Friendly CryptoJam: A Mechanism for Securing Physical-Layer Attributes

Hanif Rahbari
rahbari@email.arizona.edu

Marwan Krunz
krunz@email.arizona.edu

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

ABSTRACT

The broadcast nature of wireless communications exposes various "transmission attributes," such as the packet size, the 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's privacy and confidentiality by performing traffic analysis and classification, whereas an active attacker exploits captured attributes to launch selective jamming/dropping attacks. This so-called PHY-layer security problem is present even when the payload is encrypted. For example, by determining the modulation scheme, the attacker can estimate the data rate, and hence the payload size, and later use it to launch traffic classification or selective rate-adaptation attacks.

In this paper, we propose *Friendly CryptoJam*, a novel approach that combines analog-domain friendly jamming and modulation-level encryption. *Friendly CryptoJam* decorrelates the payload's modulation scheme from other transmission attributes by always "upgrading" it to the highest-order modulation scheme supported by the system (a concept we refer to as *modulation unification*) using a secret pseudo-random sequence. Such upgrade is a form of transmitter-based friendly jamming. At the same time, modulation symbols are encrypted to protect unencrypted PHY-layer fields (*modulation encryption*). To generate and sync the secret sequence, an efficient message embedding technique based on Barker sequences is proposed, which exploits the structure of the preamble and overlays a frame-specific seed on it. We study the implications of the scheme on PHY-layer functions through simulations and USRP-based experiments. The results confirm that *Friendly CryptoJam* is quite successful in hiding the targeted attributes, at the cost of a small increase in the transmission power.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

WiSec'14, July 23–25, 2014, Oxford, UK.

http://dx.doi.org/10.1145/2627393.2627415.

Keywords

PHY-layer security; side-channel information; IEEE802.11; modulation encryption; friendly jamming; preamble; USRP

1. INTRODUCTION

As we continue to depend on the rapidly expanding wireless ecosystem, we are challenged with serious threats related to user privacy, data confidentiality, and system availability. Using commodity radio hardware, 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-layer header) must be transmitted in the clear for correct protocol operation, device identification, and reduced complexity. For example, 802.11i, the primary security amendment of 802.11, provides confidentiality only for the MAC-layer payload, as well as integrity for this payload and its header [2]. Even if we hypothetically encrypt the entire frame, the transmission is not completely immune. In fact, an adversary can still perform low-level RF and traffic analysis, and estimate several transmission attributes, including packet sizes, modulation scheme, inter-packet times, and traffic directionality.

Transmission attributes can be correlated to create "fingerprints" of intercepted communications, which can be used to determine user identities, content, type, or stage of a communication. Depending on whether the frames are entirely encrypted or not, leaked attributes may consist of only side-channel information or may also contain lower-layer fields. Side-channel information refers to statistical traffic features, such as packet size distribution, inter-packet time sequence, and data rate (traffic volume). For example, by eavesdropping on an 802.11 wireless LAN traffic, an adversary (Eve) can determine the type of user activities with 80% accuracy (after only five seconds of eavesdropping) [4, 19] or the content of search query words [6]. Upper-layer traffic manipulation techniques, such as packet padding and traffic reshaping [18], aim at obfuscating side-channel information at the cost of traffic overhead [7]. Even then, they cannot completely hide all such information. For instance, the data rate and the length of a (re)transmitted frame can be estimated through RF analysis and by inspecting the PHY frames. Using an off-the-shelf device such as a vector signal analyzer (VSA), Eve can detect the payload's modulation scheme of even an entirely encrypted frame. This type of side-channel information has not been studied in the security literature. Eve would then estimate the data rate, and

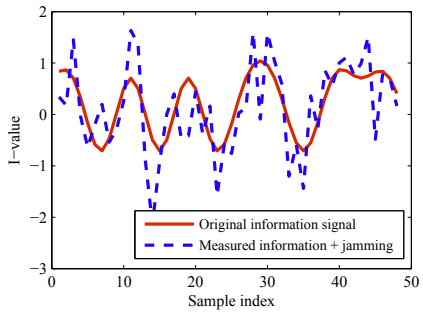
hence determine the packet size in bytes. Statistics of the packet size and total traffic volume [7] are key parameters in traffic analysis and classification. In fact, there is no effective and resource-efficient countermeasure to obfuscate the total traffic volume, which can be exploited independently for traffic classification [7].

Eve can also exploit unencrypted fields in the PHY and MAC headers to expose the privacy of a user [4, 19], or to identify the user and launch sophisticated active attacks. These lower-layer fields include the source and destination MAC addresses, data rate, modulation scheme, frame length (duration), the number of space-time streams of a MIMO system, and others. For example, Noubir *et al.* [14] demonstrated a reactive jammer that can significantly hamper the network throughput by intercepting the rate field of a frame and accordingly deciding whether to jam the rest of the frame. If a packet is not correctly decoded as a result of jamming, the transmitter (Alice) will mistakenly assume a poor channel and will lower the rate. Our approach belongs to the so-called PHY-layer security, which complements conventional data encryption and upper-layer traffic manipulation techniques by providing protection for the entire frame at the PHY-layer. In this paper, we focus on preventing the exposure of unencrypted header fields and the payload's modulation scheme, hence countering rate-adaptation and packet-length-based classification attacks.

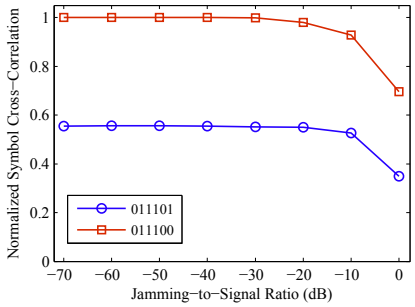
Friendly jamming (e.g., [3,8,9]) is probably the most prominent method for PHY-layer anti-wiretapping. It tries to degrade Eve's channel without impacting the channel quality at the intended receiver (Bob). This is done using MIMO techniques or by having relay nodes transmit a jamming signal that is harmless (friendly) to Bob. A mixture of the information and jamming signals could also be viewed as an encrypted signal. However, three fundamental issues limit the practicality of this approach. First, if Eve is equipped with multiple antennas, she can cancel out a transmitter-based jamming signal [15,16]. For example, Schulz *et al.* [15] exploited a known part of the transmitted signal (e.g., frame preamble) to show that Eve can estimate the precoding matrix for the friendly jamming signal and nullify its effects. This matrix is supposed to be secret and unique, as it depends on the CSI for the Alice-Bob channel, i.e., it represents a signature of the Alice-Bob channel.[1] This known-plaintext attack can thwart any security scheme that relies on prefiltering (precoding) data at Alice. Furthermore, the uniqueness of the Alice-Bob CSI has been shown to be invalid in poor scattering environments [11]. Specifically, a few adversaries located just several wavelengths away from Bob (Bob's *vulnerability zone*) can cooperatively reconstruct the link signature for the Alice-Bob channel.

Second, transmitter- and receiver-based friendly jamming (such as in [9]) are still vulnerable to cross-correlation attacks on (unencrypted) semi-static header fields, where the field can take one of a few possible values. In such attacks, Eve can detect the start of a frame, even if it is combined with a jamming signal [10]. By knowing the underlying header format (i.e., where each field is supposed to start), Eve can locate the starting time of the targeted field in the header. Figure 1(a) shows an example of the measured in-phase (I) values of a complex sequence that represents the

[1]Once the precoding matrix is approximated, Bob multiplies the received signal by the inverse of this matrix and cancels out the friendly jamming signal.



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



(b) Cross-correlation between received (information + jamming) signal and a possible value for the information signal vs. JSR (011100 is the correct value).

Figure 1: Cross-correlation attack on an information signal combined with a friendly jamming signal.

modulated value of a semi-static header field plus jamming signals. This field is probably not decodable at Eve because of friendly jamming. However, by correlating the modulated symbol of each possible value of this field with the received signal, Eve can guess the actual field value. In general, this cross-correlation attack can be formulated as a composite hypothesis testing. We show a simple example in Figure 1(b), which depicts the cross-correlation between two possible field values (011100 is the true value and 011101 is another possible value) and the received information + jamming signal (Figure 1(a)) as a function of the jamming-to-signal ratio (JSR). Each point is the mean of 100 simulation trials. The plots show that Eve can successfully determine the true value even when the jamming power is as high as the signal power.

