MIO: Enhancing Wireless Communications Security Through Physical Layer Multiple Inter-Symbol Obfuscation

Tao Xiong, Wei Lou, *Member, IEEE*, Jin Zhang, *Student Member, IEEE*, and Hailun Tan, *Member, IEEE*

***Abstract*— Communications security is a critical and increasingly challenging issue in wireless networks. A well-known approach for achieving information-theoretic secrecy relies on deploying artificial noises to blind the intruders' interception in the physical layer. However, this approach requires a static channel condition for the transmitter and receiver to generate and offset the controllable artificial noise, which can hardly be implemented in real wireless environments. In this paper, we explore the feasibility of symbol obfuscation to defend against the passive eavesdropping attack and fake packet injection attack during the wireless communications. We propose a multiple inter-symbol obfuscation (MIO) scheme, which utilizes a set of artificial noisy symbols (symbols key) to obfuscate the original data symbols in the physical layer. MIO can effectively enhance the wireless communications security. On the one hand, an eavesdropper, without knowing the artificial noisy symbols, cannot correctly decrypt the obfuscated symbols from the eavesdropped packets. On the other hand, a legitimate receiver can easily check the integrity of the symbols key and then reject the fake packets from the received packets. The security analysis reveals that, without considering the initial key, the MIO scheme can achieve information-theoretic secrecy against the passive eavesdropping attack and computational secrecy against the fake packet injection attack. Moreover, we have implemented our approach in a USRP2 testbed and conducted simulations with Simulink tools to validate the effectiveness of MIO in enhancing wireless communications security.**

***Index Terms*— Wireless communications security, physical layer security, information-theoretic secrecy, artificial noise.**

I. Introduction

WIRELESS networks are becoming an indispensable part of people's daily life. As a result, security is an imperative issue in wireless networks since the users might transmit their sensitive personal information (e.g., credit card details) over the wireless networks. In addition, wireless channels are susceptible to eavesdropping [1] and malicious message injecting [2] due to the openness and sharing of the wireless medium.

Recent research has shown that physical layer security techniques become a more essential part in the wireless communications [2]–[9]. Compared with the traditional asymmetric/symmetric cryptographic techniques which provide the computational secrecy, it has been proved that, physical layer security techniques, such as using a proper channel coding, can achieve the information-theoretic secrecy which makes the eavesdropper hardly break the encryption even it has unlimited computing power. However, the information-theoretic secrecy requires a strict positive secrecy capacity that the legitimate transmitter and receiver have to be in a better quality channel than the attacker [10]–[12]. Later works have shown that by artificially interfering the transmitting signal, the positive secrecy capacity requirement can be achieved [2], [5]–[8] in practical wireless communications. But, most of these techniques need to deploy trusted third parties [2], [6]–[8] or multiple antennas (MIMO) [13] to generate the artificial noise. Moreover, the positive secrecy capacity of these works may be compromised if the eavesdropper deploys at certain locations.

In this paper, we adopt a multiple inter-symbol obfuscation (MIO) scheme to enhance wireless communications security at the physical layer. In MIO, upon sending each data packet, a random subset of the corresponding data symbols are obfuscated with a set of artificial noisy symbols, which is called *symbols key*, so that (1) the eavesdropper's channel quality is worse than the legitimate receiver's and (2) the eavesdropper cannot decrypt the data symbols correctly since it does not know the symbols key, which is updated dynamically during the data packets' transmissions. For the legitimate receiver, it can offset the obfuscation of the symbols key by employing the reversed symbols key to derive the intended data symbols from the legitimate transmitter. In addition, the legitimate receiver can discern the fake packets sent from the adversary as it will fail to pass the integrity check of the symbols key on the fake packets through symbol cross-correlation. Fig. 1 provides an overview about how MIO defends against both the passive eavesdropping attack and fake packet injection attack.

The contributions of this paper are summarized as follows:

- We propose the MIO scheme that combines the data symbols encrypting and channel interfering at one step.

Manuscript received November 29, 2014; revised February 23, 2015; accepted April 5, 2015. Date of publication April 13, 2015; date of current version June 25, 2015. This work was supported in part by The Hong Kong Polytechnic University, Hong Kong, under Grant A-PL84 and Grant 4-BCB6, in part by the Research Grants Council, University Grants Committee, Hong Kong, through the General Research Fund under Grant PolyU-521312, and in part by the National Natural Science Foundation of China under Grant 61272463. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kah C. Teh. *(Corresponding author: Wei Lou.)*

T. Xiong, W. Lou, and J. Zhang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and also with the Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, P. R. China (email: cstxiong@comp.polyu.edu.hk; csweilou@comp.polyu.edu.hk; csjzhang@comp.polyu.edu.hk).

H. Tan is with Australian Federal Government, Brisbane, QLD 4000, Australia (e-mail: hailun.tan@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2422264

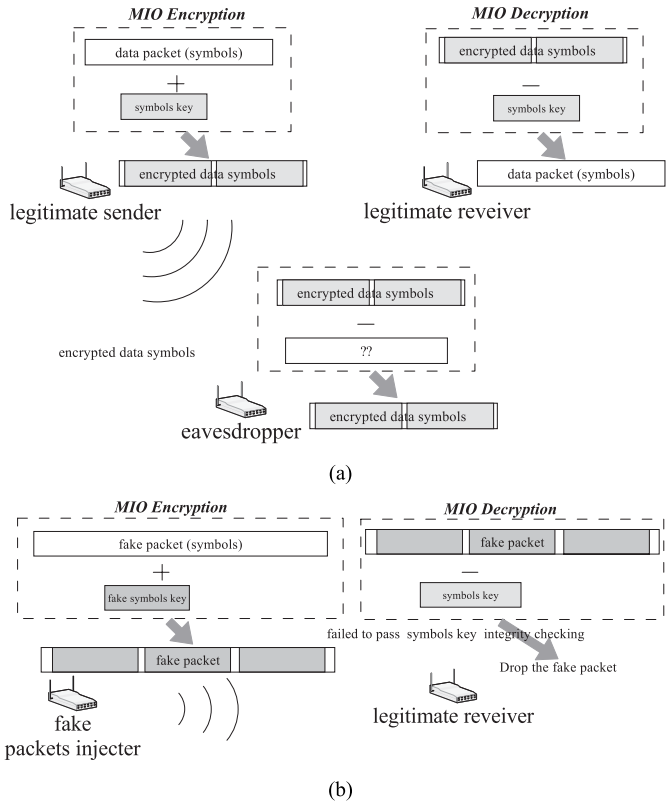


Fig. 1. The overview of Multiple Inter-symbol Obfuscation (MIO). (a) MIO against the passive eavesdropping attack. (b) MIO against the fake packet injection attack.

In MIO, the symbols key not only encrypts the baseband data symbols but also interferes these symbols, which guarantee that the secrecy capacity of the wireless communication would always stay positive regardless of the location of the eavesdropper. Thus, the information-theoretic secrecy can be achieved. On the contrary, the traditional bit-level symmetric/assymmetric cryptographic schemes cannot guarantee this feature. Also, compared with other physical layer security schemes, MIO does not have to concern or assume the channel state information (CSI), as the noisy symbols key has interfered the eavesdropping channel regardless of the location, which makes that the legitimate channel is better than the eavesdropping channel. Moreover, MIO does not need any trusted third party to deploy the noisy symbols key, and the secrecy capacity will not be decreased by the location of the eavesdropper.

- We design a dynamic key extraction mechanism to change the artificial noisy symbols key to defend against the eavesdroppers from retrieving the correct information of symbols key in MIO. In this mechanism, as the legitimate transmitter can randomly encrypt the data symbols without notifying the legitimate receiver, the receiver has to employ the key checking process to locate the symbols key's position and the corresponding dynamic encrypted symbols. Consequently, MIO can allow the legitimate receiver to decrypt those encrypted symbols without any further information.
- We prove that, without considering initial key, the MIO scheme can provide information-theoretic secrecy against the passive eavesdropping attack and computational secrecy against the fake packet injection attack. In addition, this information-theoretic secrecy would not be compromised by the location of the eavesdroppers. Moreover, we show that MIO can defend against the symbol detection attempts as well as the acknowledgement-based key disruption attack. Thus, MIO can greatly enhance the wireless communications security.
- We evaluate MIO's performance with the USRP2 [14] testbed and Simulink tools to further validate the effectiveness of MIO in real wireless environments.

The rest of the paper is organized as follows: The Section II reviews the related work in the physical layer security. We describe the threat model in Section III. In Section IV we discuss the system design of MIO and in Section V we provide the security analysis about the MIO scheme. Section VI addresses some hardware implementation issues related to MIO. The performance evaluation through hardware and simulation experiments are delivered in Section VII. We further discuss several issues about the MIO scheme in Section VIII and conclude the paper in Section IX.

II. Related Work

Although the communications security has been a popular research topic in the research community of wireless networks, the development of wireless communications security, particularly in the physical layer, remains at its early stage. Prior physical layer security research mainly falls in the following three areas, *channel coding approaches*, *signal design approaches*, and *artificial noise approaches* [3]:

1) Channel Coding Approaches: Channel coding approaches can defeat packet interception and jamming problems [3]. Code division multiple access (CDMA) [15]–[18] is a well-known channel coding scheme in the wireless communications security area. By using the bit-level pseudo noise code (PN code), the encrypted transmission message can only be decrypted by the legitimate user. However, traditional CDMA has limited PN codes, and users have to share those PN codes. To solve this PN code size problem, Li et al. [16] enhanced the CDMA security based on the advanced encryption standard (AES) operation. It specifies 3 different AES-CDMA PN code sizes (128, 192, and 256 bits) to raise the security level against eavesdropping. Unfortunately, like other channel coding approaches, this long size security code lags the wireless transmission rate, thus reduces the network goodput. Moreover, CDMA is a spread spectrum multiple access technique in which the transmission data are combined via bitwise XOR with the PN code. Thus, CDMA is still a kind of bit-level symmetric cryptographic techniques which cannot provide the information-theoretic secrecy in the wireless communications.

