Full Frame Encryption and Modulation Obfuscation Using Channel-independent Preamble Identifier

Hanif Rahbari and Marwan Krunz, *Fellow, IEEE*

***Abstract*—The broadcast nature of wireless communications exposes various "transmission attributes," such as the packet size, inter-packet times, and the modulation scheme. These attributes can be exploited by an adversary to launch passive or active attacks. A passive attacker threatens user privacy by performing traffic analysis and classification, whereas an active attacker exploits captured attributes to launch selective packet jamming/dropping. This security problem is present even when upper-layer payloads are encrypted. For example, by determining the modulation scheme, the attacker can estimate the data rate, and hence the payload size. This information can later be used for traffic classification or to launch selective rate-adaptation attacks.**

In this paper, we propose *Friendly CryptoJam* (FCJ). FCJ decorrelates the payload's modulation scheme from other transmission attributes by embedding information symbols into the constellation map of the highest-order modulation scheme supported by the system (a concept we refer to as *indistinguishable modulation unification*). Such unification is done using minimum-complexity trellis-coded modulation that is combined with a secret pseudo-random sequence to conceal the structure imposed by the code. It preserves the BER performance of the original modulation scheme (before unification). At the same time, modulated symbols are encrypted to hide PHY-/MAC layer fields. To generate and sync the secret sequence at the Tx/Rx, an efficient identifier embedding technique based on Barker sequences is proposed, which exploits the structure of the preamble and overlays a frame-specific identifier on it. We study the implications of the scheme on PHY-layer functions through simulations and USRP-based experiments. Our results confirm the efficiency of FCJ in hiding the targeted attributes.

***Index Terms*—PHY-layer security, side-channel information, modulation unification, preamble, untraceable TCM, USRP.**

I. Introduction

USING commodity radio, unauthorized parties can easily eavesdrop on wireless transmissions. Although advanced encryption algorithms like AES can be applied to ensure data confidentiality, parts of the frame (e.g., PHY/MAC headers) must be transmitted in the clear for correct protocol operation and device identification. For example, 802.11i, the primary security amendment of 802.11, provides confidentiality only for the MAC-layer payload. Even if we hypothetically encrypt the entire PHY frame, the transmission is not completely immune to eavesdropping. An adversary can still fingerprint encrypted traffic through analyzing its *side-channel information* (SCI). It refers to statistical traffic features, such as packet size distribution, traffic volume, and inter-packet time sequence. These statistical features can be obtained by estimating and correlating leaked *transmission attributes*, including frame duration, the modulation scheme, traffic directionality (uplink/downlink), and inter-packet times. Traffic fingerprints can be used to breach user privacy by tracking her or discerning her identity, activity, and interests. For example, by eavesdropping on 802.11 WLAN traffic for only 5 seconds, an adversary (Eve) can determine the type of user activities with 80% accuracy [1]. The sizes (in bytes) and direction of packets exchanged between a mobile user and an access point may reveal what phrase the user is searching for in a search engine [2], and identify the browsed page [3] or the language used in an encrypted instant messaging application. SCI can also facilitate user tracking by identifying her particular smartphone among many possible devices [4].

By analyzing transmission attributes, Eve can further learn the type or stage of a communication, and launch selective jamming attacks. For example, Noubir *et al.* [5] demonstrated a reactive jammer that can significantly hammer the network throughput by intercepting the rate field in the header and accordingly decide whether to jam the rest of the frame. If a packet is not correctly decoded as a result of jamming, the transmitter (Alice) mistakenly assumes a poor channel and lowers the rate when retransmitting the same packet, wasting network resources.

To obtain transmission attributes, Eve can intercept unencrypted fields in the PHY and MAC headers [1], [2], [5], [6]. These fields include the source/destination MAC addresses, payload transmission rate and modulation scheme, frame length/duration, traffic directionality, number of MIMO streams, and others. Eve can also perform low-level RF analysis to obtain SCI even when PHY/MAC headers are encrypted, 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 [7], 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.

A. Existing Countermeasures and Their Limitations

Before describing the various techniques that have been proposed to prevent Eve from obtaining SCI or intercepting PHY and MAC headers, we first explain why a naive approach based on encrypting these headers is not practical. To decrypt them, 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

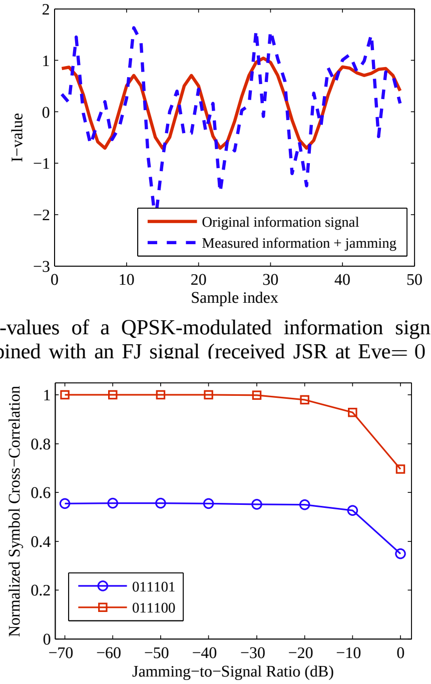
H. Rahbari and M. Krunz are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721.
E-mail: {rahbari,krunz}@email.arizona.edu

recently been employed for hiding the true address (e.g., in the probe requests in Apple iOS 8.1.3) is also not sufficiently helpful. Besides its other vulnerabilities [8], 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 [9].

Techniques to prevent SCI leakage can be divided into three categories: SCI obfuscation at upper layers [10]–[14], rate hiding in our initial work [15] and in more recent scheme [16], and eavesdropper deafening at the PHY layer. Upper-layer SCI obfuscation techniques aim at invalidating SCI, usually at the cost of traffic overhead. For example, packet padding can be used to alter the traffic statistics. However, the overhead can be as high as 400% [12]. *Traffic reshaping* [10] is a MAC layer technique that involves configuring several virtual interfaces with different MAC addresses for the same device so as to create different traffic patterns on each 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 [14]. However, these identifier concealment techniques cannot hide certain attributes, including the modulation scheme.

To hide the payload's modulation scheme, Conceal and Boost Modulation (CBM) was proposed in [16], whereby convolutional codes based on a Generalization of Trellis Coded Modulation (GTCM) are used, combined with a cryptographic interleaving mechanism. 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. Due to acute susceptibility of higher-order modulation schemes to phase offset, GTCM codes also suffers significantly from inaccurate FO estimation, reducing its coding gain.

PHY-layer eavesdropper deafening techniques include friendly jamming (FJ), e.g., [17]–[19]. 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 (friendly) to Bob. However, four 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 [20], [21]. For example, Schulz *et al.* [21] 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 deafening scheme that relies on signal prefiltering (precoding) at Alice. Furthermore, the uniqueness of the Alice-Bob CSI has been shown not to be true in the presence of strong LOS component [22]. Specifically, a few adversaries located several (~ 18) wavelengths away from Bob can cooperatively reconstruct Alice-Bob channel's signature.



(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).

Fig. 1. Cross-correlation attack on a semi-static QPSK-modulated signal that is superposed on an FJ signal.

Second, FJ requires additional transmission power and antenna(s) or relays, 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 nonzero secrecy capacity [18]. Moreover, Alice may not have sufficient number of antennas (degrees of freedom) to apply FJ.

