}$ changes linearly with the subcarrier index, and (2) a weighted sum of other symbols contributes to $\tilde{s}_{l,k}$. With respect to the first observation, FO does not impact the phase difference between the symbols in $\mathcal{S}$; and hence, the underlying pattern. The linear phase change, however, affects the estimation of φ. With respect to the second observation, we note that $\delta_f \ll 1$ after both STF- and LTF-based FO corrections and hence T_k will be close to zero.

6.6 MIMO and 802.11n/ac Systems

The aforementioned STF structure with 12 subcarriers was defined for pre-802.11n OFDM systems that operate on 20 MHz channel. In 802.11n and 802.11ac systems, however, the channel bandwidth can be up to 160 MHz. With the same subcarrier spacing Δ_f defined in 802.11a, the STF in 802.11n and 802.11ac can have up to 48 [24] and 96 [79] subcarriers, respectively. The same 802.11a preamble spans 20 MHz bandwidth is duplicated and rotated by 90° for the other segments of the channel, but without increasing the length of the STF. Therefore, the length of the sequence $\mathcal{S}$ is increased proportionally, and so the SNR. Accordingly, the Tx can increase M and Q, and embed more bits in each STF waveform. For example, with 160 MHz channel, *P-modulation* can increase the embedding capacity by at least one bit using the time shift technique and another bit using the phase rotation technique (a total of two bits). Moreover, the preamble of the mandatory HT-mixed format of 802.11n for MIMO systems contains a second STF with half the length of the first one. Similar to the first STF, we can embed a bit sequence (with one bit less than the first sequence) in the second STF and almost double the length of the embedded sequence (up to 19 bits).

6.7 Performance Evaluation

In this section, we study the performance of *P-modulation* through simulations and indoor USRP experimentation. The whole scheme was implemented in LabVIEW. The Tx embeds 8-10 randomly generated bits ($Q = 8$ or 16 and $M = 16$, 32, or 64) in the STF and transmits the preamble without appending any payload. As a convention, we refer to *P-modulation* that only uses time-shifted STF waveforms (via FD-DPSK) as "P-mod. ($Q = \dots$)". Similarly, "P-mod. ($M = \dots$)" refers to *P-modulation* with phase-rotated waveforms at a given M value. Finally, "P-mod." refers to the complete scheme with given Q and M values. The metrics of interest are



(a) $Q = 8$ and $M = 32$

(b) $Q = 8$ and $M = 64$

(c) $Q = 16$ and $M = 32$

(d) $Q = 16$ and $M = 64$

Figure 6.7: BER vs. γ (simulations). For P-mod. (FD-DPSK) and P-mod. (PSK), only the value of Q and M applies, respectively.

frame detection and FO estimation accuracy as well as BER. We vary the received SNR (γ) and the residual FO estimation error (δ_f). We also implemented BPSK and QPSK modulation schemes as the benchmarks for comparison with the proposed scheme. In this case, the Tx appends 120 bytes payload (equivalent to 10 or 20 OFDM symbols) with 12.5% cyclic prefix to the preamble and transmits it. The bit sequences are all uncoded.

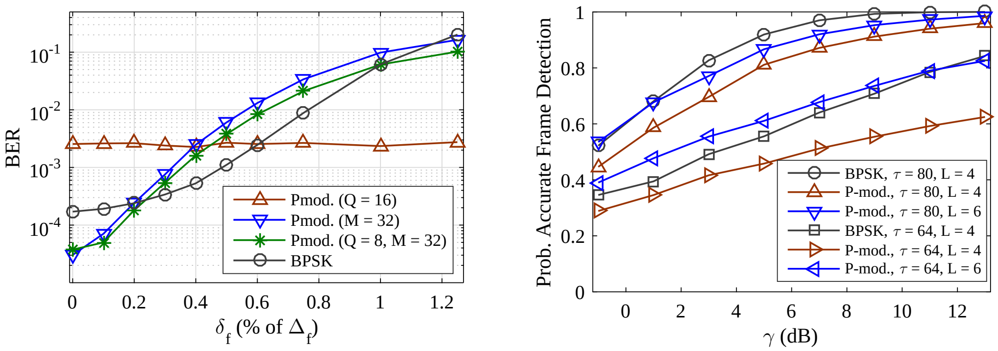
6.7.1 Simulations

First, we study *P-modulation* assuming perfect time and frequency synchronization under an AWGN channel model. In Figure 6.7, we compare the BER performance of

P-modulation with different combinations of M and Q to the performance of BPSK and QPSK. Each point is obtained by averaging the BER in 40000 iterations. When 8 bits are embedded in the STF (Figure 6.7(a)), our proposed schemes outperform the reliability of BPSK modulation scheme and confirm the analytical analysis in Section 6.4.2. However, increasing M to 64 makes the overall BER better than QPSK at low SNR (i.e., $\gamma < 7\,\text{dB}$), but it becomes worse than both BPSK and QPSK at high SNR (Figure 6.7(b)). If we instead increase Q to 16 and embed 9 bits, Figure 6.7(c) shows that the overall performance will be comparable to the one of QPSK, irrespective of the SNR value. Finally, Figure 6.7(d) shows that *P-modulation* can embed 10 bits and achieve the performance of QPSK when $\gamma < 5\,\text{dB}$.

