Hide and Seek: Waveform Emulation Attack and Defense in Cross-Technology Communication

Xiaonan Zhang*, Pei Huang*, Linke Guo* and Yuguang Fang†
*Department of Electrical and Computer Engineering, Binghamton University,
State University of New York, Binghamton, NY 13902, USA
†Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
Email: {xzhan167, phuang13, lguo}@binghamton.edu, fang@ece.ufl.edu

***Abstract*—Cross-Technology Communication (CTC) is envisioned to serve as an effective approach to address the ever-increasing spectrum shortage and interference issue in the already crowded ISM band. Given the fact that the number of Internet of Things (IoT) devices has increased exponentially, CTC becomes a viable solution to enable direct communication among heterogeneous wireless devices, and thus provide reliable data transmission. However, CTC may offer opportunities for adversaries to manipulate IoT devices. In this paper, we identify a new attack built on CTC, where the WiFi device is able to hide the pre-intercepted ZigBee control message into the signal to achieve the objective of controlling the ZigBee device by sending the WiFi emulation signal. To defend against this attack, we analyze the constellation using higher-order statistics at the ZigBee receiver for detection. Extensive experiments using the commodity devices (CC26x2R1) and the USRP-based prototype show the existence of the newly identified attack, and further validate the effectiveness of the proposed defensive approach while maintaining a very low false alarm rate.**

***Index Terms*—Cross-Technology Communication, IoT, Emulation Attack, Physical-Layer Defense**

I. Introduction

The proliferation of Internet of Things (IoT) applications enables ubiquitous connections among various wireless devices for bettering our daily life. According to a recent report [1], the number of IoT devices is expected to reach 55 billion by 2025, which will pose significant challenges on spectrum resources. Current IoT devices deploy different wireless technologies. Some of them share the same spectrum resources when they coexist in the common space. For example, IoT devices using the WiFi, ZigBee, and Bluetooth protocols occupy the Industrial, Scientific, and Medical (ISM) 2.4 GHz band, leading to intense coexistence of wireless technologies. Due to their incompatibility, multiple costly and device-independent gateways are always needed to fully connect IoT devices from different manufacturers. Nevertheless, the deployment of gateways not only incurs extra hardware costs, but also introduces more traffic overhead and longer communication delay. As one of the most promising paradigm, Cross-Technology-Communication (CTC) allows the direct communication among devices across different wireless technologies [2]–[4].

Unfortunately, the usage of CTC could potentially bring severe security concerns. Assuming the WiFi transmitter is an attacker or has been compromised by an attacker, it would be able to send a "well-prepared" packet in the same frequency band to control the Bluetooth or ZigBee receiver via CTC. It is worth noting that existing higher-layer cryptographic approaches do not work because most CTC happens in the physical layer, in the sense that most receivers get compromised soon after they receive the packet. Even worse, the wide deployment and longer transmission range render larger rooms for WiFi devices to attack the short-ranged Bluetooth and ZigBee IoT devices, such as enabling the cooling on smart thermometer during winter, unlocking the smart garage door, and turning off the security camera for break-in, etc. Given the fact that the deployment of IoT devices increases dramatically, it is critical to detect this type of attack and design an effective countermeasure to mitigate the potential threats.

In this paper, we identify a new attack named as **CTC Waveform Emulation Attack**, where a WiFi attacker pre-intercepts the control message from the communication between ZigBee devices and further *hide*s the control message into the signal to manipulate the functionality of ZigBee devices. The WiFi emulation signal is able to pass the decoding and demodulation process by the ZigBee receiver, and thus it is infeasible to be detected. As a countermeasure, we propose a new defensive strategy to *seek* the malicious WiFi attacker based on the constellation recognition. Specifically, our contribution is listed as follows,

- To the best of our knowledge, we are the first to discover this new attack. We have demonstrated the practicality of the waveform emulation attack from a WiFi device to a ZigBee device, where the WiFi emulation signal is able to bypass higher-layer protocols and further control the ZigBee device.
- An effective and efficient defensive strategy is proposed to identify the WiFi emulation signal from the authentic ZigBee signal. To be more specific, we deploy higher order statistics to analyze the constellation diagram of the received signal for identification purposes.
- Extensive simulations and experiments are conducted in both the AWGN and real environments. The results demonstrate the existence of the CTC waveform emulation attack together with the effectiveness of the proposed defensive strategy.

The rest of this paper is organized as follows. Sec. II presents the related work. In Sec. III, we demonstrate the motivation of the proposed CTC waveform emulation attack and its adversarial model. The details of the waveform emulation attack are introduced in Sec. IV while its corresponding dependence strategy is detailed in Sec. V. We evaluate the performance of both the emulation attack and its defensive strategy in Sec. VI, followed by the conclusion in Sec. VII.