Third, transmitter- and receiver-based FJ (e.g., [19]) are still vulnerable to *cross-correlation attacks* on (unencrypted) semi-static header fields, the fields that can take one of a few valid values. Eve can detect the start of a frame, even if it is combined with a jamming signal [23]. By knowing the underlying header format (i.e., where each field is supposed to start), Eve can locate the start time of a targeted field in the header. Fig. 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 Fig. 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 power ratio (JSR). Each point is the average of 100 simulation trials. The plot shows that Eve can successfully determine the true value even when JSR$=0\,$dB.

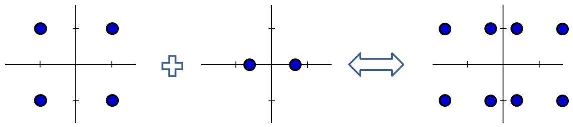


Fig. 2. Combining QPSK-modulated and BPSK-modulated signals with different powers results in a 8-symbol constellation map.

Last but not least, 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., [7], [24]) 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 Fig. 2).

B. 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. Compared to our initial proposal of FCJ [15], the modulation encryption in this paper 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 simultaneously coded and mapped (upgraded) to the constellation map of the highest-order (target) modulation scheme supported by the system. We develop a modulation coding that prevents the disclosure of the payload’s original modulation scheme, i.e., it provides *indistinguishable modulation unification*. In contrast to the uncoded modulation unification in the initial design [15] and variable-rate coding for upgrading different modulation schemes to same target modulation scheme in CBM [16], the novel mapping proposed in this paper employs only two minimal trellis-coded modulation (TCM) codes with constraint length ≤ 2 (and constant rate irrespective of the target modulation scheme). These codes are inseparably combined with the FJ traffic so as to continuously move the coded symbols on the target constellation map while maintaining BER. This way, we hide both the true modulation scheme and the structure imposed by the underlying TCM code without symbol interleaving. Compare to [16], FCJ also enjoys lower complexity, decoding delay, and susceptibility to FO, but at the expense of lower coding gain. We further provide an analytical study of the impact of uncompensated FO. In contrast to classic FJ techniques, a single antenna is sufficient to transmit both the information and FJ signals.

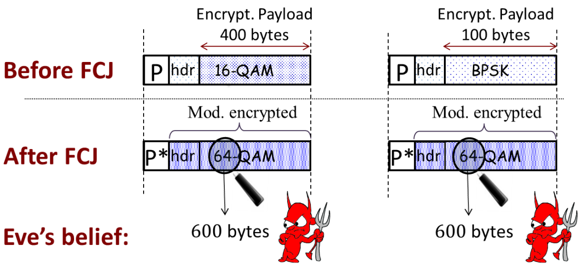


Fig. 3. Example of using *Friendly CryptoJam* to hide the header fields and the modulation scheme (and payload size) of two frames. Headers and payload are modulated-encrypted and modulation-unified without changing the information rate and frame duration. Under FCJ, an identifier ($\mathcal{ID}$) is overlaid on the original frame preamble (P), leading to a new preamble (P^*).

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 preshared 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 mitigate 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 Fig. 3.

II. Background – PHY-layer Attributes and Preamble Structure

(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 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) Frame detection and FO estimation. Each PHY header is preceded by a preamble, which is used for frame detection, FO and CSI estimation. 802.11b systems exploit a scrambled version of a 128-bit all-one preamble that is spread using an 11-chip Barker sequence (see Table I). For a Barker sequence of length N, its autocorrelation function at lag k, denoted by $\mathcal{A}(k)$, is very low at non-zero lags (orthogonality property). This can be exploited for frame detection and timing. Formally,

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

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

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

$\mathcal{R}(\mathbf{b}, n)$ 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 I
DSSS SIGNAL SPREADING BASED ON AN 11-CHIP BARKER SEQUENCE FOR DBPSK MODULATION (802.11B).

The preamble consists of several repetitions of a publicly known pattern. FO estimation involves detecting the arrival of at least two identical portions of the preamble.[1] An FO in the amount of δ_f Hz creates a time-varying phase displacement $\varphi(t) = 2\pi\delta_f\, t$. To decode a frame, Bob estimates δ_f by taking one of the repetitions in the received signal as a reference and comparing it with another repetition that is T seconds away. Specifically, Bob subtracts the phases of any pair of identical samples to find $\varphi(T)$. Because of noise, usually there will be a residual FO estimation error even after averaging over several of such identical pairs. Depending on the frame duration, the residual FO may move a received symbol to a wrong region on the constellation map, causing a demodulation error. After compensating for FO, Bob compares the known pattern in the preamble with its received value to estimate the CSI.

(3) Detection of lower-layer fields. Typically, the preamble and the PHY header are transmitted at the lowest supported rate.[2] 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, 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 correctly guess the data rate.

III. 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[3]. 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., Fig.1(b)), rate-adaptation attack [5], device-based user-tracking attacks, dictionary attacks, known-plaintext attack [21], 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 detection. 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.

Fig. 4 shows a schematic view of Alice's transmit chain and the insertion points of FCJ's components, which include modulation-aware encryption (point 1), modulation unification (point 2) and frame ID embedding (point 3). Once the frame payload, which starts with the MAC header, arrives at Alice's PHY layer, Alice computes the PHY-header fields, including the modulation scheme for the payload that is calculated based on CSI. The PHY header and the payload are then modulated and together with the spread preamble are passed through the

[1] Scrambling transforms an all-one preamble bit sequence into a sequence of zero's and one's. Methods like [25] are used to detect the zero's and change them to one's.

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

[3] 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.

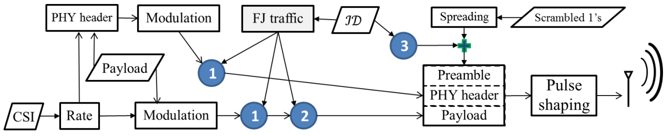


Fig. 4. Transmission chain at Alice under FCJ. Insertion points (1), (2), and (3) refer to modulation encryption, TCM-based modulation unification, and message embedding within the preamble.

FCJ components, before they are concatenated 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.

Table II provides a list of key notations used in this paper.

Variable	Definition
$\mathcal{M}_i$	Modulation scheme $i, i = 1, \ldots, M$
δ_f	The amount of frequency offset
$\varphi(t)$	Phase offset at time t elapsed from the beginning of the frame
$\gamma_i^{(q)}(M)$	Asymptotic coding gain when embedding $\mathcal{M}_i$ in $\mathcal{M}_M$ using an q-state TCM
$\mathbf{j}$	FJ traffic sequence
$\mathcal{U}_j$	Set of $\lvert\mathcal{M}_i\rvert$ elements of $\mathcal{M}_M$ corresponding to FJ bits j
$\mathcal{F}_{\mathbf{j}}$	Static mapping used for modulation unification based on $\mathbf{j}$
$\mathcal{E}_{\mathbf{j}}$	Modulation encryption function based on $\mathbf{j}$

TABLE II
MAIN NOTATIONS

IV. 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 and the header's decryption key will be explained in Section V along with the modulation encryption scheme.

A. 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 $\mathcal{M}_M$). At the same time, we need to preserve the original 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 constellation map of $\mathcal{M}_M$. As long as the distribution of these points in the target constellation map is uniform, similar to the distribution of the points of a random $\mathcal{M}_M$-modulated information signal, and a given symbol is independent of the previous and next symbols (from Eve's perspective), Eve cannot determine if $\mathcal{M}_i \neq \mathcal{M}_M$.