Next, in Figure 6.8, we study the robustness of *P-modulation* to residual δ_f in decoding and to noise in frame detection. A shown in Figure 6.8(a), the proposed DPSK-based embedding scheme is independent of the residual δ_f. A given δ_f rotates the phases of the symbols in $\mathcal{S}$ by the same amount. However, such a phase change is cancelled out when calculating the phase differences. When $\delta_f > 0.6$, 16-DPSK outperforms BPSK. (8-DPSK always performs better than BPSK, but it is not shown here.) In contrast, M-PSK scheme is very sensitive to FO estimation errors. It can maintain its superiority over BPSK only when $\delta_f \leq 0.2$. In addition, the figure shows that when the Tx embeds 8 bits in the STF, the overall BER performance is better than BPSK when $\delta_f \leq 0.2$ or ≥ 1. The latter case is due to the contribution of DPSK (pattern) bits in the bit sequence. In this case, the results (not shown here due to space limitations) confirm that the FO estimation performance is not impacted by *P-modulation* due to the identical amplitudes for the samples used in the estimation.

Finally, we consider the implications of *P-modulation* on frame detection in Figure 6.8(b). The results in this figure include the cases in which the Rx uses the whole STF for frame detection $\tau = 80$ or the cases where only the first eight STSs are used ($\tau = 64$). As discussed in Section 6.4, the time shifts in the STF due to



(a) BER vs. residual δ_f

(b) Frame detection accuracy vs. γ ($M = 32$ and $Q = 16$)

Figure 6.8: Robustness of *P-modulation* to channel impairments compared to BPSK (simulations).

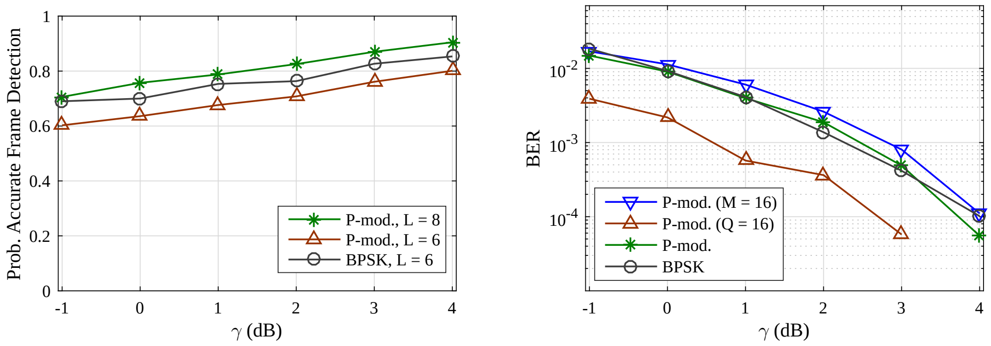
using different patterns degrade the performance of STF-based frame detection. It is mainly because the last two samples of the new STSs may have a high amplitude. Hence, when the autocorrelation window is placed one or two samples before the true start of the frame and the noise-only terms are multiplied by the last samples of one of the STSs, $\mathcal{A}_\tau$ may be very close to the $\mathcal{A}_\tau$ at the true start. We verify it by including two more time instances before the true frame start by setting ($L = 6$ vs. $L = 4$). Figure 6.8(b) depicts that the accuracy of frame detection is comparable to the one when the standard STF is used and $L = 4$. Therefore, if the Rx increments its channel estimation length by two, it can account for the implications of *P-modulation* on frame detection. We also measured the FO estimation accuracy. The results (not shown here due to space limitations) confirm that the FO estimation performance is not impacted by *P-modulation* because the set of amplitudes of the STF samples in *P-modulation* is the same as the default STF, though the order is not the same.

Altogether, the results in Figure 6.8 suggest a trade-off between the number of pattern bits ($\log_2 Q$) and the number of rotation bits ($\log_2 M$) depending on the relative accuracy of the FO estimation to the frame detection. That is, when the frame detection is good, the Tx can send more pattern bits and when the FO estimation is accurate, the Tx can send more rotation bits.