Third, depending on the channel coefficients, the jamming power may need to be even higher than the information signal power to achieve non-zero secrecy capacity [8]. This motivates the need for a more robust security framework that provides protection to all lower-layer fields at a reasonably low power expenditure.

Scheme Overview

In this paper, we propose *Friendly CryptoJam* (CJ, for short), a form of friendly jamming but with the information and jamming signals intermixed before transmission. Essentially, this intermixing makes CJ a form of robust modulation-level encryption. CJ completely encrypts a frame right after the digital modulation phase and before the frame is transmitted over the air. In contrast to classic friendly jamming techniques, a single antenna is used to transmit both the

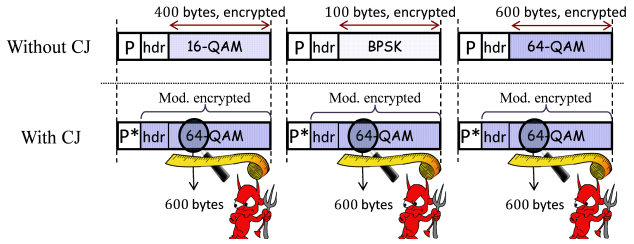


Figure 2: Example of using *Friendly CryptoJam* to hide the modulation scheme (and packet size) of three packets with the same duration. Headers and payload are modulated-encrypted and upgraded without changing the data rate. Under CJ, a seed ($\mathcal{ID}$) is overlaid on the original frame preamble (P), leading to a new preamble ($P^* = P + ID$).

information and jamming signals. The key idea in CJ is to first encrypt the modulated symbols. This is done by replacing each modulated symbol by another one according to a pseudo-random secret jamming sequence (i.e., the friendly jamming signal). Our specific encryption function leaves Eve with the highest entropy no matter what signal she observes, i.e., it achieves perfect secrecy. Then, while keeping an eye on the BER and without increasing the data/information rate, the encrypted modulated symbols are mapped (upgraded) to the constellation map of the highest-order modulation scheme supported by the system, using other parts of the same secret sequence that was used in the encryption phase. This secret sequence, hereafter called *bogus traffic*, is generated using a partially secret seed, which is independent of the link signature, i.e., independent of Bob's location, and is robust to known-plaintext attacks.

The combination of modulation encryption and modulation upgrade prevents any classification based on the modulation scheme (hence packet size and rate cannot be reliably determined), obfuscates the total traffic volume with little traffic overhead, and also keeps unencrypted fields and retransmissions indistinguishable (see the example in Figure 2). Our energy-efficient modulation upgrade approach, called *modulation unification*, preserves the BER with less than 2 dB increase in the transmission power. The design of CJ also takes into account issues such as interference and packet losses, retransmissions, channel estimation, and frequency offset estimation, while maintaining synchronization of the bogus traffic generation processes at Alice and Bob. In contrast to conventional (digital-domain) encryption, the encryption in CJ is modulation-aware. CJ is also transparent to upper layer.

One important challenge in designing *Friendly CryptoJam* is how to change the bogus traffic on a per-frame basis (to prevent a dictionary attack). In particular, relying on a single (long) PN sequence to generate the bogus traffic is prone to synchronization errors due to ACK packet losses, for example. To ensure consistency in the generation of per-frame bogus traffic at Alice and Bob, Alice conveys a frame-specific seed (e.g., packet number) whose modulated value is superposed onto the known frame preamble. This same seed is also concatenated to the session key and fed into an appropriately designed pseudo-random number generator (PRNG) to generate the bogus traffic. The seed is also responsible for sender identification (to pull up Alice's security credentials at Bob), since the MAC address is encrypted. However, superimposing any signal on the preamble may degrade the preamble's crucial functions (e.g., frame detection, frequency offset estimation). To prevent that, we use cyclically rotated Barker sequences (which exhibit low cross-correlations with the preamble) to construct a seed-bearing signal. This signal will have two identical parts (similar to the preamble), so frequency offset estimation can operate as usual. Bob then extracts the embedded seed and uses it for channel estimation and bogus traffic generation. We extensively evaluate the different components of CJ by simulations. We also use a USRP-based platform to implement and experimentally verify that CJ is highly immune to PHY-layer eavesdropping.

Paper organization– We provide background material on 802.11 PHY-layer header and preamble functions in Section 2. In Section 3, we describe the attack model and state our assumptions. Modulation unification and encryption are presented in Section 4, followed by bogus traffic generation, shifted Barker sequences, and practical challenges of message embedding in Section 5. We present our simulations and USRP experiments in Section 6. Section 7 contains a more detailed literature review. Finally, we conclude the paper in Section 8.

2. PHY-LAYER ATTRIBUTES AND PREAMBLE IN 802.11 SYSTEMS

(1) PHY-layer header fields. 802.11 standards specify the frame length and the transmission rate in the PHY-layer header. The transmission rate is typically adjusted based on channel conditions, resulting in different frame durations (in seconds) for the same payload. In 802.11b/g, the data rate and the modulation scheme (DBPSK, DQPSK, CCK, or PBCC) 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 MCS field in 802.11n is similar to the rate field in 802.11a. All 802.11 standards specify a "length" field, which represents the payload size in octets (for 802.11a/n) or in milliseconds (for 802.11b).

(2) Frame detection and frequency offset estimation. Each PHY-layer header is preceded by a preamble, which is used to detect the start of a frame (frame detection), frequency offset (FO) estimation, and channel estimation. This preamble consists of several repetitions of a publicly known pattern. The process of FO estimation requires detecting the arrival of at least two of the repetitions. A frequency offset δ_f creates a linear phase displacement $\varphi(t)$, which accumulates over time as follows:

$$\varphi(t) = 2\pi\delta_f\, t. \tag{1}$$

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 may subtract 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. In large packets, this residual error eventually shifts a data symbol to a wrong point on the constellation map, causing a demodulation error. Another reason for using a preamble is channel estimation. After compensating for FO, Bob compares the known pattern in the preamble with its received value to estimate the channel parameters (CSI).

In 802.11b systems, a scrambled version of a 128-bit preamble is modulated (spread) using an 11-chip Barker sequence (Table 1).[2] The autocorrelation of a Barker sequence at non-zero lags is very low (orthogonality property), which can be exploited for frame detection and timing. Let $\mathcal{A}(k)$ be the autocorrelation at lag k, $1 \leq k < N$. Then,

$$\mathcal{A}(k) = \Big| \sum_{j=1}^{N-k} a_j a_{j+k} \Big| \leq 1 \quad (2)$$

where $\{a_j : j = 1, \ldots, N\}$ is a Barker sequence. The receiver correlates this known sequence with the received signal sequence r and computes the square of the cross-correlation value, denoted by $\mathcal{R}(n)$:

$$\mathcal{R}(n) = \Big| \sum_{j=1}^{N} a_j^* r_{j+n-1} \Big|^2. \quad (3)$$

$\mathcal{R}(n)$ is expected to peak when the n-th sample of r marks the beginning of the transmitted Barker sequence.

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 1: DSSS signal spreading based on an 11-chip Barker sequence for DBPSK modulation.

(3) Detection of lower-layer fields. The preamble and the PHY header are both transmitted at the lowest supported rate.[3] The MAC header is considered part of the data payload, so it may be transmitted at possibly a different rate. The security amendment 802.11i only provides integrity for the MAC header. The preamble, PHY, and MAC headers are all transmitted in the clear, allowing an adversary to intercept them. Detection of the payload's modulation is another way to obtain an estimate of the data rate, packet size, or packet type (e.g., control or data packet). A modulation scheme is usually associated with two or three data rates, with different coding rates. For example, in 802.11a, 16-QAM is used when the data rate is either 24 or 36 Mbps. Hence, by determining the modulation scheme, it is rather easy for the adversary to guess the data rate. Moreover, because control packets are often transmitted using the most robust modulation scheme (e.g., one with lowest required SINR threshold), exposure of this scheme facilitates the discovery of control transmissions.