In general, a higher-order modulation scheme is more susceptible to demodulation errors. 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 (hence, the BER) at a given SNR value. This $d_{min,i}$ generally decreases with i. Tables III and IV depict $d_{min,i}$ for the 802.11a system after taking into account a modulation-dependent normalization factor K_{MOD} [26], a coefficient that is multiplied by the (I,Q) values to achieve the same average symbol-power across different $\mathcal{M}_i$'s. Let the FJ traffic sequence be $\mathbf{j}$ and let $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$ be a static mapping, known to both Alice and Bob, that is used to embed the symbols of $\mathcal{M}_i$ (where $\mathcal{M}_1$ is the lowest-order modulation scheme) into the constellation map of $\mathcal{M}_M$. To maintain the same $d_{min,i}$ after upgrading $\mathcal{M}_i$ to $\mathcal{M}_M$, any two neighboring points in the constellation of $\mathcal{M}_i$ should ideally 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. Otherwise, Eve may discern $\mathcal{M}_i$ by performing statistical analysis. In the following, we explain an uncoded mapping $\mathcal{F}_{\mathbf{j}}$, first proposed in [15], 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 disjoint 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 in $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$ depends on $\mathbf{j}$ and hence can be different from one symbol to another. For any $\mathcal{M}_i$-modulated symbol s, Alice needs $(\log_2|\mathcal{M}_M| - \log_2|\mathcal{M}_i|)$ bits in $\mathbf{j}$ to select one of these sets. Let $j, j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$, be the decimal representation of those FJ bits, and let $\mathcal{U}_j = \{u_j^0, \ldots, u_j^{|\mathcal{M}_i|-1}\}$ be the corresponding set (subconstellation) on the constellation map of $\mathcal{M}_M$. The same j always points to the same set $\mathcal{U}_j$ and the value of s determines one of the points inside $\mathcal{U}_j$. This ensures that the transmitted symbols are equally probable, assuming that the bits in $\mathbf{j}$ are uniformly distributed (so as Alice's symbols). As explained in Section V, we rely on a cryptographic hash function like SHA-3 to generate $\mathbf{j}$.

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. Note that the number of FJ bits per symbol varies for different $\mathcal{M}_i$'s and $\mathcal{M}_M$'s. During the decoding process, Bob knows $\mathbf{j}$ and $\mathcal{M}_i$. To obtain the original symbol s, Bob considers only those constellation points in $\mathcal{M}_M$ that belong 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.

Next, we discuss a strategy for constructing the sets $\mathcal{U}_j$ by optimally partitioning the constellation of $\mathcal{M}_M$ into $|\mathcal{M}_M|/|\mathcal{M}_i|$ disjoint subconstellations. Let $d_{min}(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i))$ be the minimum distance between any two elements in $\mathcal{U}_j$ over all possible $j, j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$. In here, optimality of partitioning is taken w.r.t. maximizing $d_{min}(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i))$. For a modulation scheme $\mathcal{M}_M$ whose symbols are uniformly distributed in a square grid (e.g., 16-QAM) or over a circle, several solutions were obtained in [27]. We verify their optimality by solving the *circle packing* problem with $|\mathcal{M}_i| = |\mathcal{U}_j|$ identical circles in a square [28]. Ideally, every element of a set $\mathcal{U}_j$ in this case should be surrounded by as many elements of other sets as possible in order to maximize $d_{min}(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i))$.

i	$\mathcal{M}_i$	K_{MOD}[26]	$d_{min,i}$	$d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)$	$\gamma_i(3)$	$\gamma_i^{(2)}(3)$	$\gamma_i^{(4)}(3)$
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 III
PARAMETERS OF THE OPTIMAL MAPPING FROM BPSK AND QPSK TO 16-QAM.

i	$\mathcal{M}_i$	K_{MOD}[26]	$d_{min,i}$	$d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)$	$\gamma_i(4)$	$\gamma_i^{(2)}(4)$	$\gamma_i^{(4)}(4)$
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 IV
PARAMETERS OF THE OPTIMAL MAPPING FROM 802.11A MODULATION SCHEMES TO 64-QAM.

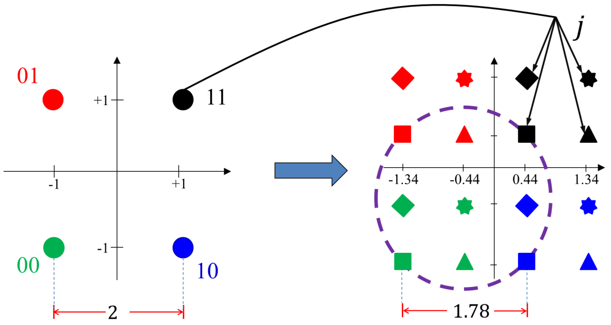


Fig. 5. Optimal mapping from QPSK to 16-QAM. The points that belong to the same set $\mathcal{U}_j, j = 0, \ldots, 3$, are shown using the same shape. For example, the squares on the dashed circle constitute $\mathcal{U}_0$.

This implies that the elements of every optimal set $\mathcal{U}_j$, $j = 0, \ldots, |\mathcal{M}_M|/|\mathcal{M}_i| - 1$, must be uniformly distributed across the grid points. As such, the maximum circle diameter in the corresponding circle packing problem upper-bounds $d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)$. We determine the size of the underlying square for each $\mathcal{M}_i$ in the circle-packing problem via aligning the centers of the quadrant of this square at the center of the corresponding quadrant in the constellation map of $\mathcal{M}_M$.

The known optimal circle packing solutions [28] confirm the optimality of the partitions in [27]. In particular, the optimal circles are automatically aligned as a grid when $|\mathcal{M}_i| = 4$, 16 or aligned on a diagonal of a grid when $|\mathcal{M}_i| = 2$; hence, $d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)$ archives its upper bound. All $\mathcal{U}_j$'s can be reconstructed via shifting and/or rotating the whole set of optimal circles and aligning them on various grid points in $\mathcal{M}_M$'s constellation. For other modulation orders $|\mathcal{M}_i|$ whose corresponding optimal circle packing solutions are not automatically a grid, the maximum circle diameter specifies how far two adjacent grid points in $\mathcal{U}_j$ can be; making it easy to obtain an optimal partitioning. After $\mathcal{U}_j$'s are constructed, the $|\mathcal{M}_i|$ bits are assigned to the symbols in each set $\mathcal{U}_j$ based on Gray coding. The correspondence between the symbols in $\mathcal{U}_j$ and $\mathcal{M}_i$ to which the same bits are assigned defines $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$.

It turns out that each subconstellation $\mathcal{U}_j$ for the constellation maps in 802.11 systems is a scaled down and shifted/rotated version of $\mathcal{M}_i$'s constellation. Hence, the mapping $\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)$ is readily available. In Fig. 5, we illustrate 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 a $\mathcal{M}_i$-modulated symbol in $\mathcal{M}_M$ constellation map is specified by the payload bits while j specifies its position within that quadrant. Next, in Fig. 6 we show a partitioning of 64-QAM constellation into four optimal sets to embed 16-QAM-modulated symbols.

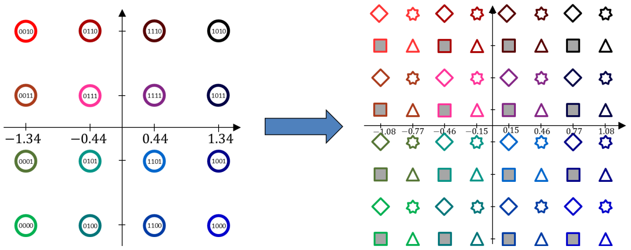


Fig. 6. Optimal mapping from 16-QAM to 64-QAM. The points that belong to the same set $\mathcal{U}_j, j = 0, \ldots, 3$, are shown using the same shape.