Liu et al. [19] showed that the low-density parity-check (LDPC) code can achieve the secrecy capacity of the wiretap channel, and proved this code can be used

to provide perfectly secret communications at low data rates. However, it is under the assumptions that the main channel must be less noisy than the eavesdropping channel and the eavesdropping channel is a general binary-input symmetric-output memoryless channel, which can hardly be true in the real wireless communications environment.

2) Signal Design Approaches: The advantage of the signal design approaches is that, by designing a different signal constellation mapping method, the eavesdropper cannot correctly map the received digital symbols into bits, which leads to the incorrect decoding of the packets. Pöpper et al. [4] proposed the symbols flipping method by rotating a preset angle for the baseband data symbol vectors before transmissions. The legitimate receiver can retrieve the data symbol vectors by reversing the angle rotation. However, the rotating angle in their scheme is fixed and the eavesdropper can brute-force the rotating angle after intercepting sufficient data packets for demodulation.

Different from Pöpper's work, Husain et al. [9] proposed a constellation diversity mapping method to secure the wireless transmission. It increases the bit error rate (BER) at the eavesdropper side by using different constellation maps (e.g., change circular constellation to rectangular constellation) in wireless transmissions (under Gaussian noise), this constellation diversity mapping can hardly be detected by normal symbol detection attempts [20], [21]. However, this scheme is demonstrated to be more suitable for the complex modulation, like M-QAM. Moreover, the information-theoretic secrecy can be compromised under certain specific symbol detection attempts.

3) Artificial Noise (AN) Approaches: Recent studies [2], [5]–[8] exploit the advantage of deploying artificial noise that can easily make the intruders' channel more noisy than the legitimate users' channel to achieve the information-theoretic secrecy. Sperandio and Flikkema [5] proposed to obfuscate the original signal by imposing the multiple orthogonal artificial noise through the multi-path transmissions. The receiver can retrieve the correct signal by having multiple orthogonal noise to offset each other while the eavesdropper is not able to retrieve the correct signal without correct location. However, their scheme is constrained to a static channel condition requirement for both the sender and receiver so that the receiver is able to receive the affected signals, together with that artificial orthogonal noise can offset each other through the multi-path effect. Such requirement on the static channel condition of sender and receiver might not be suitable for mobile networks.

Jorgensen et al. [22] proposed a wire-connected third party to send the synchronized artificial noise when intended receiver receives the packet. It uses the secrecy capacity to prove that the AN approach can achieve the information-theoretic secrecy. However, if the eavesdropper is more closer to the transmitter, the secrecy capacity of the scheme can decrease to 0, in which the information-theoretic secrecy is weakened by the locations of eavesdroppers.

Lai and Gamal [7] suggested deploying a trusted third party to send anti-artificial noise during the wireless transmission, thus, the useful information is hard to be intercepted. However, their scheme requires an additional device and static channel condition for the legitimate sender, receiver and trusted third party so that the anti-artificial noise can be synchronously offset with the artificial noise to retrieve the transmitted signals.

Gollakota and Katabi [8] adopted a redundancy mechanism to defend against the wireless signals' interception. Each signal will be required to be sent twice as it may be randomly interfered with any additive noise. The receiver can identify the interfered signal and reconstruct the clean signal. Given the redundancy mechanism, the throughput is reduced. It also requires the signal synchronization between the sender and receiver. They further improved their work by employing the full-duplex hardware to impose the noise to the transmission between the implantable medical devices (IMDs) and the sink. As a result, the mission-critical commands to IMDs cannot be forged or overheard by the unauthorized third party [2]. Similar to Lai and Gamal's design [7], their scheme requires an additional hardware to jam the channel. Moreover, an adversary is able to overhear the transmitted signal if it is sufficiently close to the legitimate transmitter or to inject the unauthorized commands to the legitimate receiver if it is close enough to the legitimate receiver.

The MIO scheme is designed to leverage the advantages of both signal design and artificial noise approaches, and is adaptable to any wireless standard. Different from the prior work, by using the artificial noisy symbols obfuscation, it does not need any trusted third party, signal synchronization or static channel condition for the legitimate sender/receiver or adversary. Moreover, MIO can defend against the symbol detection attempts.

III. Threat Model

The wireless communications security is to prevent attackers from intercepting the wireless communications, while still delivering contents to the intended recipients. In this paper, we address two types of adversaries, *passive eavesdropping attack* and *fake packet injection attack*, during the wireless communications, just like some former works [2], [4], [7]–[9], [19]:

1) Passive Eavesdropping Attack: An adversary eavesdrops on the wireless medium and intercepts the wireless transmission between the legitimate transmitter and receiver. It can attempt to decode the signal from the intercepted signal with the presence of the MIO scheme. The MIO scheme will provide the information-theoretic secrecy to enhance the wireless communications security.

2) Fake Packet Injection Attack: An adversary injects fake packets to the legitimate users, triggering the events to further disrupt the users's manner (e.g., mislead the users' operations). Unlike the passive eavesdropping attack, it can deploy the brute-force to test all possible symbols keys to inject a fake packet. The MIO scheme will enhance the computational secrecy to defend against this attack.

However, we do not consider the cases where the legitimate transmitter or receiver is physically compromised because the data confidentiality is no longer ensured no matter what security measure is adopted to secure the wireless

TABLE I
NOTATIONS

notation	meaning
γ	the size of the symbols key
Key_k	the symbols key to be superimposed for k^{th} data packet
$Key_{k,j}$	the j^{th} key symbol of Key_k to be encrypted with the data symbol ($0 \le j \le \gamma - 1$)
$E_{Key_{k,j}}(S)$	the encrypted data symbol using key symbol $Key_{k,j}$ on data symbol S
$V_{k,j}$	the angle between the key symbol $Key_{k,j}$ and the Real-axis
α	the magnitude ratio of the key symbol and unit-power symbol ($\alpha = \|key_{k,j}\|/1$)
$\hat{\theta}$	the expected normalization factor of the encrypted symbols ($0 < \hat{\theta} < 1$)
β_c	the cross-correlation threshold that the encrypted symbols can be detected
β_{SNR}	the SNR threshold that the received packet can be correctly decoded
R_{re}	the maximum retransmission times for each packet

communications between two hosts if any one of them is not secured. Additionally, we do not consider the jamming-based denial of service (DoS) attack in this paper, where the adversary simply jams the channel with extraordinary transmission power, since the legitimate sender and receiver fail to communicate with each other under this DoS attack.

IV. System Design

This section provides the design of the multiple inter-symbol obfuscation (MIO) which includes two stages: MIO encryption (adding the artificial noisy symbols key), and MIO decryption (offsetting the artificial noisy symbols key). Although the MIO scheme is designed based on the multiple inter-symbol obfuscation at the physical layer, it still needs an initial key to start the secure wireless communications. For ease of presentation, Table I lists the notations used in the following sections.

A. Initialization

To initiate the first symbols key in a non-secure wireless channel, we first take the conventional key agreement protocols, e.g., EKE or augmented EKE [23], [24], to achieve a bit-level authenticated key. Then, the bit-level authenticated key can be used to generate parameters by a one-way harsh function. After that, the parameters, which include the size of the symbols key γ, the angle between the key symbol and the Real-axis $V_{k,j}$ and the magnitude ratio of the key symbol and unit-power symbol α, are used to generate the first symbols key without any trusted third party. Obviously, the legitimate transmitter and receiver have to exchange some redundant packets and deploy the same set of hash functions to generate these parameters.

As the bit-level key agreement schemes can only provide computational secrecy but not information-theoretic secrecy, the key can be compromised if the eavesdropper has enough computational power (detailed in Section V-A). Moreover, the initial key still requires the legitimate transmitter and receiver to exchange redundant packets to generate different keys for different data packets, which introduces a high overhead. During the later data packet transmissions, the legitimate parties would employ the MIO scheme to generate the subsequent dynamic noisy symbols keys and deploy the multiple inter-symbol obfuscation scheme to interfere the eavesdropping channel, which can provide information-theoretic secrecy to wireless communications.

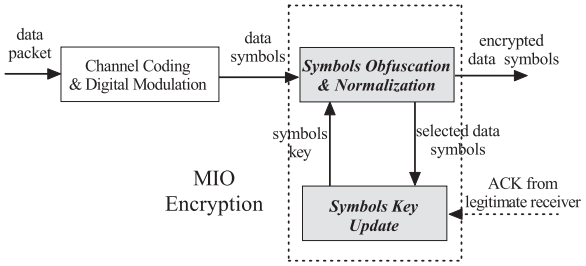


Fig. 2. The MIO encryption process at the legitimate transmitter.

B. MIO Encryption

We first consider that legitimate transmitter A is about to send N data packets to legitimate receiver B. As shown in Fig. 2, for each data packet, it goes through the MIO encryption process by two steps: (1) symbols obfuscation and normalization and (2) symbols key update at the transmitter.