6.7.2 USRP Experiments

We also implemented *P-modulation* on a USRP testbed that consists of two NI-USRP 2922 devices, placed at a distance of 2 m from each other. The Tx and the Rx each use a 3-dBi antenna. The minimum transmission power of the Tx is $-8\,\text{dBm}$. However, because the transmission power is still very high with respect to the environment's noise ($< -80\,\text{dBm}$), we first add artificial Gaussian noise to the generated signal and then apply an artificial channel gain of $-14\,\text{dB}$ before the transmission of the signal at the Tx. In the experiments, we vary the SNR at the Tx. Furthermore, to separately study the impact of the channel and the FO estimation errors in the experiments, we use an Ettus OctoClock-G clock distributer to provide an external for the USRPs and eliminate the FO. Due to the robustness of FD-DPSK to CSI and FO estimation errors, the Tx embeds 8 bits using $Q = 16$ and $M = 16$. The signal is transmitted at 2.5 GHz band to avoid external interference in the environment. The Rx uses the channel estimation of $L = 6$ for the BPSK and $L = 8$ for *P-modulation*. Frame detection lag parameter τ is set to its maximum.

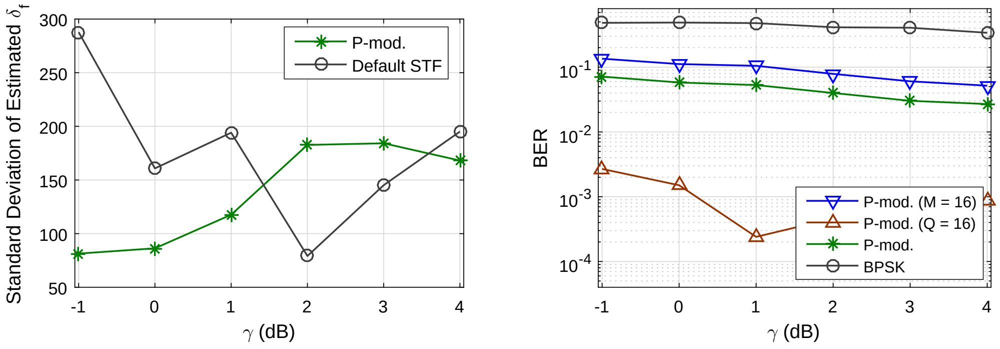
Figure 6.9 shows the performance of *P-modulation* in USRP experiments when The USRPs share a clock. Extending the channel estimation length by two samples ($\lambda_S/8\,\text{s}$) pays off and the frame detection accuracy of *P-modulation* statistically matches with the one of the default STF, as shown in Figure 6.9 (a). With respect to the BER performance, we observe in Figure 6.9 (b) that CSI estimation errors contribute to high BER of phase rotation bits even though M is set to 16. (Each point is obtained by averaging the BER in 5000 transmissions.) However, the pattern bits modulated using 16-DPSK are very robust to such errors, even more than BPSK. The low BER of the pattern bits makes the overall BER of the 8-bit *P-modulation* and BPSK comparable.



(a) Frame detection accuracy

(b) BER

Figure 6.9: Performance of *P-modulation* vs. γ when $Q = 16$, $M = 16$, and $\delta_f \approx 0$ (USRP experiments).



(a) Standard variation of estimated δ_f

(b) BER

Figure 6.10: Performance of *P-modulation* vs. γ when $Q = 16$ and $M = 16$, and under imperfect FO estimation (USRP experiments).

Finally, we disconnect the OctoClock and evaluate the BER and FO estimation performance of *P-modulation* in Figure 6.10. With respect to FO estimation accuracy, Figure 6.10(a) shows that the overall accuracy of STF- and LTF-based FO estimation in *P-modulation* in terms of the standard deviation does not have a meaningful difference compare to the default STF. Furthermore, we observe in Figure 6.10 (b) that residual FO significantly deteriorates the performance of 16-PSK, but 16-DPSK exhibits a high robustness to these errors. Nevertheless, *P-modulation* in this case has a much lower BER than BPSK.

6.8 Summary

In this chapter, we considered the STF of the preamble in the PHY layer of the IEEE OFDM systems to modulate user-information bits in order to enhance the security in the PHY layer, improve the throughput, or provide a PHY-layer signalling for new applications. To do so, we proposed *P-modulation* in which a set of new STF waveforms are constructed that comply with the requirements of the preamble of the OFDM-based IEEE WLAN systems. These new waveforms are obtained using two techniques: shift in the time domain and phase rotation in the frequency domain. We then modulated a bit sequence using these two techniques. When the robustness of BPSK modulation scheme is expected, our analysis indicates that when operating on 20 MHz bandwidth, the transmitter can embed up to 8 bits and improve the frame utilization using *P-modulation*. If the a higher bandwidth is used, higher capacity can be achieved. Our simulation and USRP experiment results confirm that practicality of *P-modulation* in real scenarios.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

7.1 Conclusions

In this dissertation, we considered the leakage of various transmission attributes in encrypted wireless communications and developed methods for hiding or obfuscating these attributes. In particular, we focused on the frame preamble and (1) revealed the vulnerability of preamble-based FO estimation mechanism in OFDM systems to stealth jamming attacks, and (2) exploited the preamble to embed information bits (e.g., transmitter identifier) and facilitate PHY-layer encryption. We also used friendly jamming signal as a random noise signal for securing communications in multi-link/path scenarios against eavesdropping, or as an modulated signal for encrypting PHY frames and obfuscating modulation schemes. Our main achievements and finding are summarized as follows.