For other modulation schemes whose optimal partitions are not known, the problem of maximizing $d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)$ (a max-min problem) can be converted via changing the sign of the distance matrix to the *min-max clustering* problem. This problem can be solved in $\mathcal{O}(|\mathcal{M}_M|^2\,|\mathcal{M}_i|\,\log(|\mathcal{M}_M|))$ time to obtain a near-optimal solution [29].

The above mapping may not maintain $d_{min}\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right) \geq d_{min,i}$ for all $\mathcal{M}_i$'s. Let $\gamma_i(M)$ be the demodulation performance gain of mapping $\mathcal{M}_i$ into $\mathcal{M}_M$:

$$\gamma_i(M) = \frac{d_{min}^2\left(\mathcal{F}_{\mathbf{j}}(\mathcal{M}_i)\right)}{d_{min,i}^2}, i = 1, \ldots, M-1. \qquad (3)$$

For optimal mapping to 16-QAM and 64-QAM, BPSK and QPSK will have a gain of about -1.18 dB (1.18 dB loss) and -0.97 dB (0.97 dB loss), respectively, as shown in Tables III and IV. We compensate for this loss by applying a novel untraceable modulation coding technique.

B. Residual FO Estimation Error

Besides the SNR and $d_{min,i}$, the demodulation performance at Bob depends on how accurate he estimates δ_f. An error in estimating δ_f manifests itself as a phase offset that increases linearly with the symbol index and may eventually displace a received symbol out of its expected region in the constellation map (see Section II). Therefore, for the same residual FO and SNR level, longer frames experience more symbol/bit errors towards the end of the frame.

While applying Gray coding in denser constellations alleviates the consequences of a symbol error on BER, higher sym-

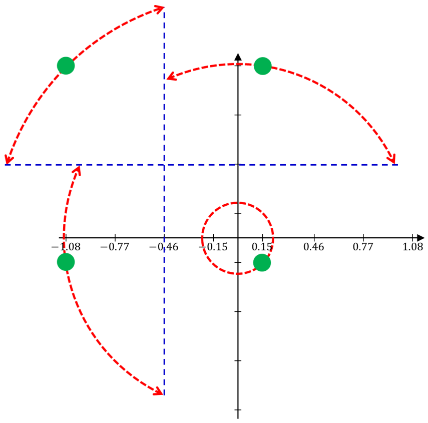


Fig. 7. Example of uneven impact of phase offset on symbols with different amplitudes when QPSK symbols are mapped to four 64-QAM symbols.

bol density in higher-order modulation schemes contributes to more symbol errors and higher susceptibility to FO estimation errors compare to sparse modulation schemes. 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 [16], [27]). One advantage of the uncoded modulation unification in FCJ over CBM [16] 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 of a subconstellation $\mathcal{U}_j$, which is used to embed the symbols of symmetric constellation map of $\mathcal{M}_i$, are asymmetrically distributed. Therefore, with the same phase offset φ, different symbols experience uneven displacements. That means some of the symbols are more robust to phase offset and other symbols are more vulnerable. We illustrate this in Fig. 7 with an example of one of the $\mathcal{U}_j$'s used to hide 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 minimum amount of displacement that can cause a symbol error. 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 Fig. 8, we numerically 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 (Fig. 8(a) and Fig. 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 shrinks 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 Fig. 8(c). In this figure we also plot the BER for 64-QAM. Comparing this plot to the ones of BPSK and QPSK, we observe high susceptibility of any unification scheme that directly maps BPSK and QPSK to 16-QAM and 64-QAM (e.g., CBM [16]).

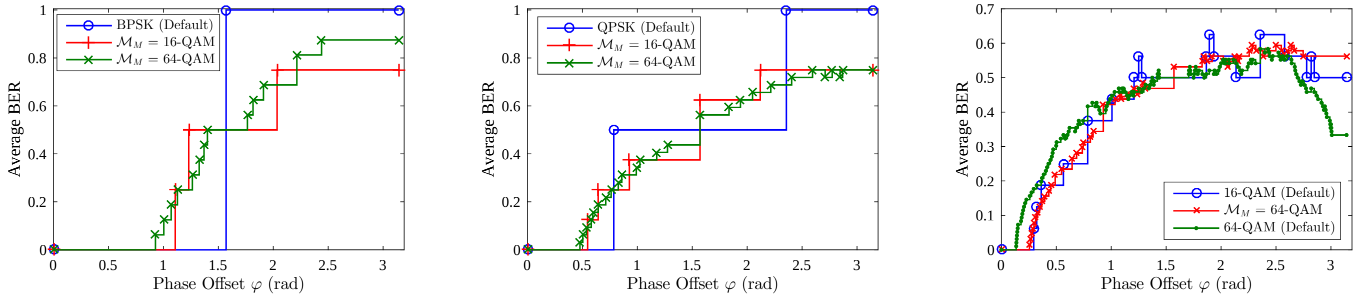
C. *Untraceable Trellis-Coded Modulation Unification*

Assuming $\delta_f = 0$, FCJ can maintain the same demodulation performance of $\mathcal{M}_i$ by coding the $\mathcal{M}_M$-modulated symbols. Coding creates dependency among successive symbols, which can be exploited at Bob to more accurately guess the symbol sequence. To identify and track the most probable paths (i.e., sequences) at Bob, a trellis diagram together with Viterbi algorithm are often employed. Trellis-coded modulation [27], [30] is a generic coding technique that instead of generating a sequence of correlated bits, directly generates a sequence of correlated symbols that belong to a higher-order modulation scheme to represent uncorrelated $\mathcal{M}_i$-modulated symbols and improve reliability. A set of "states" is defined as the encoder memory to impose the dependency. State transitions and the associated transmitted symbols are then controlled by information bits. Fig. 9 shows an example of a 2-state TCM encoder and its corresponding trellis diagram that encodes $\mathcal{M}_i$ =BPSK symbols using the symbols of a 4-symbol modulation scheme (e.g., QPSK). Introducing such a dependency in FCJ to encode a $\mathcal{M}_i$-modulated symbol s is possible because $|\mathcal{M}_M| > |\mathcal{M}_i|$. For the time being, however, let the higher-order modulation scheme consists of only $\mathcal{U}_0 + \mathcal{U}_1$, irrespective of the FJ bits *j*. The asymptotic coding gain of TCM is defined as

$$\gamma_i^{(q)}(M) = \frac{d^2_{free}}{d^2_{min,i}} \tag{4}$$

where d_{free} (free distance) is the minimum total Euclidean distance between the symbols along any two distinct paths in the trellis diagram and q is the number of states. To get benefit of TCM and prevent a performance loss due to modulation unification, we need to satisfy $d_{free} \geq d_{min,i}$. 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 [16], 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. They embed $|\mathcal{M}_i|$ symbols into $2|\mathcal{M}_i|$ symbols of $\mathcal{M}_M$. These codes are based on either two-state (Fig. 9) or four-state (Fig. 10) TCMs presented in [27] with constraint lengths of 1 and 2, respectively. One advantage of having a low constraint length is that when Bob employs Viterbi algorithm 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. (When $\mathcal{M}_i \neq$ BPSK, the same structures are used but multiple parallel edges need to be defined for each state transition.)