1) Symbols Obfuscation and Normalization: When a data packet P_k ($1 \le k \le N$) is transmitted, transmitter A will map P_k to a series of L baseband data symbols $M_k = \{m_{k,0}, \ldots, m_{k,l}, \ldots, m_{k,L-1}\}$ using the modulation constellation diagram. Each data symbol $m_{k,l}$ ($0 \le l \le L-1$) is represented as

$$m_{k,l} = |m_{k,l}| \, e^{j\phi_{k,l}}, \tag{1}$$

where $|m_{k,l}|$, $\phi_{k,l}$ are the magnitude and angle of the l^{th} symbol vector, respectively. These data symbols are generated by the channel coding & digital modulation block in Fig. 2.

After mapping, the transmitter randomly picks up ξ blocks of data symbols, where $\xi = \left\lfloor \frac{L}{\gamma} \right\rfloor$, from M_k for encryption.[1] For each chosen data symbols block that begins with the i^{th} data symbol, the corresponding $(i+j)^{th}$ data symbol vector $m_{k,i+j}$ is added with the j^{th} key symbol vector $Key_{k,j}$ to generate an encrypted data symbol $E_{Key_{k,j}}(m_{k,i+j}) = Key_{k,j} + m_{k,i+j}$, which is illustrated in Fig. 3.

As shown in Fig. 4, the average power of the encrypted symbols (dot-line curve) would not be the same as that of the original data symbols (solid-line curve) at the transmitter. This energy difference may let the eavesdropper distinguish the encrypted symbols from the non-encrypted ones according to the surge of the transmission power. To avoid this problem, the encrypted symbols should be normalized before they go to the digital to analog converter (DAC). After normalization,

[1] In case $L < \gamma$, the MIO randomly appends some dummy data symbols to make $L = \gamma$.

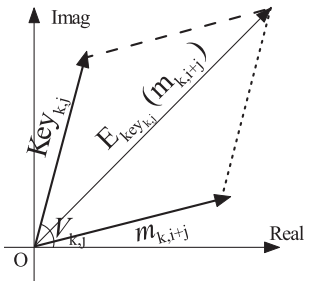


Fig. 3. The obfuscation of a baseband data symbol $m_{k,i+j}$ with a key symbol $Key_{k,j}$ on the constellation diagram.

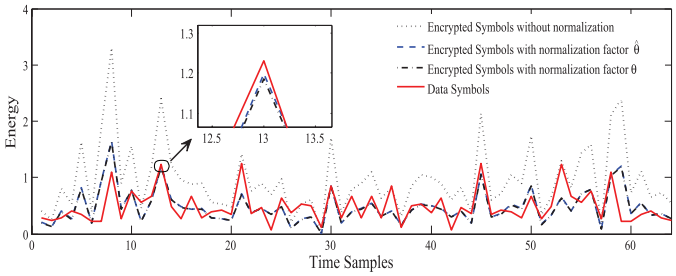


Fig. 4. Simulation result of the signal samples energy in the time domain at the transmitter.

the energy of the encrypted symbols (dot-dash-line curve) is almost the same as that of the data symbols. Consequently, the eavesdropper is hard to determine whether the received symbols are non-encrypted data symbols or encrypted symbols. The normalized factor θ can be calculated as:

$$\theta = \frac{\frac{1}{\gamma\xi}\sum_{j=0}^{\gamma\xi-1}\left|m_{k,i+j}\right|}{\frac{1}{\gamma\xi}\sum_{j=0}^{\gamma\xi-1}\left|E_{Key_{k,j}}(m_{k,i+j})\right|}. \tag{2}$$

Eq. (2) implies that the calculation of θ relies on the information of all original and encrypted data symbols, which is hard for the receiver to obtain before the MIO decryption. Fortunately, as the data symbols and key symbols are generally uniformly distributed, when γ is large enough, MIO can use the expected normalized factor $\hat{\theta}$ to replace θ.

The calculation of $\hat{\theta}$ considers all combination possibilities of data symbols and key symbols. Assume that the symbol set used for mapping data on the constellation diagram is $\{S_u\}$ and the probability that S_u is chosen is a_u; the set of key symbols is $\{Key_v\}$ and the probability that Key_v is used for encrypting data symbol is b_v. Then, the encrypted symbol $E_{Key_v}(S_u) = S_u + Key_v$ and the probability that $E_{Key_v}(S_u)$ is generated is $a_u \cdot b_v$. The $\hat{\theta}$ can be computed as:

$$\hat{\theta} = \frac{\sum_u a_u \left|S_u\right|}{\sum_{v,u} a_u \cdot b_v \cdot \left|E_{Key_v}(S_u)\right|}, \tag{3}$$

where $|S_u|$, $\left|E_{Key_v}(S_u)\right|$ are the magnitudes of S_u and $E_{Key_v}(S_u)$ on the constellation diagram. Obviously, both the legitimate transmitter and receiver can calculate the value of $\hat{\theta}$ without knowing information of original and encrypted data symbols in advance.

In Fig. 4, the QPSK modulation is used for data mapping. Both data symbol and key symbol are unit-power vectors. The angles of the data symbols are $\left\{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\right\}$, and the probability of each data symbol is $\frac{1}{4}$. For the key symbols, the angles are set $\left\{\frac{\pi}{2}, -\frac{\pi}{2}\right\}$ and the probability of each key symbol is $\frac{1}{2}$. As the figure shows, with $\gamma = 60$, the energy of the two normalized encrypted data symbols (dot-dash-line curve and dash-line curve) are almost the same.

Please note that this normalization may decrease the transmission power of the data symbols and we detail this power loss, called as dB loss, in Section VI-B.

2) Symbols Key Update at the Transmitter: After symbols encryption and normalization, the symbols key to encrypt next data symbols is dynamically updated by using the privacy amplification with one-way hash function [25]. The symbols key Key_{k+1} for the next data packet is generated from the data symbols which are encrypted in the current data packet. Because $\gamma\xi$ data symbols are randomly and independently selected, and encrypted with the noisy symbols key Key_k, when they are transmitted, the noise symbols interfere the eavesdropping channel, which makes the eavesdropping channel's quality much worse than the legitimate channel, so the adversary has a small chance to decrypt the $\gamma\xi$ data symbols without knowing the noisy symbols key Key_k. Thus, the $\gamma\xi$ data symbols are completely confidential to the adversary.

After the MIO encryption, as the selected $\gamma\xi$ data symbols are stored in array t, this array is completely confidential to the adversary. By using the array t as input to the privacy amplification with one-way hash function, the distilled symbols key Key_{k+1} is also confidential to the adversary.[2] Here, the one-way hash function can guarantee that the symbols keys are not correlated even if the data symbols in consecutive packets are correlated [26].

A problem with this key update scheme is that, the first noisy symbols key is not protected by the noise symbols, just like other physical layer security schemes [12]. Fortunately, under certain situations, even if the first symbols key is cracked, it cannot help the adversary decrypting other encrypted data packets from the first symbols key (detailed in Section VIII-A) because the succedent noisy symbols keys are dynamically updated. However, this dynamic symbols key update mechanism requires all the symbols to be decrypted successfully for the next data packet at the legitimate receiver side to synchronize the noisy symbols key, consequently, the transmitter has to wait for the *correct* acknowledgment (ACK) from the receiver before it can process the next packet. In a hostile scenario, an adversary might inject forged ACKs to disrupt the symbols key update process between the legitimate transmitter and receiver, which will be further discussed in Section V-E.

[2]The $\gamma\xi$ data symbols in the array t can be mapped into bits and these bits are also confidential to the adversary. After we get the new distilled bits key, MIO would use the one-way hash function to map the bits key into the symbols key.

Algorithm 1 MIO Encryption Process

Input: Key_1 is generated at initialization stage. N data packets are to be transmitted.

Output: Encrypted symbols of P_k.

1: **for** $k = 1$ to N **do**
2: Map the k^{th} packet P_k to L data symbols $m_{k,0}, ...m_{k,i}, ...m_{k,L-1}$;
3: Randomly select ξ blocks of data symbols out of L data symbols;
4: Store all $\gamma\,\xi$ selected data symbols in the array t; /*for next symbols key generation*/
5: **for** each selected data symbols block begins with the i^{th} data symbol **do**
6: **for** $j = 0$ to $\gamma - 1$ **do**
7: $E_{Key_{k,j}}(m_{k,i+j}) = Key_{k,j} + m_{k,i+j}$;
8: $m_{k,i+j} \leftarrow \hat{\theta} \cdot E_{Key_{k,j}}(m_{k,i+j})$; /*encrypted symbol normalization*/
9: **end for**
10: **end for**
11: Set retransmission counter $c_k = 0$;
12: Send the encrypted data symbols M_k to the receiver;
13: **while** receive no ACK packet P_{ack_k} from the receiver before timeout $\wedge\ c_k \leq R_{re}$ **do**
14: Retransmit M_k to the receiver;
15: c_k++;
16: **end while**
17: Generate Key_{k+1} for P_{k+1} by using the array t as input to the privacy amplification with one-way hash function;
18: **end for**

The MIO encryption process algorithm is shown in Algorithm 1:

C. MIO Decryption

As shown in Fig. 5, when those encrypted symbols arrive at the legitimate receiver through the wireless channel, the receiver would conduct the MIO decryption process in two steps: (1) key checking and symbols decryption and (2) symbols key update at the receiver.