First, we designed and experimentally demonstrated a highly disruptive but efficient-efficient DoS attack against OFDM systems. This attack succeeds even when the PHY frame is shielded by interleaving and channel coding. The attack focuses on the FO estimation process, and is channel-independent and robust to time-synchronization errors at Eve through applying the proposed pairing and chaining rules. The attack lasts for less than the duration of an OFDM symbol, i.e., less than 1% of a typical frame duration, and is at least 30% more efficient than previously reported attacks. Though short-lived, the attack results in a shift in subcarrier indices and the maximum possible BER (50%) even when the jamming signal at Bob is ~ 1.4 times weaker than Alice's signal. The simulation results show that different modulation schemes are equally susceptible if the FO attack can shift the subcarrier indices, and higher modulation orders are more affected when the attack impacts only the channel estimation. We also sketched four possible mitigation approaches.

Second, by jointly optimizing the powers and locations of the friendly jamming nodes deployed in small-scale multi-link wireless networks, we minimized the total jamming power required to secure legitimate transmissions in the presence of eavesdroppers. We then proposed two optimization strategies: per-link and network-wide (all links jointly). It was shown that our per-link scheme outperforms previous schemes in terms of energy efficiency (55–99 percent power saving). Moreover, the network-wide optimization was shown to be more energy-efficient than per-link scheme (14–38 percent additional power saving) and also requires about half the number of friendly jamming nodes than per-link optimization. For multihop scenarios, we proposed a routing metric that finds a secure path that requires minimal jamming power.

Third, we proposed *Friendly CryptoJam* (FCJ), a combination of friendly jamming and low-level encryption, to effectively protect the confidentiality of lower-layer fields and prevent SCI-based traffic classification, rate-adaptation, plaintext, dictionary, modulation detection, and device-based tracking attacks. FCJ employs three main techniques. (1) modulation-aware encryption is used to perfectly secure plaintext headers and readily encrypted payload, (2) an energy-efficient and indistinguishable modulation unification technique based on trellis-coded modulation (TCM) is used to obfuscate the payload's modulation scheme and partially decorrelate the modulated-frame duration from the payload size, and (3) a message embedding technique is applied to overlay a frame-specific PHY-layer sender identifier on the frame preamble, obviating the need for MAC address and facilitating session-key lookup at PHY layer. We showed theoretically and experimentally that such an identifier that is constructed using a series of shifted Barker sequences and is superposed it on the 802.11b preamble can be reliably detected at the receiver without considerably affecting typical preamble functions. The simulation and experimental results also verify that modulation unification and encryption are successful in hiding the true packet size, modulation scheme, and frame content without degrading the BER performance.

Fourth, we considered the STF of the preamble in the PHY layer of the IEEE OFDM systems to carry information bits in order to improve the throughput, identify the transmitter in the PHY-layer and facilitate PHY header encryption, or mitigate the FO attack by varying the preamble. To do so, we constructed a set of new preamble waveforms that comply with the requirements of the preamble of the OFDM systems in IEEE WLAN systems. These new waveforms are obtained using two techniques: shift in the time domain and phase rotation in the frequency domain. Our analysis indicates that in the systems that operate on 20 MHz bandwidth, the transmitter can embed up to 8 bits, and facilitate several security measures (e.g., full-frame encryption). If the a higher bandwidth is used, higher speed also can be achieved. Our simulation and USRP experiment results showed that the proposed scheme can maintain the performance of BPSK modulation scheme and also is robust to channel variations and frame detection errors.

7.2 Future Research Directions

We envision several future directions based on the schemes developed and the achieved results in this dissertation.

First, in this dissertation we assumed a single Tx-Rx-pair and presents a jamming attack against FO estimation in OFDM systems. In the case of multiple Tx-Rx pairs, Eve can construct a database of the FOs between different Tx-Rx pairs. Benefiting from CSMA/CA channel access mechanism, Eve can consider one transmission at a time and then leverage protocol semantics (e.g., data-ACK exchanges) to guess the Tx and Rx of an upcoming transmission. Extension of this attack to multi-link scenarios and further investigation of this issue can be considered for future work.

Second, this dissertation only sketches four countermeasures against the proposed FO attack. Detailed analysis and evaluation of these countermeasures are the next steps for shielding OFDM systems against FO attacks.