3. SYSTEM MODEL

Consider a wireless link that consists of a transmitter (Alice) and a receiver (Bob). The link operates in the presence of an eavesdropper (Eve). Alice and Bob first create a shared *pairwise transient key* (PTK) through the EAPOL 4-way handshake of 802.11i [2]. PTK is used to encrypt unicast payloads, but as explained later we also use it to generate bogus traffic at the PHY layer. Every node maintains a table of the PTKs and the session IDs of other handshaked neighbors in the network.[4] We assume Alice and Bob are each equipped with a single antenna. They exploit knowledge of the preamble and PHY-header format to customize *Friendly CryptoJam*, but keep the original preamble and frame content, including the header(s), intact. Any preamble modification will be in the form of superposing a signal on the original preamble rather than introducing a completely new preamble. This way, customizing the design to other systems with a known 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 and PHY format of 802.11b. For simplicity, we consider BPSK, QPSK, 16-QAM, and 64-QAM modulation schemes for the payload.

Figure 3 shows a schematic view of Alice's PHY layer and the insertion points of the proposed *Friendly CryptoJam* components, which include modulation encryption (point 1), modulation unification (point 2) and message, frame and session IDs embedding (point 3). Starting with the MAC header, once the payload arrives at Alice's PHY layer, Alice determines PHY-header fields, including the appropriate modulation scheme for this payload, based on a rate-adaptation algorithm. The preamble, PHY-header, and payload are then scrambled and modulated before being passed to the pulse filters and transmitted over the air. Bob, on the other hand, detects the preamble and extracts the frame and session IDs embedded in it to regenerate the bogus traffic and estimate the CSI. Next, he recovers and decrypts the header to extract the modulation field of the payload, which is used to recover and decrypt the rest of the frame.

Eve knows the frame structure and protocol. She can be a passive eavesdropper or a reactive jammer that jams based on her analysis of early portions of the frame. The types of attacks that can be launched by Eve include cross-correlation attacks (e.g., Figure 1(b)), rate-adaptation attack [14], known-plaintext [15] attack, and any data-rate-based traffic classification attack. We also allow Eve to be equipped with multiple antennas. She can also perform RF analysis, correlation, and modulation detection. We further assume that upper layers may employ a traffic classification mitigation technique (e.g., traffic morphing or random padding), but do not pad a packet to a set of fixed sizes (e.g., pad to MTU). For a given packet size, lower-modulation orders generate longer frame durations. So if the frames contain packets of the same size after padding, the duration of their corresponding modulation-unified signals can reveal the actual modulation order.

4. FRIENDLY CRYPTOJAM

In this section, we introduce the first two components of CJ, i.e., modulation unification and encryption, which are used to mask the entire frame and protect lower-layer fields. *Friendly CryptoJam* is essentially a form of friendly jamming that encrypts modulated symbols. However, it is different from conventional cryptography, friendly jamming, and scrambling in that it is applied right after digital modulation and before the up-conversion and antenna transmission. Conventional cryptography digitally encrypts (blocks

[2]Scrambling transforms an all-one preamble bit sequence into a sequence of zero's and one's. Methods like [20] are used to detect the zero's and change them to one's.

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

[4]MAC address, which comes after the PHY header, is encrypted, and hence cannot be used to retrieve the corresponding PTK. Session ID will be used instead to distinguish between different transmitters.

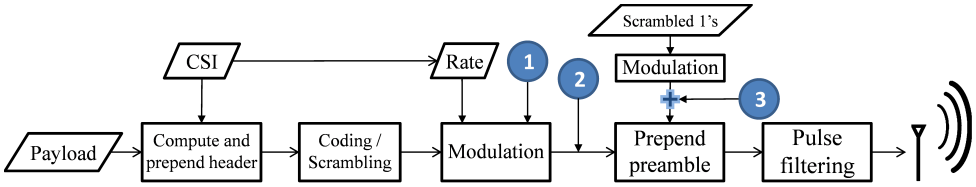


Figure 3: Transmission chain at Alice under CJ. Insertion points (1), (2), and (3) refer to modulation encryption, modulation unification, and message embedding within the preamble.

of) bits. Scrambling used in 802.11 is another digital-domain operation. Friendly jamming, on the other hand, is the concurrent transmission of analog noise from one or more antenna(s) other than the one used for the information signal.

In the following, we first explain our modulation unification and encryption scheme, assuming that the bogus traffic sequence is already available at both Alice and Bob. The problem of securely generating and synchronizing this sequence will be explained in Section 5.

4.1 Modulation Unification

The modulation scheme used for the frame payload should always look the same for the eavesdropper (Eve) so as to prevent any signal classification and modulation detection. To achieve that, we upgrade the payload's modulation scheme to the highest-order scheme supported by the underlying system (i.e., 64-QAM in our setup), which may result in a transmission rate that is higher than what the channel allows. In this upgrade, the original modulation symbols are embedded in the constellation map of the highest-order modulation scheme but the actual channel-dependent data rate remains unchanged.

Increasing the modulation order resembles the superposition of the constellation points of two colliding packets, i.e., as if the two packets are combined in the digital modulation space prior to transmission. One of these packets can be a digitally modulated version of conventional friendly jamming (i.e., an artificial collision). In a collision, the I and Q components of the two superimposed complex symbols are added to create a higher-order constellation map. For example, the superposition of two QPSK-modulated frames (each may contain four possible constellation points) results in a new constellation map with nine possible (I,Q) pairs (see Figure 4). Inspired by this and the fact that a collision is not recoverable if both packets are unknown, we combine Alice's frame (except the preamble) with the modulated bogus traffic but in a way that meets our uniformity and throughput requirements. The original preamble is required for performing the specific functions mentioned in Section 2. Because its content and modulation scheme are already known to Eve, such an upgrade is not beneficial for the preamble. An uncontrolled superposition of two signals (i.e., a collision) may result in a new modulation point that does not belong to any of the modulation schemes supported by the system and further may disclose the original modulation points (hence, the modulation schemes). In contrast to that, we propose a particular mapping from any of the available payload modulation schemes to the highest modulation order that is already supported by the system.

However, a higher modulation order can be more susceptible to demodulation errors. To illustrate, let the bogus traffic be $\mathcal{B}$ and let $\mathcal{F}_{\mathcal{B}}(m_i)$ be a mapping that is known for both

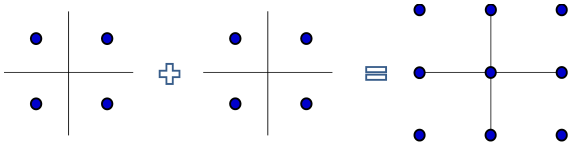


Figure 4: Combining (artificially colliding) of two QPSK-modulated signals results in a 9-symbol constellation map.

Alice and Bob and upgrades the ith modulation scheme m_i, $i = 1, \ldots, M$, to the highest-order modulation scheme m_M. The minimum distance between the symbols in the constellation of m_i, denoted by $d_{min,i}$, specifies the probability of a demodulation error at a given SNR value. This $d_{min,i}$ generally decreases with the increase in i. Table 2 depicts $d_{min,i}$ for the 802.11a system after taking into account modulation-dependent normalization factor K_{MOD} [1]. K_{MOD} is a coefficient this is multiplied by the (I,Q) values to achieve the same average power for all modulation schemes. To maintain the same $d_{min,i}$ after mapping m_i to m_M, i.e., achieve the same BER, two neighboring points in m_i should not be mapped to very close points in m_M, as much as possible. At the same time, all the resulting constellation points of m_M as observed by Eve must be equally probable (as when a random information sequence is modulated using m_M). Otherwise, Eve may guess m_i by performing statistical analysis. In the following, we define a mapping $\mathcal{F}_{\mathcal{B}}(m_i)$ based on $\mathcal{B}$ that achieves both of the above design requirements. Alice upgrades her modulation scheme m_i only when $i < M$. In the case when $i = M$, $\mathcal{F}_{\mathcal{B}}(m_M)$ is just an affine function.