1) Key Checking & Symbols Decryption: Upon receiving signals by the legitimate receiver (or adversary), the RF down-converter samples the incoming signal, and observes a stream of discrete complex baseband symbol vectors. In MIO, for any given transmitted encrypted symbol ($E_{Key_{k,j}}(m_{k,i+j})$), the received encrypted symbol $y_{k,i+j}$ can be represented as:

$$y_{k,i+j} = H \cdot \hat{\theta} \cdot E_{Key_{k,j}}(m_{k,i+j}) + w_{k,i+j}, \tag{4}$$

where H and $w_{k,i+j}$ denote the wireless channel coefficient and Gaussian noise (in complex vectors), respectively. H is learned from the pilot sequence while $Key_{k,j}$ is updated from the previous data packet. The decrypted data symbol $\hat{y}_{k,i+j}$ can be computed as:

$$\begin{aligned}\hat{y}_{k,i+j} &= y_{k,i+j} - H \cdot \hat{\theta} \cdot Key_{k,j} \\ &= H \cdot \hat{\theta} \cdot (Key_{k,j} + m_{k,i+j}) + w_{k,i+j} - H \cdot \hat{\theta} \cdot Key_{k,j} \\ &= H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j}. \end{aligned} \tag{5}$$

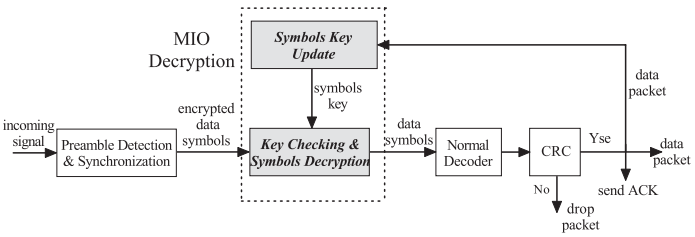


Fig. 5. The MIO decryption process at the legitimate receiver.

As described in Section IV-B1, the encrypted symbols blocks are randomly selected when a new packet (data symbols) goes to the symbols obfuscation & normalization block at the legitimate transmitter. This randomly pick-up mechanism can enhance the security level. However, at the receiver side, it would make the legitimate receiver hard to locate those encrypted symbols blocks due to (1) the positions of those encrypted symbols blocks cannot be carried in the last packet because the sizes of adjacent data packets are independent from one other and (2) the receiver cannot precisely determine whether the received symbols are the packet's data symbols at the physical layer during the wireless communications.[3]

To precisely discern those encrypted symbols blocks, the legitimate receiver adopts a *cross-correlation* operation with the assistance of the symbols key, called *key checking*. The cross-correlation value at position i with γ symbols key for k^{th} encrypted packet, $C(i, \gamma, k)$, can be computed as:

$$C(i, \gamma, k) = \left| \sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot y_{k,i+j} \right|, \tag{6}$$

where $\overline{Key}_{k,j}$ is the complex conjugate of $Key_{k,j}$. Assume $m_{k,i}$ is the first encrypted data symbol of one selected encrypted symbols blocks at transmitter A, by replacing $y_{k,i+j}$ in Eq. (6) with Eq. (4), the correlation value is:

$$C(i, \gamma, k) = |H| \cdot \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} \left| Key_{k,j} \right|^2 + |O(i, \gamma, k)|, \tag{7}$$

where

$$\begin{aligned}&|O(i, \gamma, k)| \\ &= \left| H \cdot \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot m_{k,i+j} + \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot w_{k,i+j} \right|. \end{aligned}$$

Since $Key_{k,j}$ is independent of either the data symbol $m_{k,i+j}$ or the noise symbol $w_{k,i+j}$, $|O(i, \gamma, k)| \approx 0$ [28]. In MIO, when the correlation value $C(i, \gamma, k)$ is larger than a threshold value β_c, the corresponding symbols are identified as the encrypted symbols, which is shown in Fig. 6. Normally, the threshold β_c can be defined as $\beta_c = \psi \cdot \gamma \cdot RSSI_{signal}$ [28], [29], where ψ is a constant (e.g., $\psi = 0.9$) and $RSSI_{signal}$ is the received signal

[3]In [27], it deploys a redundant "postamble" field to explore a packet's end at the physical layer. We do not consider this kind of field because it is not a standard field in current wireless communications.

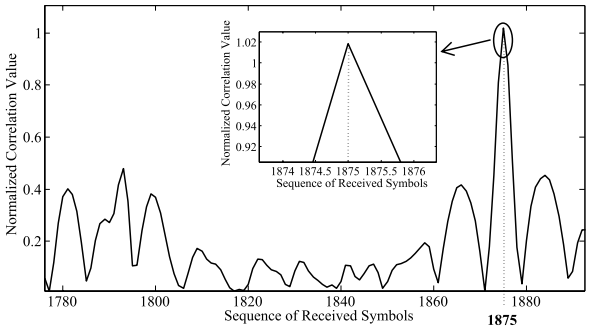


Fig. 6. Identification of one encrypted data symbols block with normalized cross-correlation: The normalized peak correlation value being larger than the threshold indicates the corresponding encrypted data symbols (e.g., from 1816^{th} to 1875^{th}, $\gamma = 60$) are obfuscated with the given symbols key.

strength indicator. Thus, the encrypted symbols' localization is conducted by checking if the inequation holds:

$$\frac{C(i, \gamma, k)}{\gamma \cdot \hat{\theta} \cdot RSSI_{signal}} \geq \psi. \tag{8}$$

It is clearly that by using this cross-correlation operation, the legitimate receiver can eliminate the channel noise influence to locate the correct position i for each encrypted symbols block without any packet information (e.g., the first symbol of the packet and the relative positions of the encrypted symbols blocks in the packet). This makes MIO more practical during wireless communications.

After identifying the position of an encrypted symbols block, the legitimate receiver can offset the symbols key by using Eq. (5) to calculate the clean data symbols in each block. We call this *symbols decryption*. To demodulate the clean data symbols same as the non-encrypted data symbols in the normal decoder (Fig. 5), the receiver has to increase the power of the clean data symbols by the factor $\frac{1}{\hat{\theta}}$. Thus, we can have:

$$\begin{aligned} \hat{y}_{k,i+j} &= \frac{1}{\hat{\theta}} \cdot \left(H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j} \right) \\ &= H \cdot m_{k,i+j} + \frac{w_{k,i+j}}{\hat{\theta}}. \end{aligned} \tag{9}$$

Note that the MIO requires frequent cross-correlation operations in the key checking and symbols decryption block to identify the encrypted symbols blocks, which introduces extra time and computation overhead on correlation calculations. However, the complexity of the key checking and symbols decryption block is at the same order as that of the preamble detection and synchronization block, because the preamble detection and synchronization block also deploys the cross-correlation to detect and synchronize the preamble.

2) Symbols Key Update at the Receiver: Once the data symbols are decrypted, the receiver maps all these plain data symbols to digital bits in the normal decoder block (Fig. 5) so that the channel coefficient and the noise (i.e., H and $\frac{w_{k,i+j}}{\hat{\theta}}$ in Eq. (9)) can be filtered out. After decoding the digital bits, receiver B will check if the packet P_k is correct through cyclic redundancy check (CRC) (With some small probability, it may contain undetected errors even if the packet passes the CRC checking. We detail this in Section VIII-B.). If the received data packet is correct, the packet acknowledgment will be sent back to transmitter A and this acknowledgment[4] will trigger A to update the symbols key for the next packet (Fig. 2). Synchronously, the symbols key for the next packet at receiver B will be updated exactly the same as at the transmitter side (Section IV-B2). Otherwise, the receiver drops the corrupted data packet and waits for the packet retransmission.

Algorithm 2 MIO Decryption Process

Input: Key_1 generated at the initialization stage; encrypted data symbols of the k^{th} packet P_k.
Output: the k^{th} packet P_k.

```
while receiving encrypted data packet P_k do
  if the first encrypted data symbol y_{k,i} is identified through the cross-correlation with symbols key Key_k (Eqs. (6) ~ (8)) then
    for j = 0 to γ − 1 do
      Calculate clean decrypted data symbol ŷ_{k,i+j} by Eqs. (5) and (9);
      y_{k,i+j} ← ŷ_{k,i+j};
      Append the position information i + j of ŷ_{k,i+j} in the array r;
    end for
  end if
  Map the received decrypted data symbols y_k to digital bits;
end while
if P_k passes the CRC check then
  Send PACK_k to the transmitter;
  Map P_k to L data symbols m_{k,0}, ..., m_{k,i}, ..., m_{k,L−1};
  Find the selected data symbols according to the position information in the array r, and store the data symbols into the corresponding positions in the array t;
  Generate Key_{k+1} for P_{k+1} by using the array t as input to the privacy amplification with one-way hash function;
else
  Discard P_k and wait for retransmission;
end if
```

It is noted that in the MIO decryption process, after filtering noises and channel coefficients, the digital bits which are mapped into the data symbols for the key updating are exactly the same as those for the transmitter. Associating the selected data symbols with their position information in the array r, the receiver can store the corresponding selected data symbols in the array t. Thus, it would guarantee that the array t at the receiver for the key updating is the same as the array at the transmitter. The MIO decryption process algorithm is shown in Algorithm 2.

V. Security Analysis

In this section, we first brief the computational secrecy of the initial key. Then, we demonstrate that, without considering the initial key, the MIO scheme can provide both information-theoretic secrecy to the passive eavesdropping

[4]The ACK may get lost during the transmission, which is dealt with in a similar way as the CRC checking.

attack in Section V-B and computational secrecy to the fake packet injection attack in Section V-C, respectively. Furthermore, we analyze the MIO's defense against several symbol detection attempts in Section V-D and the acknowledgment-based key disruption attack in Section V-E.