Third, in this dissertation, the friendly jamming devices are assumed to be single-antenna and dedicated to friendly jamming. To extend this work, one can assume these devices as both relay nodes and friendly jamming devices, and optimize their placements to minimize the jamming power while achieving certain quality of service. Using MIMO devices can be a choice when combining the role of relaying with the role of jamming. Furthermore, distributed MIMO devices can be exploited to increase the channel diversity and reduce the vulnerability zone around the receiver due to the strong line-of-sight.

Fourth, FCJ in this dissertation is designed for single-antenna devices. Extension of FCJ for MIMO devices and multi-user MIMO scenarios is a timely direction for future.

Finally, the scenarios considered in this dissertation mainly focus on WLAN systems. However, by the emerge of 5G and upcoming standards such as 802.11ax, various other applications will become pervasive. Vehicle-to-vehicle Vehicle-to-pedestrian communications are among these applications. Considering privacy in these applications and their specific features (e.g., fast changing the base station), we will need customized privacy-preserving schemes in the future.

REFERENCES

[1] "IEEE Std 802.11a-1999," *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*, 1999.

[2] "Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments to Grow 4.2 Percent in 2014," 2014. [Online]. Available: http://www.gartner.com/newsroom/id/2791017

[3] G. Wilkinson, "Black Hat Asia 2014 - the machines that betrayed their masters," 2014. [Online]. Available: https://www.youtube.com/watch?v=NiiI_oZ7y64

[4] R. Savage, "Snooping garbage bins in city of London ordered to be disabled," Aug. 2013. [Online]. Available: http://goo.gl/omktFa

[5] "IEEE 802 EC privacy recommendation study group," Jul. 2014. [Online]. Available: http://www.ieee802.org/PrivRecsg/

[6] F. Armknecht, J. Girao, A. Matos, and R. L. Aguiar, "Who said that? privacy at link layer," in *IEEE INFOCOM'07 conf.*, Anchorage, Alaska, USA, May 2007, pp. 2521–2525.

[7] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall, "Improving wireless privacy with an identifier-free link layer protocol," in *Proc. 6th Int. Conf. Mobile Syst., Appl., and Services*, Breckenridge, CO, USA, 2008, pp. 40–53.

[8] Y. Fan, B. Lin, Y. Jiang, and X. Shen, "An efficient privacy-preserving scheme for wireless link layer security," in *Proc. IEEE GLOBECOM conf.*, New Orleans, LO, USA, Nov. 2008, pp. 1–5.

[9] M. Alleven, "IEEE study group recommends improvements in Wi-Fi security," Jul. 2015. [Online]. Available: http://goo.gl/BPXMyg

[10] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, “Who do you sync you are?: Smartphone fingerprinting via application behaviour,” in *Proc. Sixth ACM Conf. Security and Privacy in Wireless and Mobile Networks (WiSec’13)*, Budapest, Hungary, 2013, pp. 7–12.

[11] C. Neumann, O. Heen, and S. Onno, “An empirical study of passive 802.11 device fingerprinting,” in *32nd IEEE Int. Conf. Distributed Computing Syst. Workshops (ICDCSW’12)*, Jun. 2012, pp. 593–602.

[12] Q. Wang, A. Yahyavi, B. Kemme, and W. He, “I know what you did on your smartphone: Inferring app usage over encrypted data traffic,” in *Proc. IEEE Conf. Commun. and Network Security CNS’15*, Sept 2015, pp. 433–441.

[13] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in *Proc. 14th ACM Int. Conf. Mobile Computing and Networking (MobiCom’08)*, San Francisco, California, USA, 2008, pp. 116–127.

[14] B. Miller, L. Huang, A. Joseph, and J. Tygar, “I know why you went to the clinic: Risks and realization of HTTPS traffic analysis,” *Privacy Enhancing Technologies*, vol. 8555, pp. 143–163, 2014.

[15] K. Dyer, S. Coull, T. Ristenpart, and T. Shrimpton, “Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail,” in *Proc. IEEE Symp. Security and Privacy (SP’12)*, May 2012, pp. 332–346.

[16] F. Zhang, W. He, X. Liu, and P. G. Bridges, “Inferring users’ online activities through traffic analysis,” in *Proc. 4th ACM Conf. Wireless Network Security (WiSec’11)*, Hamburg, Germany, 2011, pp. 59–70.

[17] S. Dai, A. Tongaonkar, X. Wang, A. Nucci, and D. Song, “Networkprofiler: Towards automatic fingerprinting of android apps,” in *Proc. IEEE INFOCOM’13*, Apr. 2013, pp. 809–817.

[18] S. Chen, R. Wang, X. Wang, and K. Zhang, “Side-channel leaks in web applications: A reality today, a challenge tomorrow,” in *Proc. IEEE Symp. Security and Privacy (SP’10)*, May 2010, pp. 191–206.

[19] G. Noubir, R. Rajaraman, B. Sheng, and B. Thapa, “On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming,” in *Proc. 4th ACM Conf. Wireless Network Security (WiSec’11)*, Hamburg, Germany, Jun. 2011, pp. 97–108.