To upgrade m_i, our scheme defines $|m_i|$ equal-size and non-overlapping sets of constellation points in m_M, where $|m_i|$ is the number of constellation points in m_i. Each distinct constellation point (or symbol) in m_i, denoted as s, is mapped to one of these predefined target sets. The selection of a point inside a given target set depends on $\mathcal{B}$. For a given s, Alice needs $\log_2 |m_M| - \log_2 |m_i|$ bits of $\mathcal{B}$ to select one of the $\frac{|m_M|}{|m_i|}$ points within a target set. So Alice picks the first $\log_2 \frac{|m_M|}{|m_i|}$ bits in $\mathcal{B}$ for the first symbol to be transmitted, the next $\log_2 \frac{|m_M|}{|m_i|}$ bits for the second symbol, and so on. The same bits in $\mathcal{B}$ always point to the same constellation point within a target set, i.e., $\mathcal{F}_{\mathcal{B}}(m_i)$ is static. This ensures that the resulting constellation points are equally probable, assuming that the bits in $\mathcal{B}$ are uniformly and randomly distributed. We rely on a cryptographic hash function like SHA-2 to generate such a secret $\mathcal{B}$ (see Section 5 for details).

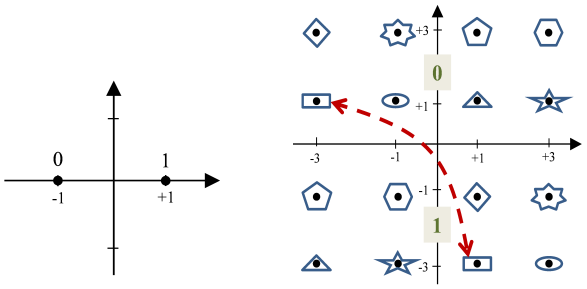
During the decoding process, Bob knows $\mathcal{B}$. He needs to obtain the original data symbol s. Let b be the decimal representation of the bits in $\mathcal{B}$ that correspond to s. Based on b, Bob selects $|m_i|$ candidate points in m_M, denoted by C_b, for the unknown data symbol. Each candidate point belongs to one of the target sets. Therefore, Bob's job is

i	m_i	$K_{MOD}[1]$	$d_{min,i}$	$d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)$	$b_{4,i} \stackrel{\text{def}}{=} \frac{d_{min,i}}{d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)}$
1	BPSK	1	2	$8/\sqrt{21}$	$\sqrt{21}/4$
2	QPSK	$1/\sqrt{2}$	$2/\sqrt{2}$	$8/\sqrt{42}$	$\sqrt{21}/4$
3	16-QAM	$1/\sqrt{10}$	$2/\sqrt{10}$	$8/\sqrt{42}$	$\sqrt{4.2}/4$
4	64-QAM	$1/\sqrt{42}$	$2/\sqrt{42}$	$2/\sqrt{42}$	1

Table 2: Parameters of the optimal mapping from the modulation schemes in 802.11a to 64-QAM.

x \ y	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

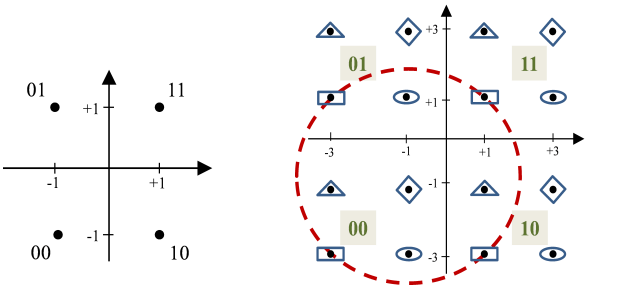
Table 3: Modulation encryption for QPSK (resulting z values for various (x, y) pairs).



(a) BPSK modulation.

(b) $C_0, \ldots, C_7$ in 16-QAM. For example, the points in the rectangles belong to C_0.

Figure 5: Optimal mapping from BPSK to 16-QAM.



(a) QPSK modulation.

(b) $C_0, \ldots, C_3$ in 16-QAM. For example, the points in the rectangles on the dashed circle belong to C_0.

Figure 6: Optimal mapping from QPSK to 16-QAM.

to find the most likely symbol in C_b, given the observed symbol, which is a classical demodulation process. Because the modulation scheme of the PHY header portion is known a priori, Bob is able to first decode the header and obtain the original payload's modulation scheme, and then demodulate the rest of the frame.

Next, we explain an optimal strategy for selecting the target sets and mapping the original constellation points to these sets. In here, optimality is taken w.r.t. maximizing the minimum distance between symbols. Let $d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)$ be the minimum distance between any two points in $C_b, \forall b = 0, \ldots, |m_M|/|m_i| - 1$. Bob uses a standard demodulation technique (e.g., ML) over the constellation points in C_b to decode s. An optimal mapping for m_i maximizes $d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)$. The following formulation solves for such a mapping:

$$\begin{aligned} &\max_{C_b} \quad d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big) \\ \text{s.t.} \quad &|C_b| = |m_i|, \forall b = 0, \ldots, |m_M|/|m_i| - 1 \\ &\bigcup_{b=0}^{|m_M|/|m_i|-1} C_b = m_M \\ &\bigcap_{b=0}^{|m_M|/|m_i|-1} C_b = \emptyset. \end{aligned} \tag{4}$$

Figures 5 and 6 illustrate an optimal mapping from BPSK and QPSK to 16-QAM, respectively. In part (b) of each figure, the constellation points that belong to the same set C_b are enclosed in the same shape. The bits that correspond to any given constellation point in m_M consist of the user payload bits (MSBs) and bogus bits b (LSBs). On the constellation map of m_M, the MSBs specify the region (e.g., quadrant), while the LSBs specify a point inside that region. For example, in QPSK the payload data bits specify one of the quadrants in 16-QAM and b specifies a point within that quadrant.

However, the optimal mapping may not fully satisfy the initial $d_{min,i}, \forall i \in 1, \ldots, M$. Let $b_{M,i}$ be the ratio between $d_{min,i}$ and $d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)$. To guarantee $d_{min,i}$, Bob boosts the transmission power by scaling up the (I,Q) values of m_M by

$$b_{M,max} = \max_i b_{M,i}. \tag{5}$$

Although this energy boost is not required for all m_i's, Alice always applies $b_{M,max}$ to eliminate any difference in the energy levels of different modulation schemes, which may leak the underlying m_i. For an m_i with $b_{M,i} < b_{M,max}$, however, this boost will reduce the demodulation error. For the optimal mapping to 64-QAM, $d_{min,i}$ of BPSK and QPSK is not preserved, and $b_{4,max} = \sqrt{21}/4 \simeq 1.15$, which is equivalent to only $b^2_{4,max} = 1.181$ dB increase in transmission power (see Table 2).

4.2 Modulation Encryption

Modulation unification introduced in the previous section hides the true modulation scheme of the frame's payload. However, because the mapping $\mathcal{F}_{\mathcal{B}}(.)$ is not necessarily secret (with sufficient randomness) and the modulation scheme for the PHY *header* is often fixed and publicly known, Eve may still be 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 m_M-modulated symbols of the target header field. From the inverse function $\mathcal{F}_{\mathcal{B}}{}^{-1}$, Eve can determine the original symbols s from their m_M-modulated counterparts, revealing the true content of that field. This is especially the case if Eve exhibits a high SNR to demodulate the received m_M-modulated symbols (e.g., Eve is close to Alice). To remedy this situation, we apply a modulation-level stream encryption $\mathcal{E}_{\mathcal{B}}(m_i)$ to the m_i-modulated symbols of the frame (payload + header)[5]

[5]We do not encrypt the preamble, since otherwise Bob cannot detect the start of the frame without knowing in advance

to randomize the location of the original symbols in constellation map of m_i. This way, sole knowledge of $\mathcal{F}_\mathcal{B}$ is not sufficient to disclose the symbol s that corresponds to an observed m_M-modulated symbol. $\mathcal{E}_\mathcal{B}(.)$ is applied before $\mathcal{F}_\mathcal{B}$ and should be uniquely decodable; i.e., it is a 1-to-1 mapping. Note that if we alternatively upgrade the modulation scheme first and then apply encryption, Bob may not reliably decode an m_M-modulated symbol with low SNR.