Using Ungerboeck's assignment rules to assign the symbols to the edges/transitions (as shown in Fig. 9 and 10), we maximize d_{free} and hence the gain for each TCM scheme, without incurring significant decoding complexity. (Note that, for example, the least-complex code in [16] for mapping from

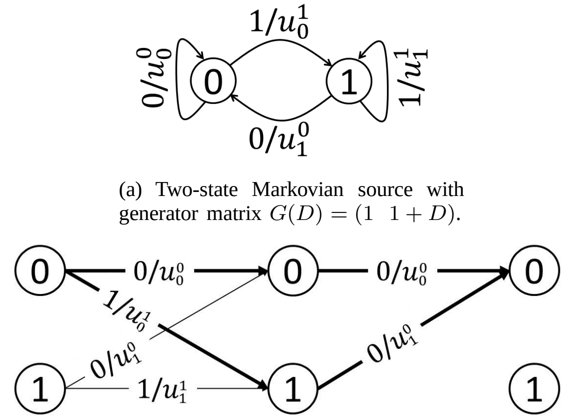


(a) $\mathcal{M}_i$ = BPSK.

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

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

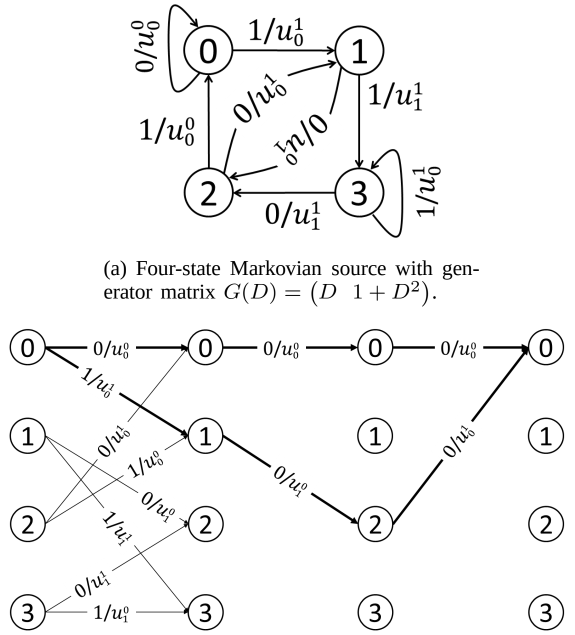
Fig. 8. Impact of phase offset on different $\mathcal{M}_i$ and $\mathcal{M}_M$ combinations in modulation unification.



(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.

Fig. 9. Minimal two-state TCM scheme. The edge label I/O denotes the transmission of symbol O if the input is I.



(a) Four-state Markovian source with generator matrix $G(D) = \left(D \;\; 1+D^2\right)$.

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

Fig. 10. Minimal four-state TCM scheme. The edge label I/O denotes the transmission of symbol O if the input is I.

BPSK to 64-QAM has the constraint length of 5.) The $2|\mathcal{M}_i|$ symbols consist of the symbols of any 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 Fig. 9 and 10, $\mathcal{U}_0$ and $\mathcal{U}_1$ are used.) The coding gains of the proposed TCM schemes and sets $\mathcal{U}_j$ are shown in Tables III and IV. The two-state TCM maintains 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.

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 [16], which use all $|\mathcal{M}_M|$ symbols to achieve higher gains. Such selection exhibits lower constellation density than $\mathcal{M}_M$, and so is less susceptible to inaccurate FO estimation than CBM [16]. In addition, 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, a feature that is exploited in FCJ to facilitate indistinguishable modulation unification.

The known code rate and/or the dependency among coded symbols in these TCM codes may leak $\mathcal{M}_i$. 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, static assignment of symbols to the edges and so the dependency among the pairs of symbols along the trellis diagram also can reveal $|\mathcal{M}_i|$. 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, and then, project the observed symbols on the trellis structure to find out $\mathcal{M}_i$.

To prevent the leakage of $\mathcal{M}_i$, we propose exploiting the FJ bits j to dynamically change the sets $\mathcal{U}_{j_1}$ and $\mathcal{U}_{j_2}$ to $\mathcal{U}_j$ and $\mathcal{U}_{(j+1) \bmod |\mathcal{M}_M|/|\mathcal{M}_i|}$, respectively, at each state transition. From a security perspective, randomly replacing $\mathcal{U}_j$'s allows us to generate all $|\mathcal{M}_M|$ symbols with equal probability. Because Bob knows j, he can limit the set of possible symbols to the ones mandated by the FJ bits, and keep track of transitions and compute the distances/errors bas before. So since the sets $\mathcal{U}_j$ are identical (they are shifted versions of each other) 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 payload bits together with pseudo-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, as long as the payload is random. This scheme also makes it hard for Eve to correctly 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 **j** to modulate an input bit, depending on the current state (see Fig. 10(b)). As long as the current TCM state is unknown to Eve, she cannot discern 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, discern the true 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*. Note that TCM does not impact $\mathcal{F}_{\mathbf{j}}(.)$. 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.

Another attack occurs on a semi-static field if $\mathcal{M}_i$ is known. In this case, Eve can identify the true 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 payload's $\mathcal{M}_i$.

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 different sessions (e.g., Charlie-Bob and Alice-Bob sessions). 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 PHY layer. Moreover, generating one-time pads requires a good PRNG 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 [22].

V. Preamble-based PHY-Layer Identifier

Alice and Bob need to establish an identification method that is channel-independent (robust to mobility) and can be used as a means to synchronously generate the FJ traffic at PHY-layer. Such a PHY-layer identifier should also vary from one frame to another; otherwise, users and semi-static header fields become vulnerable to fingerprinting and dictionary attacks, respectively. A field is semi-static when its set of valid values is a small subset of all possible values (e.g., the 8-bit Signal field in 802.11b takes one of four possible values). When encrypted using the same **j**, those values are mapped to a fixed set of encrypted values. After eavesdropping on several frames that may have different values for that field, Eve may extract the part of **j** used to encrypt that field, launching a dictionary attack. If Alice and Bob instead try to synchronously use different parts of a pre-shared **j** for different frames, the loss of an ACK would make Alice and Bob out-of-sync. Furthermore, in the case of a packet retransmission, applying the same **j** results in the same sequence of $\mathcal{M}_M$-modulated symbols. Eve may detect retransmissions via correlating successive frames and then exclude them from the statistics used to fingerprint the session (e.g., packet size histogram). As such, we require **j** and the PHY-layer identifier to vary on a per-frame basis. In this section, we explain how we achieve this goal through generating and conveying time-rolling sender identifiers.

To generate **j**, we exploit a PRNG that is constructed based on a cryptographic hash function from the standardized family of SHA-3 algorithms (e.g., [31], [32]). The choice of input seed is very crucial for generating **j**. If it contains nothing but the secret PTK, the stream cipher **j** will always remain the same. To vary **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 [32], [33]). FCJ embeds $\mathcal{ID}$ in the frame preamble and transmits it in the clear. $\mathcal{ID}$ 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.