A. Computational Secrecy of the Initial Key

As we have described in Section IV-A, the MIO scheme has to use the conventional key agreement protocols to start the secure wireless communications. As the MIO scheme does not deploy any trusted third party to issue a certificate authority to the legitimate pairs, it would inevitably make the secrecy of the first symbols key, which is generated by the password-authenticated key agreement scheme, computationally bounded. Thus, the adversary with enough computational power can crack the first symbols key if the same symbols key has been used over a long time. Furthermore, if the eavesdropper can correctly receive all the subsequent encrypted data packets, it can determine all the symbols keys and crack the encrypted data packets. In this situation, the MIO's secrecy is bounded by the first key agreement scheme. Moreover, to avoid the long-term use of the pairwise authenticated password, the password also has to be updated when each communication session is completed. Please note that the MIO scheme is not restricted to the key agreement protocol described in this paper. It can apply to any bit-level key agreement schemes as the symbols key's parameters and the first symbols key can be generated from any bit-level keys by using the one-way hash function.

B. Information-Theoretic Secrecy against the Passive Eavesdropping Attack

In this section, we adopt the *secrecy capacity* model to prove that the MIO scheme can achieve the information-theoretic secrecy to the passive eavesdropping attack without considering the initial key. Normally, the secrecy capacity C_s is defined in [30] as:

$$C_s = C_M - C_T, \tag{10}$$

where C_M and C_T denote the Shannon capacities of the legitimate and eavesdropping channels, respectively. As MIO focuses on the real time wireless communications, by considering the realization of the quasi-static fading channels [12] and white Gaussian noise, the secrecy capacity in MIO can be written as:

$$C_s = \begin{cases} \log_2(1+\vartheta_M) - \log_2(1+\vartheta_T), & \text{if } \vartheta_M \geq \vartheta_T; \\ 0, & \text{if } \vartheta_M < \vartheta_T. \end{cases} \tag{11}$$

Here, ϑ_M and ϑ_T are the SNRs of the legitimate and eavesdropping channels, respectively. Obviously, the secrecy capacity C_s can be enlarged by either increasing ϑ_M or decreasing ϑ_T. It is proved that, when $C_s > 0$, the information-theoretic secrecy can be achieved [12], [30].

According to Eqs. (4) and (5), ϑ_M and ϑ_T are

$$\vartheta_M = \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2}, \tag{12}$$

$$\begin{aligned} \vartheta_T &= \frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot \left|m_{k,i+j}\right|^2}{\hat{\theta}^2 \cdot |H_E|^2 \cdot \left|Key_{k,j}\right|^2 + \left|w^T_{k,i+j}\right|^2} \\ &= \frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot \left|m_{k,i+j}\right|^2}{\alpha^2 \cdot \hat{\theta}^2 \cdot |H_E|^2 \cdot \left|m_{k,i+j}\right|^2 + \left|w^T_{k,i+j}\right|^2}, \end{aligned} \tag{13}$$

where H_L, H_E, $w^M_{k,i+j}$, $w^T_{k,i+j}$ are the channel coefficients and Gaussian noise of the legitimate and eavesdropping channels, respectively.

Generally, with no channel state information (CSI) of the eavesdropping channel, that is, the eavesdropping channel is totally unknown to the legitimate parties, Eq. (11) in MIO can be expressed as:

$$\begin{aligned} C_s &= \log_2(1+\vartheta_M) - \log_2(1+\vartheta_T) \\ &= \log_2\left(1 + \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2}\right) \\ &\quad - \log_2\left(1 + \frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot \left|m_{k,i+j}\right|^2}{\alpha^2 \cdot \hat{\theta}^2 \cdot |H_E|^2 \cdot \left|m_{k,i+j}\right|^2 + \left|w^T_{k,i+j}\right|^2}\right) \\ &> \log_2\left(1 + \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2}\right) - \log_2\left(1 + \frac{1}{\alpha^2}\right). \end{aligned} \tag{14}$$

Note that we assume the eavesdropping channel is white Gaussian noise free ($\left|w^T_{k,i+j}\right|^2 = 0$), which is obviously a worst case assumption, and if the secrecy capacity $C_s > 0$ in that case, it can always be considered as the secrecy capacity C_s can maintain a positive number with the white Gaussian noise.

From Eq. (14), when

$$\alpha \geq \frac{\left|w^M_{K,i+j}\right|}{\hat{\theta} \cdot |H_L| \cdot \left|m_{k,i+j}\right|}, \tag{15}$$

we can have

$$\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2} \geq \frac{1}{\alpha^2}$$

and

$$C_s > \log_2\left(1 + \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2}\right) - \log_2\left(1 + \frac{1}{\alpha^2}\right) \geq 0,$$

then, the requirement for the information-theoretic secrecy, i.e., $C_s > 0$, can be guaranteed [10], [11].

Note that $\vartheta_M = \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{\left|w^M_{k,i+j}\right|^2}$ is the SNR of the legitimate channel. In 802.11, this SNR's value should be larger than a threshold β_{SNR} (Normally, $\beta_{SNR} > 0dB$), so that legitimate transmitter and receiver can have normal communications, which means

$$\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot \left|m_{k,i+j}\right|^2}{\left|w^M_{k,i+j}\right|^2} > 1. \tag{16}$$

Combining Eqs. (14) and (16), when $\alpha \geq 1$, C_s in MIO can be a positive secrecy capacity. Associating with the privacy amplification with one-way hash function which is used in MIO's dynamic symbol key update mechanism, MIO can achieve information-theoretic secrecy without considering the first symbols key. Furthermore, different from other artificial noise approaches [2], [5]–[8], [22], MIO does not need any CSI of the eavesdropping channel. Most importantly, this secrecy capacity C_s will always stay positive regardless of the eavesdropper's location. Thus, MIO's information-theoretic secrecy will not be compromised by the location of the eavesdropper. However, this information-theoretic secrecy of the MIO scheme only holds under the assumption that the first symbols key is computationally unbounded.

C. Computational Secrecy against the Fake Packet Injection Attack

As the symbols key cannot be correctly derived from the received encrypted symbols, the attacker may attempt the brute-force strategy[5] to test all possible symbols keys to inject the fake packet, i.e., it has to try $\tau^{\frac{\gamma}{2}}$ combinations on average to test a symbols key, where γ is the length of the symbols key and τ is the total number of possible key symbols. Compared with standard AES and DES, in which the computational complexity of brute-forcing the key is $2^{\frac{\gamma}{2}}$ (with same key size γ), the MIO's computational complexity against the fake packet injection attack would be much greater than the traditional ones when τ is larger than 2. Fig. 7 gives the computational complexity for brute-forcing the symbols key with different key symbols number τ and key size γ. It is clearly that more computations are required to test the symbols key as τ increases. Moreover, the attacker has to inject $\tau^{\frac{\gamma}{2}}$ fake packets to achieve a successful fake packet injection attack, which is hardly true in wireless communications. According to Fig. 3, the total number of possible key symbols, τ, is determined by V_j and α, which can be much larger than 2. Thus, using the symbols key can achieve a better computational secrecy to the fake packet injection attack compared with the traditional bit-level key.

[5]In fact, we do consider some other attacks, such as the dictionary attack. As the privacy amplification with one-way hash function can effectively defend against these attacks, due to the pages limitation, we only provide the brute force attack's computational complexity in our paper.

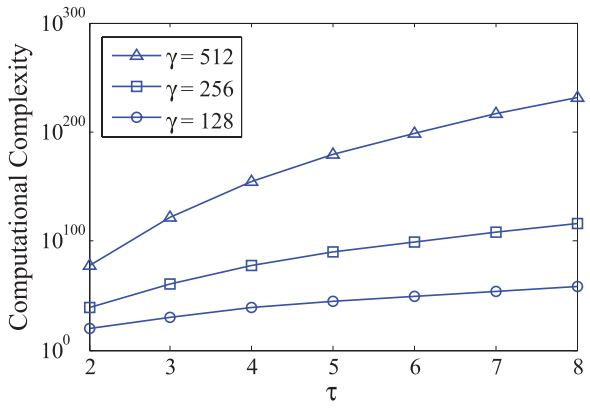


Fig. 7. The computational complexity for brute-forcing the key with various τ $(2 \leq \tau \leq 8)$ and key size γ.

D. Defense against Symbol Detection Attempts

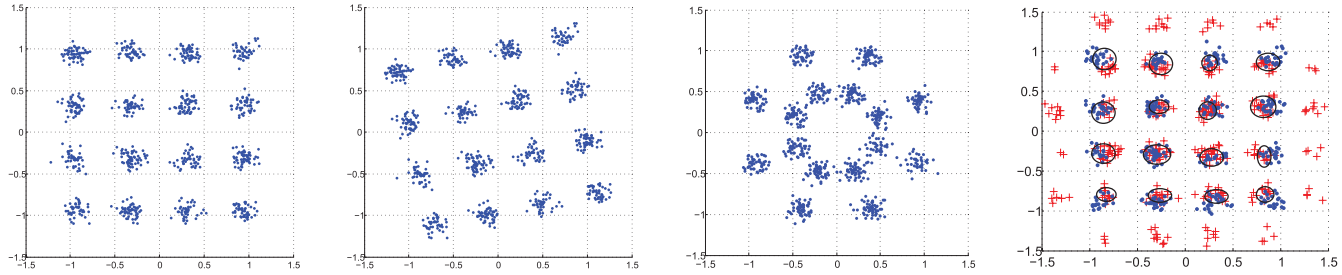
As the MIO encryption would change the location of the original data symbol in the constellation map, the eavesdropper may attempt to deploy symbol detection techniques, such as AMC [20], DMC [21] or constellation mapping [4], [9], to distinguish the encrypted/non-encrypted symbols. In this subsection, we explore whether MIO can defeat these symbol detection approaches.