The encryption function $\mathcal{E}_\mathcal{B}(m_i)$ is performed as follows. Consider $\log_2 |m_i|$ information bits, corresponding to one symbol of the modulation scheme m_i. Select $\log_2 |m_i|$ successive bits from $\mathcal{B}$ and modulate them using m_i. Let x and y be the decimal values of the information and bogus symbols, respectively (in the range $0, 1, \ldots, |m_i|-1$). The value of the encrypted symbol, denoted by z, is given by:

$$z = (x + y) \bmod |m_i|. \tag{6}$$

Bob uses the same bogus symbol y to obtain x:

$$x = \big(|m_i| - (z - y)\big) \bmod |m_i|. \tag{7}$$

This is a symmetric function; x and y are interchangeable. Table 3 depicts an example of encrypting QPSK symbols. As mentioned earlier, Eve may obtain the original m_i-modulated symbol of an observed m_M-modulated one. However, the z value determined by Eve can potentially correspond to any possible x. In other words, the observation of z does not reduce the entropy, which means mutual information is zero at Eve, and Eve cannot use z to recover the original symbol s; i.e., we achieve perfect secrecy for the header and the payload (provided that $\mathcal{B}$ is secret).

However, the encryption operation $\mathcal{E}_\mathcal{B}(m_i)$ destroys the Gray code structure of 802.11 modulation schemes. In Gray coding, adjacent points in the constellation map differ by only one bit and a demodulation error almost always causes a single-bit error. After randomizing the points using $\mathcal{E}_\mathcal{B}$, the average BER due to demodulation errors increases by a factor that corresponds to the average hamming distance between any pair of adjacent constellation points in $\mathcal{E}_\mathcal{B}(m_i)$. This factor is about 1.33, 1.17, 2.13, and 3.04 for (D)QPSK, CCK, 16-QAM, and 64-QAM, respectively. This increase in BER is especially important when there is a considerable residual error in frequency offset estimation, which causes many demodulation errors in long frames. In Section 6 we show and argue that this loss is often small and can be compensated for with a slightly higher transmission power. For a modulation scheme m_i with $b_{M,i} < b_{M,max}$, (e.g., 16-QAM and 64-QAM in Table 2), the power boost discussed in Section 4.1 can mitigate this loss.

At Bob, the modulation-encrypted header and payload are treated is 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 $\mathcal{B}$ are needed to decrypt and recover the header. The modulation scheme for the payload is determined after the PHY header has been decoded and the rate field recovered. Eve, on the other hand, cannot decode the header because it is modulation-encrypted by the secret $\mathcal{B}$. As long as the rate field in the header is unknown, Eve cannot determine m_i of the payload, and hence does not know how many information bits are associated with an observed symbol.

the sender's identity (the frame may originate from several possible sources).

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

5. BOGUS TRAFFIC GENERATION

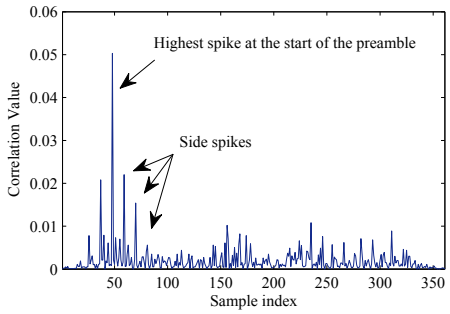
If Alice and Bob were to use the same secret bogus bits for different frames, a given header field that can take a few possible values (e.g., the 8-bit Signal field in 802.11b takes four possible values) would produce a fixed set of constellation points in m_i. After eavesdropping on several frame transmissions that may have different values for that field, Eve may estimate the part of the secret sequence used to protect that field. This can disclose the field values and facilitate a *dictionary attack* against the header content. Moreover, in the case of a retransmission, applying the same $\mathcal{B}$ results in the same sequence of modulated symbols. Eve may then correlate successive transmissions and detect retransmissions. She could then exclude these retransmissions from the statistics used to create the fingerprint of the session (e.g., packet size histogram). Furthermore, if Alice and Bob synchronously use different parts of a common $\mathcal{B}$ for different frames, the loss of an ACK would make Alice and Bob out-of-sync. For this reason, we require $\mathcal{B}$ to vary from one frame to another, in addition to being secret. In this section, we explain how a secret frame-specific $\mathcal{B}$ is generated based on the PTK.

We exploit a one-way cryptographic hash function from the SHA-2 hash family (recommended by NSA) to generate $\mathcal{B}$ based on a seed value. These functions enjoy the property that even a bit change in the seed makes the hash value ($\mathcal{B}$ in our case) completely different. Also, if the hash value is extracted, it cannot be used to recover the seed value, i.e., it is one-way. If the seed is to be completely secret, i.e., the PTK is solely the seed, the hash value $\mathcal{B}$ will always remain the same. So the idea is to concatenate an unprotected frame ID, denoted by $\mathcal{ID}$, to the PTK and compose a partially secret seed for the given frame. *Friendly CryptoJam* embeds $\mathcal{ID}$ in the frame preamble and transmits it in the clear. If there are other nodes (e.g., Charlie) that may also communicate with Bob, we assume that the session ID is part of $\mathcal{ID}$, allowing Bob to distinguish between Alice's and Charlie's transmissions in the absence of MAC addresses and accordingly apply the right PTK.

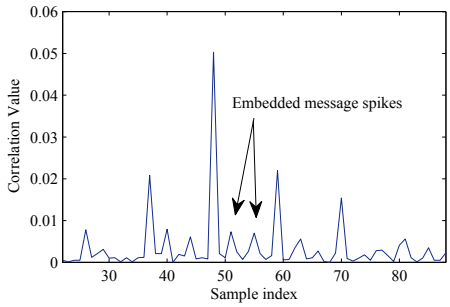
5.1 Embedding the $\mathcal{ID}$

To embed the non-secret $\mathcal{ID}$, we can introduce a new field between the preamble and the standard PHY header. However, to keep the standard PHY-layer frame format intact for interpretability purposes and also to avoid increasing the frame size, we take advantage of the known preamble to embed $\mathcal{ID}$ into it in the form of an analog-signal superposition. (Note that we cannot use any reserved bits in the header(s) because the entire header is supposed to be encrypted by *CryptoJam*.) The design below is specific to the 802.11b long preamble, but the basic idea can be generalized to other preamble structures.

Correct $\mathcal{ID}$ extraction from the superposition is highly 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



(a) The highest spike indicates the start of a frame.



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

Figure 7: $\mathcal{R}(n)$ computed over a frame.

using cyclically rotated Barker sequences (Section 2) to encode Alice's $\mathcal{ID}$. When a Barker sequence is aligned with the original preamble, the function $\mathcal{R}(n)$ (defined in (3)) 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. Because of the orthogonality property of Barker sequences, this overlaid message is easily detectable with RF correlation. Moreover, the frame detection process will not be noticeably affected because the encoded message will have little contribution to the correlation with the reference sequence, when aligned properly. Figure 7(a) is an example drawn from our experiments (Section 6) 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 multiple of 11 indices away from the beginning of the preamble. Figure 7(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 spikes.