II. Related Work

A. Cross-Technology Communication

Existing works on Cross-Technology Communication (CTC) mostly focus on how to improve the communication throughput and alleviate the cross-technology interference. B^2W^2 [3] enables the high throughput and long distance concurrent N-way cross-technology communication between Bluetooth low energy and WiFi by leveraging channel state information. Zheng *et. al* in [5] discuss interference-resilient CTC in coexisting environment. In FreeBee [6], Esense [7] and GSense [8], the communication between WiFi and ZigBee devices is enabled using RSS to measure the WiFi signal. Different from existing CTCs deploying packet-level modulation using the packet length [7], timing [6], and sequence patterns [9], [10], Li *et. al* in [2] propose a physical-level emulation technique, which motivates our newly identified attack.

B. Constellation Recognition

Automatic modulation classification (AMC) of digital modulations mounts to identify the constellation used by a digital communication system [11]. Generally, AMC algorithms can be categorized into two classes, relying on likelihood function or features of the received signal [12]. As for the QPSK constellation recognition, a hybrid likelihood ratio test (HLRT) structure is utilized to classify QPSK and BPSK modulation with unknown parameter signal level and the angle of arrival in [13], [14] respectively. Second - and fourth - order moments of the received signal were applied to distinguish between QPSK and 16-QAM in [15]. Similar but different, second and fourth order cyclic cumulants are deployed to differentiate the QPSK, 16QAM and 64QAM constellations in [16], [17]. Since the feature-based cumulant analysis has lower complexity than the likelihood function in classifying the modulation [12], we consider the cumulant analysis in our work.

III. Adversarial Model

A. Motivation

The Cross-Technology Communication (CTC) enables direct communication between heterogeneous wireless devices using different protocols. Given the above facts on CTC, it is highly possible for attackers to mimic the designated ZigBee packets, and then intentionally control passive IoT devices. Specifically, we identify a new type of attack, **CTC Waveform Emulation Attack**, where a WiFi device leverages CTC to control the ZigBee device while bypassing the original ZigBee gateway. Due to the lack of detection methods, the ZigBee device is unable to distinguish whether the control message is coming from the authentic gateway or the malicious WiFi device, and thus severe consequences may occur along with the controlled devices.

To launch the CTC waveform emulation attack, a malicious WiFi attacker first creates the desired time-domain waveform, and then makes the ZigBee receiver believe that the received signal is coming from the legal ZigBee transmitter. This newly identified attack is very critical and needed to be mitigated due to the following reasons: 1) the waveform emulation attack fools the passive ZigBee device from the physical-layer, so the existing higher-layer cryptographic method cannot detect it; 2) WiFi devices have longer transmission distance $(\max.\ 100m)$ than ZigBee devices $(1-10m)$, where WiFi attackers can launch the attack without being noticed in the line-of-sight (LoS); 3) the wide deployment of WiFi-enabled mobile devices extends opportunities for launching the attack.

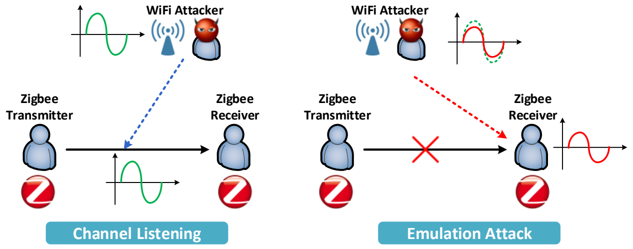


Fig. 1: CTC Waveform Emulation Attack Process

B. Adversarial Model

We give an example to demonstrate our adversarial model in Fig.1. Two ZigBee devices work at the central frequency 2435MHz with 2MHz bandwidth whereas WiFi devices occupy the 20MHz bandwidth centered at the frequency 2440MHz. The attacking process consists of two steps as follows.

1) Channel Listening: In the time slot t_1, a pair of ZigBee devices communicate with each other (e.g., a ZigBee gateway and a smart light bulb), where a WiFi attacker located close to the ZigBee receiver eavesdrops the ZigBee communication channel. Since the spectrum of ZigBee and WiFi devices are overlapped, the WiFi attacker would be able to observe and record the time-domain waveform from the ZigBee transmitter. In particular, we assume that no other devices occupy the overlapped spectrum and the WiFi attacker knows the beginning of the received ZigBee time-domain waveform.

2) Waveform Emulation Attack: In the time slot t_2, when the WiFi attacker confirms that the ZigBee device does not transmit the signal, it emulates the received ZigBee waveform and then transmits it to the ZigBee receiver. After receiving the “legal and authentic” time-domain waveform, the ZigBee receiver continues the higher layer processing. The WiFi attacker achieves its objective of controlling the ZigBee device.

IV. ZigBee Waveform Emulation Attack

In this section, we first describe some basic principles of ZigBee and WiFi protocols, and then illustrate how the WiFi device emulates a time-domain waveform such that it can be correctly received and demodulated by the ZigBee device.