AMC [20] is based on a cyclic feature that different digital modulations have different periodical information associated with time, which could be deployed by the attacker to distinguish different modulations, such as BPSK, QPSK and QAM. However, MIO does not belong to any digital modulations. Even with large amount of training data and supervised learning, AMC still cannot find the encrypted symbols from the received symbols.

DMC [21] uses the standard constellation shape as basis for finding received symbol's modulation. In its algorithm, the eavesdropper constructs a scatter constellation map of the received symbols and uses the fuzzy c-means clustering to recover the robust constellation map. The reconstruction of constellation map is based on the maximum likelihood with predefined digital modulation templates. Similar as AMC, DMC cannot identify the unknown constellation map template (i.e., the MIO's constellation map) from the received symbols. Also, it requires symbol training and supervised learning which cannot be done through a single packet [9].

As we discussed above, the traditional symbol detection techniques can hardly identify the encrypted symbols due to that they more focus on exploring the received symbols' modulation. Next, we explore another symbol detection attempt that is based on a large amount of received symbols: The attacker plots all the received symbols on the constellation map to check if the key can be disclosed from this constellation map. We simulate this constellation mapping method based on 16-QAM and compare the MIO with symbol flipping based encryption [4] and CD-PHY encryption [9].

As shown in Fig. 8(a), without encryption, the rectangular 16-QAM can be easily identified from the constellation map. So does the symbol flipping based encryption (Fig. 8(b)). Compared with Fig. 8(a), the key (rotation angle) can be



(a) (b) (c) (d)

Fig. 8. Comparison of constellation maps (SNR = 20 dB). (a) Standard 16-QAM rectangular constellation map. (b) Symbol flipping based encryption [4]. (c) CD-PHY based encryption [9]. (d) MIO based encryption ("." denotes non-encrypted data symbols, "+" denotes encrypted symbols.)

easily conquered from the figure. The same problem exists in the CD-PHY modulation (Fig. 8(c)). Although the attacker may not know the symbol-to-bit mapping yet, the constellation diversity key can be easily identified from the large amount of symbols' plotting. However, due to the encryption feature that the MIO will mix up the normalized encrypted symbols with the non-encrypted symbols (Fig. 8(d)), the attacker can hardly identify the encrypted symbols from non-encrypted symbols. Even with clear evidences that the received symbols are manipulated by the MIO scheme, the attacker cannot correctly locate the positions of the encrypted symbols blocks because they are mixed up with data symbols (the circled regions in Fig. 8(d)). Consequently, the eavesdropper fails to conquer the symbols key from this symbol detection attempt.

E. Acknowledgment-Based Key Disruption Attack

As described in Sections IV-B2 and IV-C2, the acknowledgment plays a crucial role in MIO's real time wireless communications. Each data packet is acknowledged before the next data packet is prepared to be transmitted since the previous data packet generates the symbols key to decrypt the data symbols in the next data packet. As a result, an adversary might try to disrupt the MIO's key updating by sending the ACK packets to the legitimate transmitter even though the legitimate receiver fails to receive the correct data packet. Upon the receipt of the ACK, the transmitter might send the next data packet to the receiver, which has not received the previous correct data packet for the key update (Line 15 in Algorithm 2). Therefore, the receiver fails to retrieve any subsequent data packet from the transmitter. Such attack is called *acknowledgment-based key disruption attack*.

In MIO, a similar symbols key update mechanism (Sections IV-B.2 and IV-C.2) for the ACKs is adopted to defend against the acknowledgment-based key disruption attack. In this case, this attack will be the same as the packet injection attack except the roles of the transmitter and receiver are swapped. Although the size of ACK is not large as the size of the data, the privacy amplification with one-way hash function can still generate significantly different symbols key even the input is quite similar [25], [31], [32]. The symbols key which encrypts the ACK's symbols is updated for each ACK. The transmitter will deny the ACK which is not encrypted by the correct ACK's symbols key through the cross-correlation in Section IV-C. Moreover, even if an adversary might occasionally guess correctly an ACK's symbols key and inject a fake ACK, leading to an acknowledgment-based key disruption attack, the legitimate receiver will be aware of the attack because it cannot receive the subsequent data packets any more.

VI. Implementation Issues

Some implementation issues in MIO are further analyzed as follows:

A. Symbols Key Checking

The symbols key checking is a crucial part in the MIO system (Section IV-C). Missing the first encrypted symbol (false negative error) or finding a wrong symbol (false positive error) is fatal to the MIO's decryption. From Eq. (8), the γ and ψ are key parameters in symbols key's identification. Enlarging both γ and ψ can minimize the false positive/negative errors [28]. In our testbed, when $\psi = 0.75 \sim 0.90$ and $\gamma \geq 40$, the false positive/negative errors can be reduced to 0.

Additionally, if the packets are transmitted in a low SNR environment, these packets cannot be correctly demodulated. Consequently, there is no need to find the first encrypted symbol. Therefor, to minimize the computational cost of symbol decryption under the low SNR environment, Eq. (8) can be changed to:

$$\frac{C(i,\gamma,k)}{\gamma \cdot max\left(RSSI_{signal},\ \beta_{SNR} + RSSI_{noise}\right)} \geq \psi. \quad (17)$$

Here, $RSSI_{noise}$ is the environment noise (typically, $-98 \sim -95dBm$) and β_{SNR} is the SNR threshold that can correctly decode a packet. If the $RSSI_{signal}$ is not high enough to decode packets correctly, by using Eq. (17), the receiver would drop this packet because it would not pass the symbols key checking.

B. dB Loss in MIO

In the MIO system, the legitimate transmitter normalizes the encrypted symbol $E_{Key_{k,j}}(m_{k,i+j})$ by multiplying the normalization coefficient θ (Line 8 in Algorithm 1). As a result, this will attenuate the transmission power of the data

symbol. We call such power loss as *dB loss* in MIO, which can be computed as:

$$\begin{aligned} &dB_{loss} \\ &= Original\ data\ energy - Normalized\ data\ energy \\ &= 10 \cdot \lg\left(|m(k, i+j)|^2\right) - 10 \cdot \lg\left(\left|\hat{\theta} \cdot m(k, i+j)\right|^2\right) \\ &= -20 \cdot \lg \hat{\theta}. \end{aligned} \tag{18}$$

Obviously, associated with Eq. (3), the dB loss is closely related to α. When α increases, there is more dB loss in the encrypted symbol normalization.

VII. Performance Evaluation

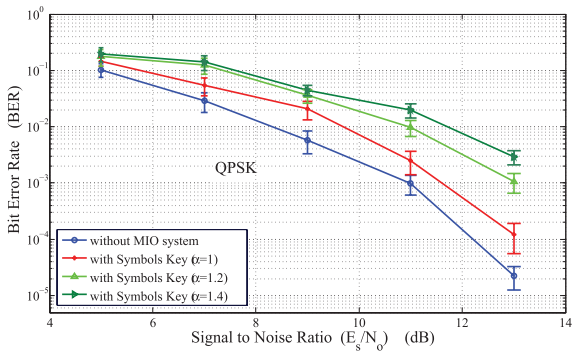
We employ 5 USRP2 [14] with 2.4GHz-based RFX2400 daughter-board, and each USRP2 is connected to a notebook. All the notebooks are operated under Ubuntu 10.04 and installed the GNURadio software [33]. We evaluate the signal to noise ratio (SNR) by calculating the $SNR \approx RSSI_{signal} - RSSI_{en}$, where $RSSI_{signal}$ and $RSSI_{en}$ are the received signal strength indicators of the transmitted signal and the environment noise, both of which are measured from the USRP2 hardware. As we can adjust the transmitter's sending power and distance between the transmitter and receiver, we can get various $SNRs$ to test the MIO's performance. We choose QPSK as our data modulation scheme, which is widely adopted in the IEEE 802.11 standards for WLAN implementation [2]. For the key symbols, we take the key angles to be $\{\frac{\pi}{2}, -\frac{\pi}{2}\}$ and $\alpha = 1, 1.2$ and 1.4 to test the performance of the MIO scheme. The data packet consists of an 80-bit preamble, and a 200-byte payload, which means there are 800 data symbols for each packet. For each run of the experiment, 5000 data packets are sent and we repeat the experiment for 12 times. Also, we simulate the MIO scheme using the MATLAB with Simulink.

A. Experimental Results

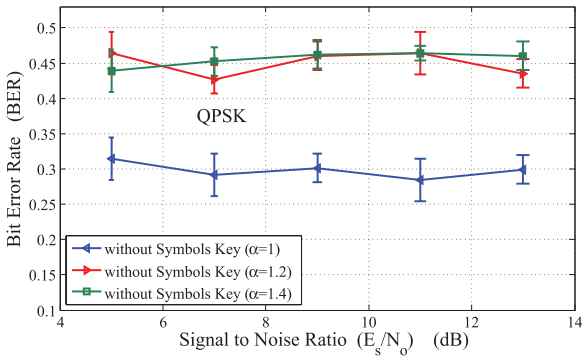
In the hardware experiments, we first verify the BER performance at the legitimate receiver with different α values. Compared with the BER in a non-MIO scenario, the decrypted data at the legitimate receiver has a slight dB loss ($1 \sim 1.6 dB$) when $\alpha = 1$ (Fig. 9(a)). However, this dB loss at the legitimate receiver would go up to $2.5 dB$ and $3.7 dB$ when $\alpha = 1.2$ and $\alpha = 1.4$, respectively.