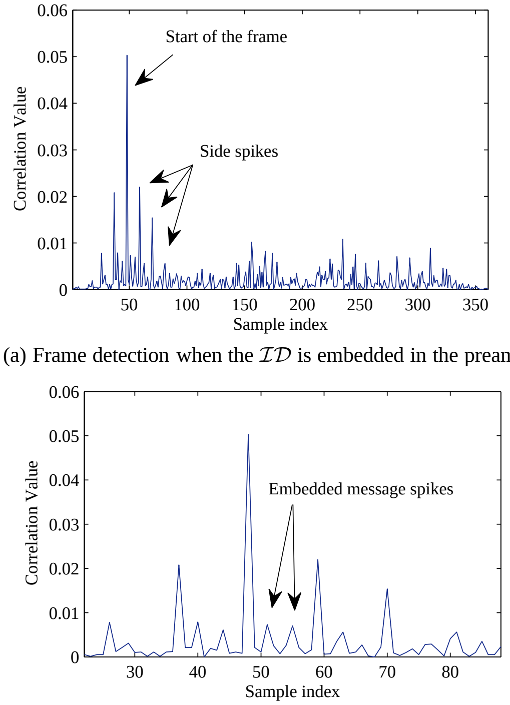
With frame-specific and time-rolling $\mathcal{ID}$s during a session, Eve will not be able to identify and track the user or correlate different frames that belong to the same session. However, Bob must be able to associate different $\mathcal{ID}$s to the same sender (e.g., Alice). We adopt an $\mathcal{ID}$ generation method similar to [34] and create a *chain* of confidential $\mathcal{ID}$s at both Alice and Bob using SHAKE256 instance of SHA-3 hash algorithm and PTK. We suggest this instance of SHA-3 because its output size can be adjusted with the size of $\mathcal{ID}$. For the first frame, they agree on an initial $\mathcal{ID}$ (e.g., during the 4-way handshake). The $\mathcal{ID}$ for subsequent frames (including retransmissions) will be the hash of the previous $\mathcal{ID}$ 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.

SHA-3 algorithms enjoy several attractive properties. First, a single-bit change in the input seed results in a completely different hash value (equivalently, **j** generated by the PRNG). Therefore, as long as $\mathcal{ID}$ is not repeated, **j** will not be repeated (similar to an ideal one-time pad), thus preventing dictionary attacks. The randomness of $\mathcal{ID}$ in our method will be discussed in Section V-B. Second, if Eve captures the hash value, she cannot use it to recover the key or the seed value used to generate **j**, i.e., it is one-way hash and robust against *chosen-plaintext* attacks [32], which are stronger than known-plaintext attacks. Third, if Eve captures some part of **j** (or the frame $\mathcal{ID}$), she cannot predict subsequent values of **j** or $\mathcal{ID}$ (i.e., robustness to generic state recovery attacks) [31], [32]. Fourth, similar to *Keccak-f[200]* [31], such a PRNG can be built in a compact core and can be implemented using bitwise Boolean operations and rotations within 200-byte memory. This makes it very resource-efficient and suitable for embedded devices with low overhead/delay requirements. Fifth, the security of **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 [31]. Altogether, the sequence **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.

A. 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 those fields, if allowed to be modified, do not provide high randomness.) 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 II) 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)) 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.



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

(b) Small but distinct spikes during the preamble due to embedded $\mathcal{ID} = 37$.

Fig. 11. $\mathcal{R}(., n)$ computed over a frame.

Fig. 11(a) is an example drawn from our experiments (Section VI) that 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. Fig. 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.

Alice represents the specific $\mathcal{ID}$ of a frame via concatenating several k-shifted versions of the Barker sequence, which is superimposed on the original preamble in the analog domain. Specifically, let $(k_1 k_2 \ldots k_L)_{10}$ 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 V). Bob is still able to detect the preamble and the $\mathcal{ID}$, as shown in Fig. 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 CSI estimation using the new preamble.

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

TABLE V

EXAMPLE OF THE CONCATENATION OF TWO BARKER SEQUENCES TO EMBED $\mathcal{ID}$ VALUE $(3\ 7)$ IN THE PREAMBLE.

5) Look up the PTK associated with the session ID and start generating **j**.

B. 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)). 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 $1, \ldots, 10$, from the start of the preamble. To avoid creating an alias of the true preamble start, Alice minimizes the repetitions of the same rotation value over preamble bits by dividing these bits into groups of $l < 11$ ($l \neq 6$) successive bits. In each group, a given rotation value may not appear more than once. 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 a sequence of l distinct 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 II, 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 $K \leq 128$ 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 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-based 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 by setting $l = 9$, in every group of l successive preamble bits, 10! different $\mathcal{ID}$s of the decimal form $(k_1 k_2 \ldots k_9)_{10}$ can be embedded. So the preamble can carry up to $(10!)^{\lfloor 64/9 \rfloor}$ distinct $\mathcal{ID}$s, which is sufficient to build a PRNG that passes the statistical tests proposed by NIST and has a resistance of about 2^{136} against state-recovery attacks [31]. Using DQPSK, we can further double the number of possible $\mathcal{ID}$s. Given this large number, FCJ can ensure the randomness required by the PRNG even when Alice employs 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.

C. 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)[4] 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 **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 $u_j^{k \bigoplus x \text{ mod } |\mathcal{M}_i|}$ 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 property is preserved (in contrast to the encryption scheme in [15]). Therefore, the BER performance 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 **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

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

j are needed to decrypt and recover the header. Payload's $\mathcal{M}_i$ is determined after the PHY header has been decoded and the rate field recovered. Eve, on the other hand, cannot correctly decode the PHY header because it is modulation-encrypted by the secret **j**. As long as the rate field in the header is unknown, Eve cannot determine $\mathcal{M}_i$ of the payload and the number of information bits that are associated with an observed symbol.

VI. Performance Evaluation

We implement *Friendly CryptoJam* in NI LabVIEW programming environment. Our LabVIEW PHY-layer libraries include the transmitter components in Fig. 4, as well as frame timing and detection, CSI 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 LabVIEW USRP driver.

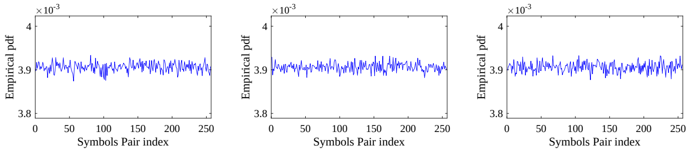
(a) Metrics. We evaluate the BER performance and preamble-related operations, such as frame detection and FO estimation, for different SNR (transmission power in the experiments) values and modulation schemes. The $\mathcal{ID}$ extraction success rate is another important metric of interest.

(b) FJ traffic. To generate **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 paper. 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., Fig. 5 and Fig. 6) and Fig. 9 and Fig. 10 for TCM-based modulation unification. The parameters of such upgrades are shown in Tables III and IV.

(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 $\mathcal{ID} = (k_1k_2k_3)_{10}$ 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, the maximum decoding delay imposed by FCJ is bounded by 10-30 symbol times, depending on $\mathcal{M}_M$.



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

Fig. 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.

A. Indistinguishability Tests

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 the 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 dependency between two successive symbols (provided that the payload 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 Fig. 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 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.

B. 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, δ_f is a controllable parameter, whereas in the experiments, it is a feature of the USRP radio oscillator.

1) 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 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

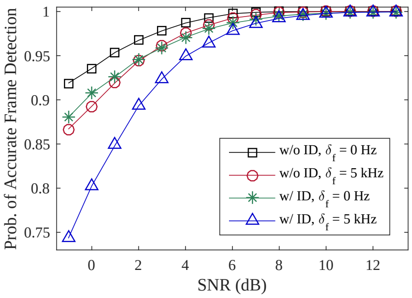


Fig. 13. Impact of embedded $\mathcal{ID} = k_1 k_2 k_3$ on frame detection (emulations).

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. Fig. 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 V-B).