[20] M. J. LaPan, T. C. Clancy, and R. W. McGwier, "Physical layer orthogonal frequency-division multiplexing acquisition and timing synchronization security," *Wireless Commun. and Mobile Computing*, vol. 16, no. 2, pp. 177–191, Feb. 2016.

[21] ——, "Phase warping and differential scrambling attacks against OFDM frequency synchronization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'13)*, Vancouver, BC, Canada, May 2013, pp. 2886–2890.

[22] C. Shahriar, S. Sodagari, R. McGwier, and T. Clancy, "Performance impact of asynchronous off-tone jamming attacks against OFDM," in *Proc. IEEE Int. Conf. Commun. (ICC'13)*, Budapest, Hungary, Jun. 2013, pp. 2177–2182.

[23] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Trans. on Wireless Commun.*, vol. 7, no. 6, pp. 2180–2189, Jun. 2008.

[24] "IEEE Std 802.11n-2009," *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 Amendment 5: Enhancements for Higher Throughput*, 2009.

[25] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

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

[27] L. Weng, R. Murch, and V. Lau, "SISO-OFDM channel estimation in the presence of carrier frequency offset," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC'06)*, vol. 3, Las Vegas, NV, USA, Apr. 2006, pp. 1444–1449.

[28] Y. Zhang, "Method, apparatus and system for carrier frequency offset estimation," Oct. 2013, US Patent App. 13/597,204.

[29] G. Lin and G. Noubir, "On link layer denial of service in data wireless LANs," *Wireless Communications and Mobile Computing*, vol. 5, no. 3, pp. 273–284, 2005.

[30] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, “Denial of service attacks in wireless networks: The case of jammers,” *Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 245–257, 2011.

[31] M. Strasser, B. Danev, and S. Čapkun, “Detection of reactive jamming in sensor networks,” *ACM Trans. Sensor Networks*, vol. 7, no. 2, pp. 16:1–16:29, Aug. 2010.

[32] T. C. Clancy, “Efficient OFDM denial: Pilot jamming and pilot nulling,” in *Proc. IEEE Int. Conf. Commun. (ICC’11)*, Kyoto, Japan, Jun. 2011.

[33] 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.*, Kyoto, Japan, 2007, pp. 385–396.

[34] C. Mueller-Smith and W. Trappe, “Efficient OFDM denial in the absence of channel information,” in *Proc. Military Commun. Conf. (MILCOM’13)*, San Diego, CA, USA, Nov. 2013, pp. 89–94.

[35] L. Sanguinetti, M. Morelli, and H. Poor, “Frame detection and timing acquisition for OFDM transmissions with unknown interference,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp. 1226–1236, Mar. 2010.

[36] H. Rahbari, M. Krunz, and L. Lazos, “Security vulnerability and countermeasures of frequency offset correction in 802.11a systems,” in *Proc. IEEE INFOCOM’14*, Toronto, ON, Canada, Apr. 2014, pp. 1015–1023.

[37] T. Pollet, M. Van Bladel, and M. Moeneclaey, “BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise,” *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 191–193, 1995.

[38] H. Rahul, H. Hassanieh, and D. Katabi, “SourceSync: a distributed wireless architecture for exploiting sender diversity,” in *Proc. ACM SIGCOMM*, New Delhi, India, Sep. 2010, pp. 171–182.

[39] H. Rahbari, M. Krunz, and L. Lazos, “Jamming attack on frequency offset estimation in OFDM systems,” University of Arizona, Department of ECE, Tech. Rep. TR-UA-ECE-2015-1, Jun. 2015. [Online]. Available: http://goo.gl/pAxxaR

[40] D. Nguyen, C. Sahin, B. Shishkin, N. Kandasamy, and K. R. Dandekar, “A real-time and protocol-aware reactive jamming framework built on software-defined radios,” in *Proc. 2014 ACM Workshop Software Radio Implementation Forum*, Chicago, Illinois, USA, 2014, pp. 15–22.

[41] N. Sufyan, N. Saqib, and M. Zia, “Detection of jamming attacks in 802.11b wireless networks,” *EURASIP J. Wireless Commun. and Networking*, vol. 2013, 2013.

[42] J. Luo, J. Andrian, and C. Zhou, “Bit error rate analysis of jamming for OFDM systems,” in *Wireless Telecommun. Symp. (WTS’07)*, Apr. 2007, pp. 1–8.

[43] C. Shahriar *et al.*, “PHY-Layer resiliency in OFDM communications: A tutorial,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 292–314, 2015.

[44] A. Wyner, “The wire-tap channel,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.

[45] I. Csiszar and J. Korner, “Broadcast channels with confidential messages,” *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.

[46] J. Zhang, L. Fu, and X. Wang, “Asymptotic analysis on secrecy capacity in large-scale wireless networks,” *IEEE/ACM Trans. Networking*, vol. 22, no. 1, pp. 66–79, Feb. 2014.