At the eavesdropper side, without knowing the symbols key, the BER would remain constant, roughly at 0.27 (when $\alpha = 1$) no matter what SNR is adopted (Fig. 9(b)). We believe this BER can already ruin the packet reception at the eavesdropper. Moreover, this BER does not increase along with the rising value of α and it stabilizes around 0.48 when $\alpha > 1$.

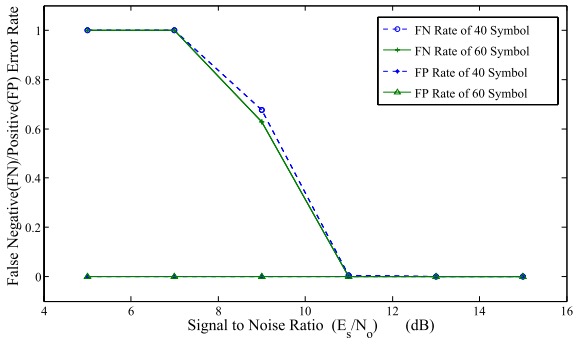
Regarding the symbols key checking, two sizes of symbols key ($\gamma = 40$ and $\gamma = 60$) are testified. The result shows that the false negative (FN) error rate and false positive (FP) error rate can be 0% at the legitimate receiver when $\beta_{SNR} = 11 dB$ (Fig. 9(c)). It guarantees that the legitimate receiver would not make any mistakes at the key checking process when the SNR is good enough to correctly decode the packet.[6]



(a)



(b)



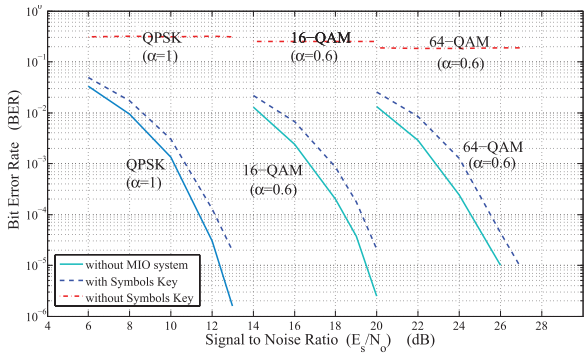
(c)

Fig. 9. Experimental results based on the USRP2 testbed. (a) Bit error rate at the legitimate receiver (with symbols key, $\gamma = 800$). (b) Bit error rate at the eavesdropper (without symbols key, $\gamma = 800$). (c) False negative/positive error rate at the legitimate receiver with two different symbols key sizes under different SNRs. ($\alpha = 1$, $\beta_{SNR} = 11 dB$, $\psi = 0.8$).

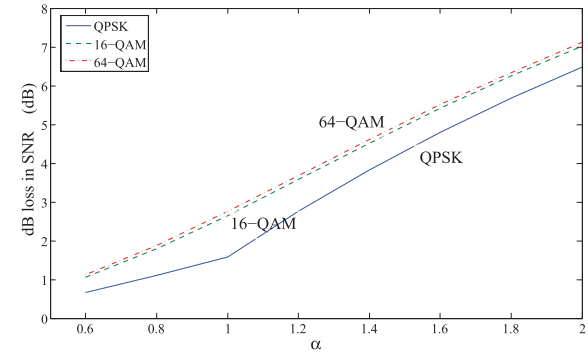
B. Simulation Results

To fully evaluate the MIO scheme's performance, we simulate the MIO scheme in various digital modulations (QPSK, 16/64-QAM). Fig. 10(a) depicts the BER between the non-MIO scenario and the MIO one. In regard to 16/64-QAM,

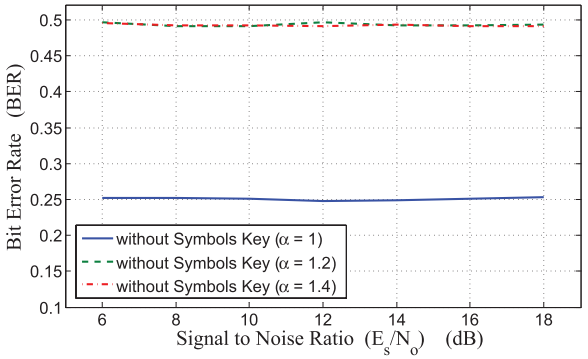
[6]This SNR value is much related to the modulation and channel coding [34]. For example, for the BPSK modulation with gray coding, the SNR requires above 9.7 dB for correctly receiving a packet. This value becomes 11 dB if QAM-16 is adopted.



(a)



(b)



(c)

Fig. 10. Simulation results. (a) Bit error rate (QPSK: $\alpha = 1$; 16/64-QAM: $\alpha = 0.6$). (b) The dB loss in the MIO system. (c) The relationship among BER, SNR and α (QPSK).

it is clear that the MIO system does not need to suffer too much dB loss ($1.3dB$ when $\alpha = 0.6$) to meet the encryption requirement. Please note that, in the IEEE 802.11, the QAM modulation requires high SNRs ($\beta_{SNR} > 11dB$) to demodulate the bits, as a result, $\alpha = 0.6$ can still guarantee a positive secrecy capacity. Also, in the QPSK modulation, it has the same BER trend for the testbed as the one for the simulation. Furthermore, without knowing the symbols key, the BERs are 0.27 (using 16-QAM) and 0.18 (using 64-QAM). Those BERs are enough to ruin the packet reception even with a channel coding [35].

As described in Section VI-B, the dB loss in the MIO scheme should be carefully considered in the system implementation. Although, in Section V-B, it has been proved that when $\alpha = 1$, the positive secrecy capacity can be guaranteed, we still simulate the dB loss trend as long as α increases. Fig. 10(b) reveals that the dB loss is increasing with the rising value of α.

Also, we simulate the BER performance at the eavesdropper side by using the QPSK modulation. Compared with Fig. 9(b), which is the hardware experimental result, the simulation result in Fig. 10(c) shows that the BER at the eavesdropper side is stabilized around 0.25 (when $\alpha = 1$) and 0.5, which reaches at the worst case of BER (when $\alpha > 1$). This simulation result is almost the same as the hardware experimental result.

Moreover, from Figs. 10(b) and 10(c), when $\alpha > 1$ ($\alpha = 1.2$ and $\alpha = 1.4$), the BER is stabilized around 0.5, but the dB loss is around $3dB$ and $3.8dB$ (using QPSK), respectively. The similar results are also shown for the USRP2 testbed (Fig. 9(a) and Fig. 9(b)). Thus, increasing α would not increase the BER, but only enlarge the gap of the dB loss in the MIO system when the BER reaches the maximum (i.e., BER = 0.5).

VIII. Discussion

We further discuss some issues arisen from the MIO scheme that remain unaddressed in this paper.

A. Dynamic Key Updating vs. Static Key Updating

Most existing security schemes would adopt a static key updating algorithm to generate the key. Once the legitimate users know the initial key, it would use the key generation algorithm to calculate the corresponding keys for synchronous key updating at both the legitimate transmitter and receiver. However, MIO adopts a dynamic key updating mechanism to generate the symbols key. As described in Sections IV, MIO's dynamic key updating contains two aspects of dynamic updating: (1) the new key is generated from the current encrypted packet and (2) the encrypted data symbols are randomly and independently picked up from data packets.

Note that the new symbols key is generated from the current decrypted data symbols (Sections IV-B2 and IV-C2). Even if the eavesdropper is aware of the perfect CSI of the legitimate channel, it is hard for the eavesdropper to calculate the symbols keys from one to another, because the eavesdropper is nearly infeasible to correctly receive all subsequent encrypted packets to track the symbols keys in real time wireless environments with background noises. When the decrypted packet cannot pass the CRC checking at the legitimate receiver, it is dropped and will be retransmitted again (within maximum retransmissions). However, if this happens at the eavesdropper side, because the dynamic key cannot be calculated from the previous keys, the eavesdropper fails to update the new symbols key, and loses the chance to decipher the subsequent encrypted packets. Thus, under the condition that the eavesdropper hardly receives all encrypted packets correctly, even if the eavesdropper knows the first symbols key, MIO can still achieve the information-theoretic secrecy when a transmission error occurs at the eavesdropper. Fig. 11 gives the bit/packet error rates under different SNRs. It shows that, even under a high SNR environment

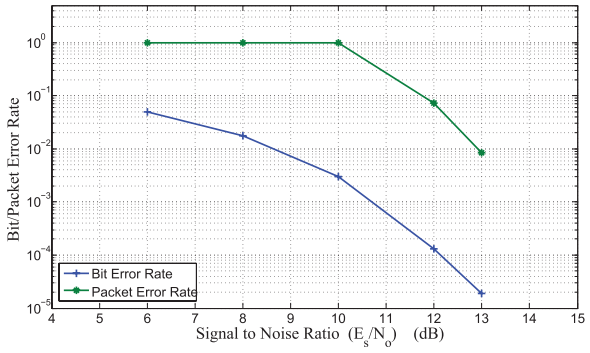


Fig. 11. Bit/packet error rates with symbols key in MIO. (QPSK modulation with $\alpha = 1$).

(SNR $= 13dB$), the packet error occurs within 1000 packet transmissions, which means that, if this error occurs at the legitimate receiver, the receiver will wait for the retransmission of the packet; however, if this error occurs at the adversary side, the adversary will lose the chance to track down the new symbols key, and the subsequent encrypted packets can achieve the information-theoretic secrecy. Note that in this case the former correctly received encrypted packets can be cracked by the eavesdropper.

B. CRC Checking Failure

As we have mentioned in Section IV-C, an incorrect packet may pass the CRC checking with a small probability, which may disrupt the synchronization process of symbols key update between the transmitter and receiver. This only happens if the incorrect bits are used for the key updating. To solve this CRC checking failure, error correct coding can be built in the channel coding procedure. However, this would increase the redundant part in the packet and decrease the network's throughput.