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 **j**. In Fig. 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 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., $0.9985^{10} = \%98.5$, 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 Fig. 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.

2) TCM-based modulation unification: Now we study the performance of the employed TCM schemes compared to the uncoded unification scheme [15] and the default operation of 802.11 without FCJ (referred to as DF), as our benchmark. 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.

Fig. 15 and Fig. 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 (Fig. 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 Fig. 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 III.

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 Fig. 16. For example, using the two-state TCM for $\mathcal{M}_i =$ QPSK is no longer sufficient in this case (Fig. 16(b)). However, the two-state TCM is good enough when $\mathcal{M}_i =$ 16-QAM (Fig. 16(c)). As a general rule, the higher the order of $\mathcal{M}_i$ is, less complex TCM codes can be sufficient.

C. *USRP Experiments*

We now exploit our USRPs, one acting as Alice and another as Bob, to evaluate real transmissions in an indoor environment. Alice and Bob each are equipped with a 3 dB antenna and the distance between them is 2.2 m. The noise level at the receiver is about -84 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 64-QAM, 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. To study the bit errors due to only channel impairments and the noise level, we also performed a set of experiments in which we used Ettus OctoClock clock distribution module to externally synchronize the USRPs, which significantly reduces the FO.

In our empirical evaluations, we encountered a few challenges. First, the USRPs truncate 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.

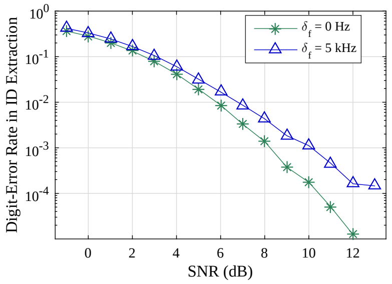
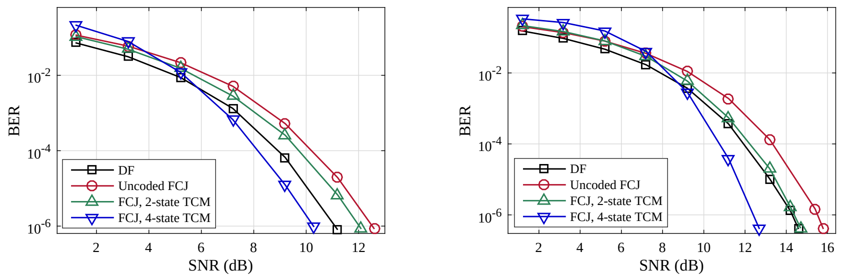


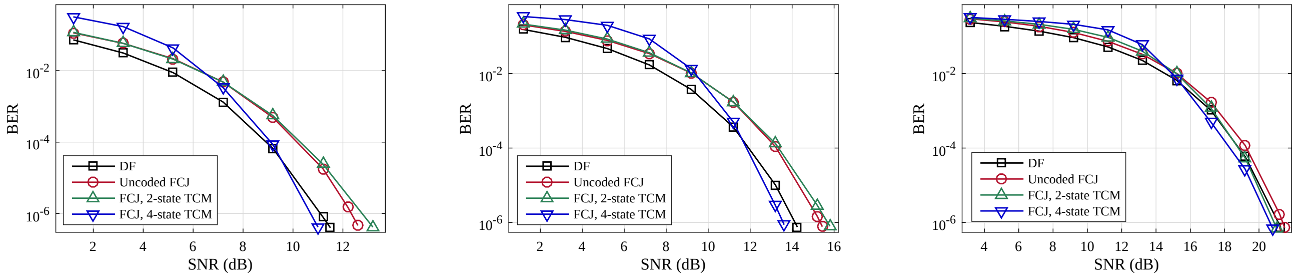
Fig. 14. Digit-error rate in $\mathcal{ID}$ detection vs. SNR (emulations).



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

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

Fig. 15. BER versus received SNR of modulation unification at Bob when $\mathcal{M}_M$ = 16-QAM (emulations).



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

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

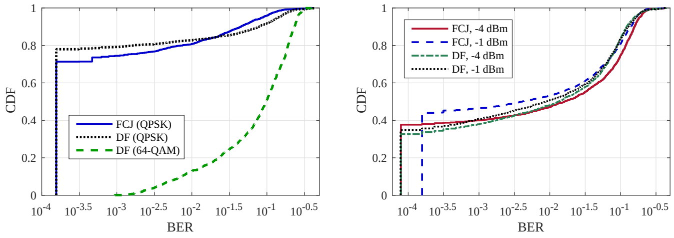
(c) $\mathcal{M}_i$ = 16-QAM

Fig. 16. BER versus received SNR of modulation unification at Bob when $\mathcal{M}_M$ = 64-QAM (emulations).

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 Alice 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 power varies from one frame to another.

The second challenge is inaccurate FO estimation when OctoClock is not used. As reported in [15], [16] and discussed in Section IV-B, 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 maintained a coarse estimate of δ_f based on previous transmissions and compensated for it before performing the normal FO estimation in each run. However, the estimate may still be inaccurate and result in high BER. When averaging the BER of several transmissions, a (small) subset of transmissions with high BER values (e.g., 10^{-1}) dominates the rest of transmissions whose BER values are low. Instead, 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. (In CBM [16], a two-pass mechanism is employed to significantly reduce the errors in FO correction and phase tracking. This mechanism is not implemented here.)

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). 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) before taking the absolute value (similar to the method in [25]). 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.



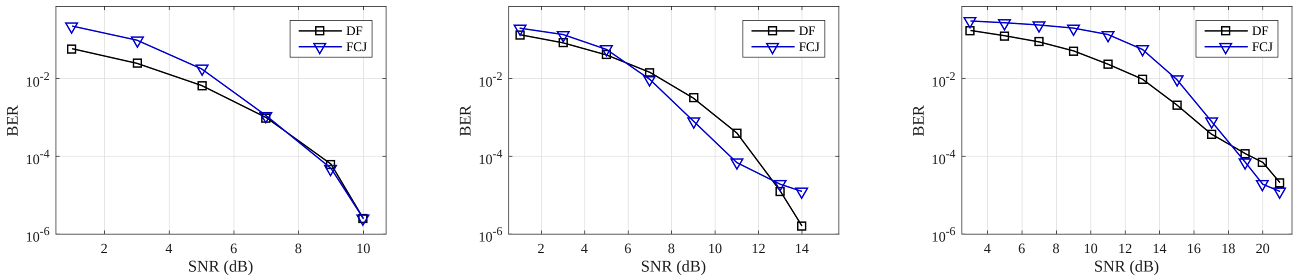
(a) $\mathcal{M}_i$ = QPSK and Tx power = -8 dBm (-1 dBm for 64-QAM)

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

Fig. 17. Empirical cumulative distribution function of BER (USRP results).

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 Fig. 17, we compare the performance of FCJ with the four-state TCM to the one of the default scheme when OctoClock is not used. In these experiments, the SNR was so



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

Fig. 18. USRP results: BER versus received SNR of modulation unification at Bob when $\mathcal{M}_M$ = 64-QAM and 4-state TCM is used ($\delta_f \approx 0$).

high (~ 35–40 dB) that the decoding errors were often due to inaccurate FO estimation only, even when Alice's transmission power is set to its lowest values. Fig. 17(a) depicts the BER distribution when $\mathcal{M}_i$ = QPSK and $\mathcal{M}_M$ = 64-QAM. 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 IV-B). 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 Fig. 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 more error-free transmissions.