[47] O. Koyluoglu, C. Koksal, and H. Gamal, “On secrecy capacity scaling in wireless networks,” *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3000–3015, May 2012.

[48] A. Sheikholeslami, D. Goeckel, H. Pishro-Nik, and D. Towsley, “Physical layer security from inter-session interference in large wireless networks,” in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1179–1187.

[49] M. Ghaderi, D. Goeckel, A. Orda, and M. Dehghan, “Efficient wireless security through jamming, coding and routing,” in *Proc. 10th IEEE Int. Conf. Sensing, Commun., Netw. (SECON)*, New Orleans, USA, Jun. 2013, pp. 505–513.

[50] ——, “Minimum energy routing and jamming to thwart wireless network eavesdroppers,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1433–1448, Jul. 2015.

[51] D. Goeckel, S. Vasudevan, D. Towsley, S. Adams, Z. Ding, and K. Leung, “Artificial noise generation from cooperative relays for everlasting secrecy in two-hop wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 10, pp. 2067–2076, Dec. 2011.

[52] M. Chiang, C. W. Tan, D. Palomar, D. O’Neill, and D. Julian, “Power control by geometric programming,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, Jul. 2007.

[53] T. Shu and M. Krunz, “Coverage-time optimization for clustered wireless sensor networks: A power-balancing approach,” *IEEE/ACM Trans. on Netw.*, vol. 18, no. 1, pp. 202–215, Feb. 2010.

[54] C. Beightler and D. Phillips, *Applied Geometric Programming.* Wiley, 1976.

[55] C. Cardoso, A. Castro, and A. Klautau, “An efficient FPGA IP core for automatic modulation classification,” *IEEE Embedded Syst. Lett.*, vol. 5, no. 3, pp. 42–45, Sep. 2013.

[56] J. Freudiger, “How talkative is your mobile device?: an experimental study of Wi-Fi probe requests,” in *Proc. 8th ACM Conf. Security and Privacy in Wireless & Mobile Networks (WiSec’15)*, New York City, USA, Jun. 2015.

[57] S. U. Rehman, K. W. Sowerby, and C. Coghill, “Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers,” *J. Comput. and Syst. Sci.*, vol. 80, no. 3, pp. 591–601, 2014, special Issue on Wireless Network Intrusion.

[58] T. D. Vo-Huu and G. Noubir, “Mitigating rate attacks through crypto-coded modulation,” in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc’15)*, Hangzhou, China, 2015, pp. 237–246.

[59] H. Rahbari and M. Krunz, “Friendly CryptoJam: A mechanism for securing physical-layer attributes,” in *Proc. 7th ACM Conf. Security and Privacy in Wireless & Mobile Networks (WiSec’14)*, Oxford, United Kingdom, Jul. 2014, pp. 129–140.

[60] F. Zhang, W. He, Y. Chen, Z. Li, X. Wang, S. Chen, and X. Liu, “Thwarting Wi-Fi side-channel analysis through traffic demultiplexing,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 86–98, Jan. 2014.

[61] N. Kolokotronis, A. Katsiotis, and N. Kalouptsidis, “Short paper: attacking and defending lightweight PHY security schemes for wireless communications,” in *Proc. 7th ACM Conf. Security and Privacy in Wireless & Mobile Networks (WiSec’14)*, Oxford, United Kingdom, Jul. 2014, pp. 177–182.

[62] A. Katsiotis, N. Kolokotronis, and N. Kalouptsidis, “Physical layer security via secret trellis pruning,” in *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Commun. (PIMRC’13)*, Sep. 2013, pp. 507–512.

[63] ——, “Secure encoder designs based on turbo codes,” in *Proc. IEEE ICC’15*, London, UK, Jun. 2015, pp. 4315–4320.

[64] N. Anand, S.-J. Lee, and E. Knightly, “STROBE: Actively securing wireless communications using zero-forcing beamforming,” in *Proc. IEEE INFOCOM’12*, Mar. 2012, pp. 720–728.

[65] S. Goel and R. Negi, “Guaranteeing secrecy using artificial noise,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2180–2189, Jun. 2008.

[66] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, “They can hear your heartbeats: Non-invasive security for implantable medical devices,” in *Proc. ACM SIGCOMM’11*, Toronto, Ontario, Canada, Aug. 2011, pp. 2–13.

[67] N. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun, “On limitations of friendly jamming for confidentiality,” in *Proc. IEEE symp. Security and Privacy (SP ’13)*, May 2013, pp. 160–173.

[68] M. Schulz, A. Loch, and M. Hollick, “Practical known-plaintext attacks against physical layer security in wireless MIMO systems,” in *Proc. Network and Distributed Syst. Security Symp. (NDSS’14)*, San Diego, CA, USA, Feb. 2014.