For MIO, the symbols key initialization procedure can solve this CRC checking failure. After the maximum retransmissions, the legitimate transmitter would go back to initialize the first symbols key Key_1 for the legitimate receiver (Section IV-A). As the unique private wireless channel parameter has changed in the wireless environment, the symbols key initialization procedure would not compromise the MIO's security.

C. Influence to the Network Throughput

The MIO scheme requires the symbols key to be synchronously updated at both the legitimate sides. It inevitably causes the legitimate transmitter to wait for a feedback by which the legitimate receiver acknowledges the correct reception of an encrypted packet. Then, the ACK can trigger the key update process at both the legitimate sides. This suggests that the whole communication system which implements the MIO scheme actually adopts the DATA-ACK model.

In current IEEE 802.11 MAC protocols, two standard mechanisms, CSMA/CA and RTS/CTS, are used to handle the packet collision problem in the wireless environments. The DATA-ACK model is also implemented for the unicast transmission scenario in the 802.11 MAC protocols.[7] The MIO scheme adopts the same DATA-ACK model for the packet unicast transmissions, consequently, the network throughput of the MIO scheme is the same as the standard 802.11 MAC protocols in the packet unicast scenario.

However, for the broadcast or multicast scenarios, as the MIO scheme requires the synchronized symbols key to secure the wireless communications, it does not work in the broadcast or multicast scenarios since the transmitter does not require any ACKs to acknowledge the packet transmissions. A possible solution to this problem is that the MIO scheme can force the packet transmission in the broadcast or multicast scenarios to adopt the packet unicast transmission mechanism. However, it will significantly affect the packet reception, as in the broadcast or multicast scenarios, a packet is transmitted once to all or many receivers, but in the unicast scenario, the packet will be transmitted n times to n receivers. We leave this MIO broadcast and mutilcast problem as our future work.

IX. Conclusions

In this paper, we propose a multiple inter-symbol obfuscation (MIO) scheme to secure the wireless transmissions between two legitimate entities. MIO does not need any trusted third party to interfere the packet interception by the eavesdropper or static channel condition to cancel artificial noises. Rather, it employs the data symbols from the previous data packets to generate the symbols key which obfuscates the current data packets. By dynamically updating the symbols key as the packets are disseminated, it is hard for an adversary to brute-force the symbols key by intercepting a number of encrypted symbols and analyzing them off-line. We establish the mathematical model for MIO, and prove that MIO can provide both the information-theoretic secrecy and computational secrecy without considering the initial key. Additionally, the experimental results reveal that without knowing the symbols key, the BER in the MIO scheme can effectively ruin the packet reception at the eavesdropper side, and the key checking process would defend against the packet injection attack in wireless networks.

References

[1] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Proc. 15th Annu. Int. Conf. ACM MobiCom*, Sep. 2009, pp. 321–332.

[2] 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*, Aug. 2011, pp. 2–13.

[3] Y.-S. Shiu, S. Y. Chang, H.-C. Wu, S. C.-H. Huang, and H.-H. Chen, "Physical layer security in wireless networks: A tutorial," *IEEE Wireless Commun.*, vol. 18, no. 2, pp. 66–74, Apr. 2011.

[4] C. Pöpper, N. O. Tippenhauer, B. Danev, and S. Capkun, "Investigation of signal and message manipulations on the wireless channel," in *Proc. ESORICS*, Sep. 2011, pp. 40–59.

[5] C. Sperandio and P. G. Flikkema, "Wireless physical-layer security via transmit precoding over dispersive channels: Optimum linear eavesdropping," in *Proc. IEEE MILCOM*, Oct. 2002, pp. 1113–1117.

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

[7] With the RTS/CTS, the unicast transmission is implemented with the RTS-CTS-DATA-ACK model.

[7] L. Lai and H. El Gamal, "The relay–eavesdropper channel: Cooperation for secrecy," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 4005–4019, Sep. 2008.
[8] S. Gollakota and D. Katabi, "Physical layer wireless security made fast and channel independent," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1125–1133.
[9] M. I. Husain, S. Mahant, and R. Sridhar, "CD-PHY: Physical layer security in wireless networks through constellation diversity," in *Proc. IEEE MILCOM*, Oct./Nov. 2012, pp. 1–9.
[10] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.
[11] I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.
[12] M. Bloch, J. Barros, M. Rodrigues, and S. McLaughlin, "Wireless information-theoretic security," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2515–2534, Jun. 2008.
[13] F. Oggier and B. Hassibi, "The secrecy capacity of the MIMO wiretap channel," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4961–4972, Aug. 2011.
[14] Ettus Research. (2015). *Universal Software Radio Peripheral*. [Online]. Available: http://www.ettus.com
[15] D. León, S. Balkir, M. Hoffman, and L. C. Pérez, "Fully programmable, scalable chaos-based PN sequence generation [CDMA]," *Electron. Lett.*, vol. 36, no. 16, pp. 1371–1372, Aug. 2000.
[16] T. Li, J. Ren, Q. Ling, and A. Jain, "Physical layer built-in security analysis and enhancement of CDMA systems," in *Proc. IEEE MILCOM*, Oct. 2005, pp. 956–962.
[17] C.-C. Chen, K. Yao, K. Umeno, and E. Biglieri, "Design of spread-spectrum sequences using chaotic dynamical systems and ergodic theory," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 9, pp. 1110–1114, Sep. 2001.
[18] S. Bhashyam and B. Aazhang, "Multiuser channel estimation and tracking for long-code CDMA systems," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1081–1090, Jul. 2002.
[19] R. Liu, Y. Liang, H. V. Poor, and P. Spasojević, "Secure nested codes for type II wiretap channels," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2007, pp. 337–342.
[20] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits Syst. Mag.*, vol. 9, no. 2, pp. 27–45, Jun. 2009.
[21] B. G. Mobasseri, "Digital modulation classification using constellation shape," *Signal Process.*, vol. 80, no. 2, pp. 251–277, Feb. 2000.
[22] M. L. Jorgensen, B. R. Yanakiev, G. E. Kirkelund, P. Popovski, H. Yomo, and T. Larsen, "Shout to secure: Physical-layer wireless security with known interference," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 33–38.
[23] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, May 1992, pp. 72–84.
[24] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise," in *Proc. 1st ACM Conf. CCS*, Dec. 1993, pp. 244–250.
[25] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[26] V. Goyal, A. O'Neill, and V. Rao, "Correlated-input secure hash functions," in *Proc. 8th Conf. Theory Cryptogr.*, Mar. 2011, pp. 182–200.
[27] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *Proc. ACM SIGCOMM*, Aug. 2007, pp. 409–420.
[28] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," in *Proc. 16th Annu. Int. Conf. ACM MobiCom*, Sep. 2010, pp. 25–36.
[29] S. Gollakota and D. Katabi, "ZigZag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, Aug. 2008, pp. 159–170.
[30] S. Leung-Yan-Cheong and M. E. Hellman, "The Gaussian wire-tap channel," *IEEE Trans. Inf. Theory*, vol. 24, no. 4, pp. 451–456, Jul. 1978.
[31] D. R. Stinson, "Universal hashing and authentication codes," *Designs, Codes, Cryptogr.*, vol. 4, no. 3, pp. 369–380, Oct. 1994.
[32] G. Van Assche, *Quantum Cryptography and Secret-Key Distillation*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[33] GNU GPL. (2015). *GNU Radio—The Free and Open Source Software Radio Project*, [Online]. Available: http://www.gnuradio.org
[34] J. Thomson *et al.*, "An integrated 802.11a baseband and MAC processor," in *IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, Feb. 2002, pp. 126–127.
[35] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 2007.

Tao Xiong received the B.E. degree in computing science from the China University of Geoscience, China, in 2003, and the M.E. degree in computer science and engineering from the University of New South Wales, Australia, in 2008. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests focus on cross-layer protocol design and implementation for wireless networks.

Wei Lou received the B.E. degree in electrical engineering from Tsinghua University, China, in 1995, the M.E. degree in telecommunications from the Beijing University of Posts and Telecommunications, China, in 1998, and the Ph.D. degree in computer engineering from Florida Atlantic University, USA, in 2004. He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. He has worked intensively on designing, analyzing and evaluating practical algorithms with the theoretical basis, and building prototype systems. His current research interests are wireless networking, mobile ad hoc and sensor networks, peer-to-peer networks, and mobile cloud computing. His research work is supported by several Hong Kong GRF grants and The Hong Kong Polytechnic University ICRG grants.

Jin Zhang received the B.E. and M.E. degrees in applied mathematics from the Harbin Institute of Technology, China, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests are peer-to-peer streaming, mobile cloud computing, and game theory.

Hailun Tan (M'10) received the M.S. degrees in telecommunication engineering and information technology from Australian National University, in 2004 and 2005, respectively, and the Ph.D. degree from the University of New South Wales (UNSW), in 2009. From 2004 to 2005, he was a part-time Software Engineer for the joint hardware demonstrator project with the CSIRO ICT Centre, and CSIRO Energy Technology. From 2005 to 2006, he was a Research Assistant for the Smart Road and Traffic project with National Information Communication Technology for Excellence, Australia. From 2009 to 2012, he was a Research Associate with UNSW for wireless sensor monitoring. He has been a Senior Technical Consultant with the Australian Federal Government for key security infrastructure development and support since 2013. His main research interests are in security protocol design in wireless mesh networks and wireless sensor networks.