Finally, we study the BER performance with varying SNR when OctoClock is used. In addition to the relatively fixed noise in the environment, we introduce Gaussian noise at Bob right after the channel estimation/equalization so as to vary the received SNR. The results in Fig. 18 show that the proposed 4-state TCM scheme is sufficient to maintain the default BER.

VII. Related Work

Several upper-layer techniques, such as padding, traffic morphing [11], and packet features masking at the application layer [13], have been proposed to prevent the leakage of SCI by altering the true traffic statistics. These techniques, however, trade off higher traffic overhead for increased privacy. In fact, most of the existing techniques and in particular the padding techniques have been shown to be insufficient in thwarting classification attacks, despite their high bandwidth overhead. Dyer *et al.* [12] 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. To reduce the overhead, traffic reshaping at the MAC layer [10] is used to dynamically distribute the traffic among several virtual MAC interfaces; hence reshaping the statistical traffic profile of each of the interfaces. However, even if the devices support multiple virtual MAC addresses, this method requires modifying protocols of multiple layers. Furthermore, none of the above techniques can hide lower-layer fields such as the modulation scheme and the data rate. FCJ, however, obfuscates packet lengths and the total traffic volume (among others) without imposing high overhead or modifying higher-level protocols. For example, upgrading BPSK-modulated frames to 64-QAM-modulated frames can translate to 600% increase in the total traffic volume for Eve.

A number of PHY-layer protection schemes have also been proposed. Scrambling can be used to securely obfuscate the input bit sequence. However, this does not obfuscate the channel-dependent modulation scheme. Directional antennas try to shrink the vulnerability zone by steering in the direction of the legitimate receiver. Yet, the LOS from Alice to Bob is vulnerable to wiretapping, in addition to side lobes. Also, in some circumstances, these techniques may fail to provide directionality (e.g., see [35]). Other signal precoding techniques such as beamforming and orthogonal blinding (e.g., [17]) have also been shown to be insufficient (see Section I-A).

More recently, trellis-based encoders has been employed for providing data confidentiality [36], [37] and rate hiding [16]. In [16], the authors generalize conventional TCM to simultaneously hide the rate information/modulation scheme and boost the system resiliency (up to 8 dB) against interference. In order to eliminate the dependency among successive coded symbols, the authors proposed cryptographically 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. This can result in a large decoding delay whereas the delay of FCJ is less than $10 - 30$ symbols. More importantly, the authors encrypt the header but without providing any alternative for the sender identification.

VIII. Conclusions

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) 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, 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 friendly jamming sequence generation and session-key lookup at PHY layer. Second, modulation-aware encryption is used to perfectly secure plaintext headers and readily encrypted payload. Third, 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. 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.

Acknowledgment

This research was supported in part by the Army Research Office (grant # W911NF-13-1-0302) and in part by the NSF (grants # IIP-1265960 and CNS-1409172). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of ARO or NSF.

References

[1] 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.

[2] 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.

[3] 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.

[4] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, "Who do you sync you are? smartphone fingerprinting via application behaviour," in *Proc. 6th ACM Conf. Security and Privacy in Wireless and Mobile Networks (WiSec'13)*, Budapest, Hungary, 2013, pp. 7–12.

[5] 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.

[6] J. S. Atkinson *et al.*, "Your WiFi is leaking: Inferring user behaviour, encryption irrelevant," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC'13)*, Apr. 2013, pp. 1097–1102.

[7] 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.

[8] J. Freudiger, "How talkative is your mobile device? an experimental study of Wi-Fi probe requests," in *Proc. 8th ACM WiSec Conf.*, New York City, USA, Jun. 2015.

[9] 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.

[10] F. Zhang *et al.*, "Thwarting Wi-Fi side-channel analysis through traffic demultiplexing," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 86–98, Jan. 2014.

[11] C. V. Wright, S. E. Coull, , and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proc. Network Distributed Syst. Security symp. (NDSS'09)*, Feb. 2009, pp. 237–250.

[12] 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.

[13] A. Iacovazzi and A. Baiocchi, "Internet traffic privacy enhancement with masking: Optimization and tradeoffs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 353–362, Feb. 2014.

[14] B. Greenstein *et al.*, "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.

[15] H. Rahbari and M. Krunz, "Friendly CryptoJam: A mechanism for securing physical-layer attributes," in *Proc. 7th ACM WiSec Conf.*, Oxford, United Kingdom, Jul. 2014, pp. 129–140.

[16] T. D. Vo-Huu and G. Noubir, "Mitigating rate attacks through crypto-coded modulation," in *Proc. ACM MobiHoc'15 Conf.*, Hangzhou, China, Jun. 2015, pp. 237–246.

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

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

[19] S. Gollakota *et al.*, "They can hear your heartbeats: Non-invasive security for implantable medical devices," in *Proc. ACM SIGCOMM'11 Conf.*, Toronto, Ontario, Canada, Aug. 2011, pp. 2–13.

[20] 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.

[21] 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.

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

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

[24] 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.

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

[26] "IEEE Std 802.11a-1999," *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, 1999.

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

[28] E. Specht, "The best known packings of equal circles in a square," Last updated: 18-10-2013. [Online]. Available: http://goo.gl/Fx4B0a

[29] 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.

[30] 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.

[31] 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.

[32] I. Dinur *et al.*, "Cube attacks and cube-attack-like cryptanalysis on the round-reduced Keccak sponge function," *Advances in Cryptology–EUROCRYPT*, pp. 733–761, 2015.

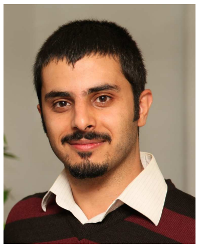
[33] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Duplexing the sponge: single-pass authenticated encryption and other applications," in *Selected Areas in Cryptography*. Springer, 2011, pp. 320–337.

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

[35] M. Buettner *et al.*, "A phased array antenna testbed for evaluating directionality in wireless networks," in *Proc. 1st ACM Int. Workshop Syst. Evaluation for Mobile Platforms (MobiEval'07)*, San Juan, Puerto Rico, 2007, pp. 7–12.

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

[37] ——, "Secure encoder designs based on turbo codes," in *Proc. IEEE ICC'15 Conf.*, London, UK, Jun. 2015, pp. 4315–4320.



Hanif Rahbari received his PhD degree in electrical and computer engineering from the University of Arizona in 2016. He is currently a research specialist at the University of Arizona. Before his PhD, he received his BSc in information technology from Sharif University of Technology and his MSc in computer networks from AmirKabir University of Technology, Iran. His research interests include network (wireless) security, wireless communications, hardware experimentation, dynamic spectrum access networks, and multimedia networking.

marwan Krunz [S’93, M’95, SM’04, F’10] received his Ph.D. degree in electrical engineering from Michigan State University in 1995. He is the Kenneth VonBehren Endowed Professor of electrical and computer engineering and the site co-director of the NSF Broadband Wireless Access and Applications Center. His research interests are in wireless communications and networking, with emphasis on resource management, adaptive protocols, and security issues. He has published more than 225 journal articles and peer-reviewed conference papers. He received numerous awards, including the 2012 IEEE TCCC Outstanding Service Award and the NSF CAREER Award. He was an Arizona Engineering Faculty Fellow (20112014) and an IEEE Communications Society Distinguished lecturer (2013 and 2014). He has served on the editorial boards of several IEEE journals. He has been General and Program Chair for numerous conferences, including INFOCOM’04, SECON’05, WoWMoM’06, and Wisec’12.