[69] X. He, H. Dai, W. Shen, and P. Ning, “Is link signature dependable for wireless security?” in *Proc. IEEE INFOCOM’13*, Apr. 2013, pp. 200–204.

[70] S. Gollakota and D. Katabi, “ZigZag decoding: Combating hidden terminals in wireless networks,” in *Proc. ACM SIGCOMM’08*, Seattle, WA, USA, Oct. 2008, pp. 159–170.

[71] F. Hameed, O. Dobre, and D. Popescu, “On the likelihood-based approach to modulation classification,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 5884–5892, Dec. 2009.

[72] S. Ravanbakhsh, C. Srinivasa, B. Frey, and R. Greiner, “Min-max problems on factor graphs,” in *Proc. 31st Int. Conf. Machine Learning (ICML’14)*, Beijing, China, Jun. 2014, pp. 1035–1043.

[73] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Information Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[74] E. Biglieri, D. Divsalar, M. K. Simon, and P. J. McLane, *Introduction to Trellis-Coded Modulation with Applications*, 1st ed., J. Griffin, Ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.

[75] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, “Sponge-based pseudo-random number generators,” *Cryptographic Hardware and Embedded Systems (CHES)*, vol. 6225, pp. 33–47, 2010.

[76] I. Dinur, P. Morawiecki, J. Pieprzyk, M. Srebrny, and M. Straus, "Cube attacks and cube-attack-like cryptanalysis on the round-reduced Keccak sponge function," *Advances in Cryptology–EUROCRYPT*, pp. 733–761, 2015.

[77] Y. Fan, B. Lin, Y. Jiang, and X. Shen, "An efficient privacy-preserving scheme for wireless link layer security," in *Proc. IEEE Int. Global Telecommun. Conf. (GLOBECOM'08)*, New Orleans, LA, USA, Nov. 2008.

[78] O. Bejarano, S. Quadri, O. Gurewitz, and E. Knightly, "Scaling multi-user MIMO WLAN: the case of concurrent uplink control messages," in *Proc. IEEE SECON'15*, Seattle, WA, USA, Jun. 2015.

[79] "IEEE Std 802.11ac-2013," *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–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, 2013.

[80] N. Anand, S.-J. Lee, and E. Knightly, "STROBE: Actively securing wireless communications using Zero-Forcing Beamforming," in *Proc. IEEE INFOCOM'12*, Orlando, FL, USA, Mar. 2012, pp. 720–728.

[81] Z. Feng, J. Ning, I. Broustis, K. Pelechrinis, S. Krishnamurthy, and M. Faloutsos, "Coping with packet replay attacks in wireless networks," in *Proc. IEEE SECON'11 conf.*, Jun. 2011, pp. 368–376.

[82] T. C. Clancy, "Efficient OFDM denial: Pilot jamming and pilot nulling," in *Proc. IEEE Int. Conf. Commun. (ICC'11)*, 2011, pp. 1–5.

[83] C. Shahriar, R. McGwier, and T. Clancy, "Performance impact of pilot tone randomization to mitigate OFDM jamming attacks," in *IEEE Consumer Commun. Networking Conf. (CCNC'13)*, Jan. 2013, pp. 813–816.

[84] Q. Wang, A. Yahyavi, B. Kemme, and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," in *Proc. IEEE Conf. Commun. Network Security (CNS'15)*, Sep. 2015, pp. 433–441.

[85] C. J. Bernardos, J. C. Zúńiga, and P. O'Hanlon, "Wi-Fi Internet connectivity and privacy: hiding your tracks on the wireless Internet," in *Proc. IEEE Conf. Standards for Commun. Netw. (CSCB'15)*, Tokyo, Japan, Oct. 2015.

[86] M. Gruteser and D. Grunwald, “Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis,” *Mobile Networks and Applications*, vol. 10, no. 3, pp. 315–325, 2005.

[87] “Specification framework for TGah,” May 2013, doc.: IEEE P802.11-11/1137r15.

[88] R. Boehnke and T. Doelle, “Alternative proposal for BRAN SYNCH preamble,” Mar. 1999, doc.: IEEE 802.11-99/048. [Online]. Available: http://goo.gl/NIy6BM

[89] “P802.11a draft D5.5 comments,” Jun. 1999, doc.: IEEE P802.11-99/1460. [Online]. Available: http://goo.gl/Rw3Zxi

[90] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. Prentice-Hall, 1996.

[91] K. Zhong, T. T. Tjhung, and F. Adachi, “A general SER formula for an OFDM system with MDPSK in frequency domain over Rayleigh fading channels,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 584–594, 2004.

[92] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.

[93] K. Sathananthan and C. Tellambura, “Probability of error calculation of OFDM systems with frequency offset,” *EEE Trans. Commun.*, vol. 49, no. 11, pp. 1884–1888, Nov. 2001.