For each frame (including retransmissions), Alice picks a frame $\mathcal{ID}$ that has not been recently used. It is conveyed by concatenating several k-shifted versions of the Barker sequence, which are 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 corresponds to a cyclically shifted Barker sequence with $(k_i + 1)$ amount of shift. Concatenation of the l shifted Barker sequences produces $\mathcal{ID}$. Bob is still able to detect the preamble and the $\mathcal{ID}$, as shown in Figure 7. The steps taken by Bob to extract $\mathcal{ID}$ and perform the preamble functions are summarized as follows:

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

5.2 Practical Issues

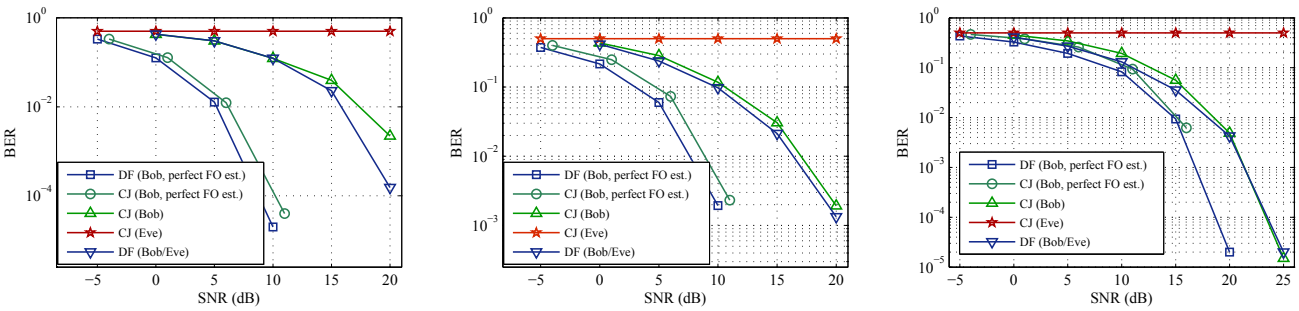
Embedding a frame $\mathcal{ID}$ in the preamble may affect some of the preamble's common functions. We discuss how an appropriately designed message embedding mechanism can maintain these functions.

(1) Frame detection. A typical receiver performs sliding-window correlations using different time offsets (parameter n in (3)). In the case of CJ, the superposed $\mathcal{ID}$ will cause a few spikes when Bob correlates the reference preamble with the received signal at time offsets $k_1, \ldots, k_l$, from the start of the preamble. To avoid creating an alias of the actual start of the preamble, Alice makes sure that she uses different rotation values over successive preamble bits. Let the number of such successive rotations be $l < 11$ ($l \neq 6$). Excluding the noise and multipath channel effect, the message spikes cannot be larger than $\frac{(6-l)^2}{(5l)^2}$ of the highest spike. Because in every sequence of l rotations, at most one of them will perfectly aligned 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) Frequency offset estimation. As explained in Section 2, frequency offset estimation requires two identical repetitions of an arbitrary sequence. We satisfy this requirement by repeating the $\mathcal{ID}$-bearing signal at least twice. Specifically, if Bob uses K repetitions of the Barker sequence (preamble bits) for FO estimation, Alice places the $\mathcal{ID}$-bearing signal in the first $K/2$ sequences and then repeats it over the other $K/2$ sequences. (If $K > 2l$, Alice uses the last l bits in each half to superimpose the $\mathcal{ID}$.) If Alice does not know K a priori, she only exploits the portion of the preamble that will likely be detected by Bob. Bob then can find the start of the $\mathcal{ID}$ signal either by an energy-increase detection, or by iteratively running (on each preamble bit) a series of threshold-based correlations with nonzero rotations of the Barker sequence. Once a correlation value exceeds the threshold, this indicates the start of the $\mathcal{ID}$ signal.

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

(4) Message capacity and error correction. There are 10 distinct rotations of an 11-chip Barker sequence. In DBPSK, this translates into 10 different messages per preamble bit. So in every nine out of 128 bits of the preamble, 10! different $\mathcal{ID}$s of the form $(k_1 k_2 \ldots k_9)_{10}$ can be embedded. Given this large number, Alice can define a coding scheme over the set of $\mathcal{ID}$s to reduce the message detection errors (e.g., using messages with large Hamming distances).



(a) BER vs. SNR (m_i = BPSK). (b) BER vs. SNR (m_i = QPSK). (c) BER vs. SNR (m_i = 16-QAM).

Figure 8: BER performance of modulation encryption and unification (simulations).

6. PERFORMANCE EVALUATION

We evaluate *Friendly CryptoJam* using the LabVIEW simulation environment. We also implement it on an NI-2922 USRP testbed controlled by the LabVIEW USRP driver. Our LabVIEW PHY-layer libraries include the transmitter components in Figure 3, as well as channel and frequency offset estimations modules at the receiver. In the simulations, FO is a controllable parameter, whereas in the experiments, it is a feature of the USRP radio oscillator.

(a) Modulation. We use three basic modulation schemes, BPSK, QPSK, and 16-QAM. The modulation mappings follow Figures 5 and 6. When various modulation schemes are mapped to 16-QAM, $b_{M,max} = b_{3,max}^2 \backsimeq 1$ dB.

(b) Physical frame. Unless specified otherwise, each frame consists of a 44-bit Barker code DBPSK modulated preamble (four 11-chip Barker sequences) followed by a 512-bit random payload. The frame is transmitted over a 2.4 GHz frequency band at a symbol rate of 1 Msamples/s.

(c) Bogus traffic. For bogus traffic generation, we did not implement SHA-2. Instead, we generated a sufficiently large random sequence, shared between Alice and Bob. In [13], it was shown that the data rate of the hardware (FPGA) implementations of SHA-2 hash family exceeds one Gbps, which is quite sufficient for *Friendly CryptoJam*. To support the highest data rate in 802.11n (600 Mbit/s), a bogus traffic generator with at least this data rate is required.

(d) Metrics. We evaluate the BER performance and preamble-related operations, such as frame detection and FO estimation, for different SNR values and modulation schemes. The message extraction success rate is another important metric of interest.

6.1 Simulations

To assess the performance of individual components of CJ, in the simulations we first decouple the unification/encryption schemes from the message embedding approach. AWGN channel model is used with channel coefficient of 1. We then evaluate all the components together in the USRP experiments.

6.1.1 Modulation encryption and unification

Because Alice uses only 16-QAM symbols for transmission, Eve always receives 16-QAM symbols. So Eve always detects 16-QAM as the underlying modulation scheme. If Eve applies this modulation scheme to demodulate symbols that were originally modulated using BPSK or QPSK, her BER will be high. Besides the BER, we evaluate Eve's capability in wiretapping the encrypted header by assuming that she knows *Friendly CryptoJam* and the header's original modulation scheme. BER in digital communications not only depends on SNR and $d_{min,i}$, but also on FO estimation accuracy. For this reason, we obtain the results with and without perfect FO estimation.

Figure 8 depicts the BER performance of CJ as a function of the SNR at Bob/Eve for different modulation schemes m_i. In the figures, the DF scheme refers to the default operation without CJ, which is used as our benchmark. When BPSK is used (Figure 8(a)), modulation encryption does not impact Gray coding, so $d_{min}\big(\mathcal{F}_{\mathcal{B}}(m_i)\big)$ of modulation unification is the only important parameter. When FO estimation is perfect, CJ can achieve almost the same BER but with about 1 dB increase in the transmission power. This verifies our analysis in Section 4. However, when Bob has to estimate FO, not only the BER increases due to FO estimation errors, but also the accumulation of phase errors over time starts to impact 16-QAM-modulated symbols of CJ more than the default scheme (withe BPSK-modulated symbols). To account for the BER increase, Alice can increase her transmission power by about 2 dB. Eve, on the other hand, cannot perform better than a random guess (BER = 0.5).

For the QPSK case in Figure 8(b), 1 dB power increase may not be sufficient even with perfect FO estimation, because the structure of the Gray code is no longer preserved. However, for 16-QAM in Figure 8(c), this loss of structure is partially compensated for by the excess power boost (because $b_{3,max} > b_{3,3}$), and hence the gap between CJ and the default case narrows. Higher modulation orders are less vulnerable to FO estimation errors, because under Gray coding, a large phase offset flips a smaller fraction of bits. This explains why the perfect FO estimation case is closer to the imperfect estimation case when m_i is set to 16-QAM. An interesting observation is that different modulation schemes have similar BER performance under CJ (with imperfect FO estimation). Figure 8(c) also verifies that even without modulation unification, modulation encryption is sufficient to protect unencrypted fields.

6.1.2 Message embedding

Next, we disable modulation encryption and unification but embed an $\mathcal{ID}$ in the preamble. We study how much the superposition of $\mathcal{ID}$ into the preamble affects frame detection (timing), FO estimation, and channel estimation. We also measure the efficiency of the embedded message extraction method. In all examined cases, the modulation scheme is QPSK and the embedded message signal has the same energy as the preamble, unless specified otherwise.

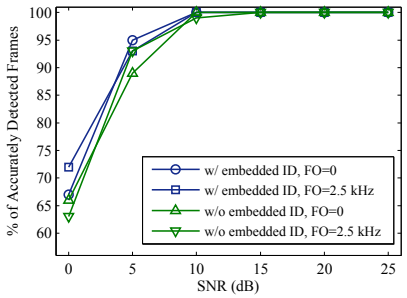


Figure 9: Frame detection accuracy vs. SNR with and without an embedded $\mathcal{ID}$ for different FO values (simulations).

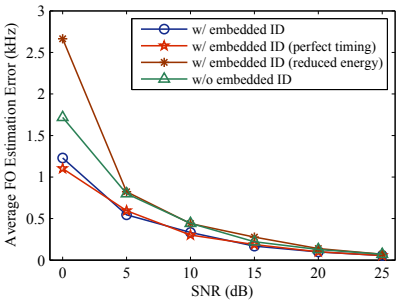


Figure 10: FO estimation error vs. SNR with and without an embedded $\mathcal{ID}$ in the preamble (simulations).

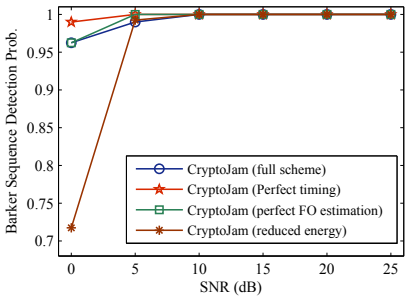


Figure 11: Performance of shifted Barker sequence ($\mathcal{ID}$) detection at Bob vs. SNR (simulations).

Accurate frame detection is the first requirement in the decoding process. It starts by a threshold-based energy detection, followed by the preamble (Barker sequence) correlation. Figure 9 shows that the embedded message does not noticeably impact frame detection even when FO is not zero. In fact, the higher received energy due to the superposition makes energy detection with message embedding slightly more accurate in presence of noise. Although four distinctly shifted Barker sequences generate additional spikes when correlated with an incorrectly aligned reference sequence, the highest of these spikes will not be more than 15% of the spike of the correctly aligned reference sequence (see Figure 7).

The receiver then moves on to the next phase; frequency offset estimation. The symmetry between two parts of the preamble-message combo together with the higher energy improves FO estimation, as illustrated in Figure 10. To specifically study the effect of $\mathcal{ID}$ embedding on FO estimation, we also simulate the scheme assuming perfect frame detection. The results show that FO estimation is not considerably impacted by the frame detection errors. The reason is that even though a frame timing error eliminates some of the samples of the combo from the FO estimation process, the symmetry property still holds for most of the samples included in the estimation process and the effect of the samples that do not belong to the combo averages out. We then add a forth curve to study the effect of the $\mathcal{ID}$ signal's energy. In particular, we multiplied its samples by $1/\sqrt{2}$ before transmission. According to this curve, Alice can reduce the $\mathcal{ID}$ signal's energy to achieve similar FO estimation performance when the noise level is not high.

The performance of CJ highly depends on correct extraction of the $\mathcal{ID}$. Figure 11 provides more details about the impacts of FO estimation and frame detection on the message detection performance by comparing the rate of correct detection of CJ with the cases when either of the aforementioned processes was perfectly done in CJ. When the SNR is high ($\geq$ 5), Bob can always extract the embedded $\mathcal{ID}$. However, the error increases with the increase in the noise power and frame detection errors. Correct $\mathcal{ID}$ extraction depends more on correct frame detection than on FO estimation, as perfect FO estimation does not improve the detection rate, but perfect timing almost always extracts the message correctly. We also consider the reduced energy case of Figure 10. Inline with the effect of low SNR, the results show that a reduction in the energy is not beneficial to Alice. We also note that the preferred 3 dB energy increase during the preamble transmission is not considerable because the duration of the preamble is much less than the duration of the payload.

Last but not least, we evaluate the BER when a message is embedded in the preamble (Figure 12). To measure the sensitivity to FO estimation errors, we also simulate the cases with perfect FO estimation. It turns out that the FO estimation has a crucial impact on the BER. This also justifies why CJ achieves a better BER when the superposition energy is higher; so FO estimation is more accurate. Likewise, when the $\mathcal{ID}$ signal's energy is reduced, the BER increases.

6.2 USRP Experiments

In our testbed, one of the USRPs always acts as Alice while the other one can be either Bob or Eve. Since the scheme has been extensively studied under an AWGN channel model in the simulations, we exploit our USRPs to emulate a real transmission in an indoor multipath environment. In particular, we eliminate the LOS component by placing an obstacle between Alice and Bob/Eve. The experimental scenarios, listed in Table 4, are based on the Alice-Bob/Eve distance and type of the obstacle. It is worth to mention that we also experimented with simpler scenarios without an obstacle in which the distance was the varying parameter. However, the BER under this setup is either zero (in most of the cases) or similar to the BER in the setup with an obstacle, and so we do not report them here.

Scenario #	Alice-Bob Distance	Obstacle
1	70 cm	Cardboard box
2	1.2 m	Cardboard box
3	1.2 m	PC case (metal)

Table 4: Scenarios used in the USRP experiments.

In analyzing the measured data, we filter out cases in which transmissions were not detected by the USRP. Those cases constituted less than 0.3% of the transmitted frames. We also distinguish between cases based on whether or not the $\mathcal{ID}$ is correctly detected. Basically, any message detection error will result in a packet drop and we exclude these samples in the averaging. Nevertheless, the successful detection rate is always higher than 99.83% in our experiments.

Figure 13 depicts the BER for different payload's modulation schemes but with the same packet size. Surprisingly, QPSK shows the best performance. FO error can explain the reason. For the same frame length, BPSK results in a higher frame duration than QPSK. So due to FO

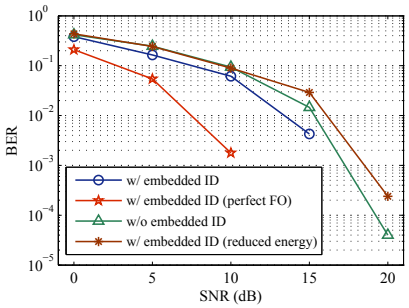


Figure 12: BER vs. SNR with and without an embedded $\mathcal{ID}$ (simulations).

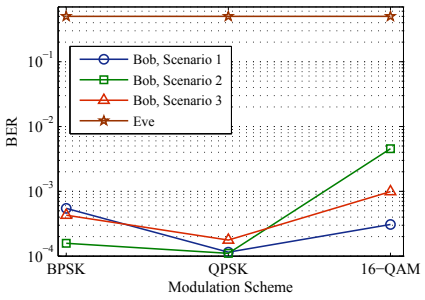


Figure 13: BER vs. modulation schemes with fixed packet size (USRP experiments).

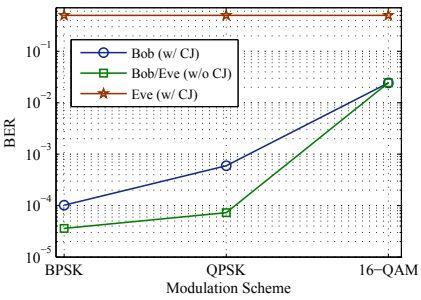


Figure 14: BER vs. modulation schemes with fixed frame duration (USRP experiments).

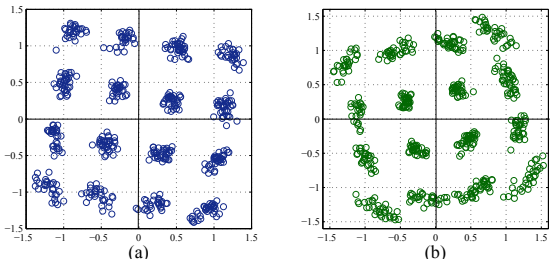


Figure 15: Examples of received 16-QAM symbols at Eve for (a) m_i = BPSK, (b) m_i = QPSK (USRP experiments).

estimation errors, phase error accumulation over a longer time period causes more demodulation errors when BPSK symbols are extracted from the 16-QAM constellation map. The frame durations under BPSK, QPSK, and 16-QAM in our experiments are approximately 6.78, 3.68, and 2.05 ms, respectively. On the other hand, QPSK by design has a higher $d_{min,i}$ than 16-QAM, which in this case dominates the small frame duration difference compared to the 16-QAM-modulated frame.

Next, we consider a situation in which Alice wants to confuse Eve about the actual payload size by transmitting packets of three different sizes but with the same frame duration. In particular, Alice always transmits for 3.64 ms (preamble + payload). Depending on the CSI, she may actually send 250 bits with BPSK, 500 bits with QPSK, or 1000 bits with 16-QAM. From Eve's perspective, however, these packets look the same in duration and modulation scheme. Figure 15 illustrates two example constellation maps of what Eve observes when the payload's original modulation scheme is BPSK and QPSK (assuming scenario 3). Eve will always detect 16-QAM symbols and decode them into 1000-bit packets. The corresponding BER performance was shown in Figure 14. Even when m_i is 16-QAM, Eve's BER is high due to modulation encryption. As for Bob, he can identify the underlying m_i and correctly decode the symbols. While Eve cannot decode a frame or estimate its size, modulation scheme, rate, etc., Bob can achieve a BER close to the default case. Disorganization of Gray code together with FO estimation errors can explain the small performance loss.

7. RELATED WORK

Several upper-layer techniques, such as padding, traffic morphing [17], and packet features masking at the application layer [12], have been proposed to prevent the leakage of side-channel information by changing the 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 [7]. Dyer *et al.* [7] demonstrated that even if packet lengths are obfuscated, training a 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 [18] is used to split 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, traffic splitting requires modifying the MAC protocol. Furthermore, none of the above techniques can hide lower-layer fields such as the modulation scheme and the data rate. *Friendly CryptoJam*, 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 from Eve's perspective.

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 still vulnerable to wiretapping, in addition to side lobes. Also, in some circumstances, these techniques may fail to provide directionality (e.g., see [3, 5]). Other techniques such as beamforming and orthogonal blinding (e.g., [3]) are essentially based on prefiltering (precoding) a signal, which have been shown to be insufficient [15].

8. CONCLUSIONS

Preventing leakage of wireless transmission attributes, including unencrypted header fields and modulation scheme, is challenging. In this paper, we proposed *Friendly CryptoJam*, a combination of friendly jamming and low-level encryption, to effectively protect lower-layer fields using a single antenna, and prevent traffic classification and rate-adaptation attacks. The scheme employs three main techniques: First, a modulation-level encryption technique that can perfectly secure the headers and the payload. Second, an optimal

and energy-efficient modulation unification technique that obfuscates the modulation scheme and partially decorrelates the modulated-frame duration from the packet size and the total traffic volume. Third, a message embedding technique that overlays an $\mathcal{ID}$ on the preamble for exchanging packet/sender identifiers instead of the (encrypted) MAC address, which is used for session-key look up in existing systems. We showed theoretically and experimentally that constructing an $\mathcal{ID}$ using a series of shifted Barker sequences and then superposing it on the 802.11b preamble is reliable, without affecting the preamble functions, such as frequency offset estimation. The simulation and experimental results also verify that with a slight increase in the transmission power, modulation unification and encryption are successful in hiding the true packet size, modulation scheme, and the content of a frame without degrading the BER.

9. ACKNOWLEDGEMENTS

This research was supported in part by the Army Research Office (grant # W911NF-13-1-0302) and in part by the National Science Foundation (grants # IIP-1265960 and CNS-1016943). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of ARO or NSF.

10. REFERENCES

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

[2] *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Networks–Specific Requirements: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (Amendment 6: Medium Access Control (MAC) Security Enhancements)*, IEEE Std 802.11i-2004, 2004.

[3] N. Anand, S.-J. Lee, and E. Knightly. STROBE: Actively securing wireless communications using zero-forcing beamforming. In *Proc. IEEE INFOCOM'12*, pages 720–728, March 2012.

[4] J. Atkinson, O. Adetoye, M. Rio, J. Mitchell, and G. Matich. Your WiFi is leaking: Inferring user behaviour, encryption irrelevant. In *Proc. IEEE Wireless Communications and Networking Conf. (WCNC'13)*, pages 1097–1102, April 2013.

[5] M. Buettner, E. Anderson, G. Yee, D. Saha, A. Sheth, D. Sicker, and D. Grunwald. A phased array antenna testbed for evaluating directionality in wireless networks. In *Proc. 1st ACM Int. Workshop System Evaluation for Mobile Platforms (MobiEval'07)*, pages 7–12, San Juan, Puerto Rico, 2007.

[6] 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)*, pages 191–206, May 2010.

[7] 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)*, pages 332–346, May 2012.

[8] S. Goel and R. Negi. Guaranteeing secrecy using artificial noise. *IEEE Trans. Wireless Communications*, 7(6):2180–2189, June 2008.

[9] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: Non-invasive security for implantable medical devices. In *Proc. ACM SIGCOMM Conf. Data Communication (SIGCOMM'11)*, pages 2–13, Toronto, Ontario, Canada, August 2011.

[10] S. Gollakota and D. Katabi. ZigZag decoding: Combating hidden terminals in wireless networks. In *Proc. ACM SIGCOMM Conf. Data Communication (SIGCOMM'08)*, pages 159–170, Seattle, WA, USA, October 2008.

[11] X. He, H. Dai, W. Shen, and P. Ning. Is link signature dependable for wireless security? In *Proc. IEEE INFOCOM'13*, pages 200–204, April 2013.

[12] A. Iacovazzi and A. Baiocchi. From ideality to practicability in statistical packet features masking. In *Proc. 8th Wireless Commun. Mobile Computing Conf. (IWCMC'12)*, pages 456–462, August 2012.

[13] R. P. McEvoy, F. Crowe, C. Murphy, and W. P. Marnane. Optimisation of the SHA-2 family of hash functions on FPGAs. In *Proc. IEEE symp. Emerging VLSI Technologies and Architectures*, March 2006.

[14] G. Noubir, R. Rajaraman, B. Sheng, and B. Thapa. On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming. In *Proc. Fourth ACM Conf. Wireless Network Security (WiSec'11)*, pages 97–108, Hamburg, Germany, June 2011.

[15] M. Schulz, A. Loch, and M. Hollick. Practical known-plaintext attacks against physical layer security in wireless MIMO systems. In *Proc. Network and Distributed System Security Symp. (NDSS'14)*, February 2014.

[16] 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)*, pages 160–173, May 2013.

[17] C. V. Wright, S. E. Coull, , and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *Proc. Network and Distributed System Security symp. (NDSS'09)*, pages 237–250, February 2009.

[18] F. Zhang, W. He, and X. Liu. Defending against traffic analysis in wireless networks through traffic reshaping. In *Proc. 31st IEEE Int. Conf. Distributed Computing Systems (ICDCS'11)*, pages 593–602, June 2011.

[19] F. Zhang, W. He, X. Liu, and P. G. Bridges. Inferring users' online activities through traffic analysis. In *Proc. Fourth ACM Conf. Wireless Network Security (WiSec'11)*, pages 59–70, Hamburg, Germany, 2011.

[20] Y. Zhang. Method, apparatus and system for carrier frequency offset estimation, Oct. 17 2013. US Patent App. 13/597,204.