A. ZigBee Transmitter and Receiver

As shown in Fig. 2, we first briefly review how the ZigBee device transmits and receives packets. For physical-layer transmission, the ZigBee transmitter packages the data from the MAC layer and adds a prefix (e.g., 0x0007A) to each packet header. Then, Direct Sequence Spread Spectrum (DSSS) is used to improve interference and noise resilience by multiplying original bits with a pseudo-random noise spreading code. Specifically, each 4-bit ZigBee symbol is mapped into a 32-chip sequence, followed by the Offset Quadrature Phase-Shift Keying (OQPSK) modulation. OQPSK offsets the timing of the odd and even chips by one chip-period and maps the new pair of chips in each chip-period into QPSK symbols, which are sent in every $0.5\mu s$. Hence, the duration of each ZigBee symbol will last $16\mu s$. At the ZigBee receiver, after OQPSK demodulation and clock recovery, every 32 chips are collected and mapped into one ZigBee symbol according to the predefined symbol-to-chip spreading relationship in the DSSS process. Specifically, in DSSS, a correlation threshold is defined to control the maximum Hamming distance between the received 32-chip sequence and the predefined chip sequence that the receiver can tolerate [18]. If the Hamming distance is less than the threshold, the received chip sequence is decoded to the corresponding ZigBee symbol. Otherwise, the chip sequence is dropped.

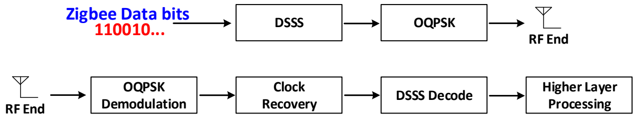


Fig. 2: ZigBee Transmitter and Receiver

B. WiFi Transmitter

Different from the ZigBee transceiving, the WiFi transmitter processes the data from MAC layer using IEEE802.11g standard as given in Fig. 3. Following the channel coding and interleaving, every 6 bits are mapped into one of the 64 Quadrature Amplitude Modulation (QAM) constellation points. Then, every 48 constellation points, together with 4 pilot symbols and 12 null symbols are modulated onto 64 subcarriers to form a frequency Orthogonal Frequency Division Multiplexing (OFDM) symbol. With 312.5 KHz subcarrier space, each OFDM symbol occupies 20 MHz bandwidth. The 64-point Inverse Fast Fourier Transform (IFFT) is then employed to turn each OFDM symbol into a time-domain signal lasting $3.2\mu s$. By cyclic prefixing, a guard $0.8\mu s$ interval, which is the repetition of the time-domain signal end, is added to the beginning, forming a complete WiFi symbol lasting $4\mu s$.

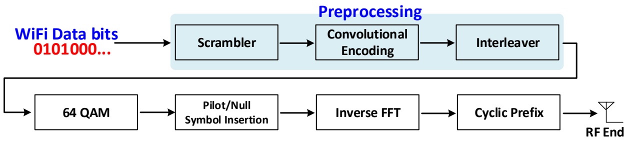


Fig. 3: WiFi Transmitter

C. ZigBee Waveform Emulation

1) Overview and Technical Challenges: To emulate a perfect ZigBee waveform signal is a non-trivial task for WiFi attackers. Since each ZigBee symbol lasts $16\mu s$ whereas each WiFi symbol lasts $4\mu s$, the WiFi attacker needs to create 4 WiFi symbols to emulate one complete ZigBee symbol. Here, we focus on using one WiFi symbol to emulate $1/4$ of the time-domain waveform corresponding to one ZigBee symbol.

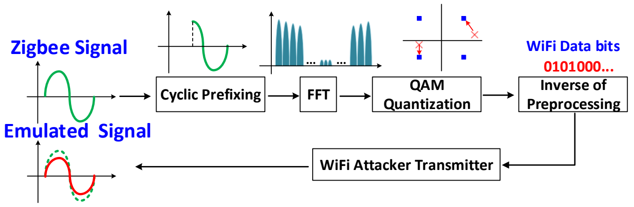


Fig. 4: ZigBee Waveform Emulation

In order to obtain the emulated signal without any change on the WiFi transmitter, the attacker needs to get the original source bit information from the received ZigBee waveform. Hence, motivated by [2], a reverse process is carried out on ZigBee waveform as shown in Fig. 4. We take $1/4$ of the time-domain waveform as a unit waveform to illustrate the emulation process. FFT takes the first $1/4$ of the unit waveform and takes the last $3/4$ to get its frequency information of the subcarriers. From the bandwidth relationship between the ZigBee and WiFi signal, we observe that at most 7 subcarriers (2MHz $\approx$ 7 $\times$ 0.3125MHz) of each WiFi signal can carry the information of the ZigBee waveform, while 64-point FFT operation will get the 64 frequency points of the ZigBee Signal. Thus, the information on 7 constellation points is kept while other information on other points has to be discarded, which becomes one of the main reasons to cause the difference between the original and the emulated ZigBee signal. Due to the different modulation schemes between the ZigBee and WiFi, the chosen frequency points of the ZigBee signal cannot match the original QAM points of the WiFi signal. Therefore, QAM quantization is needed to map the FFT output to QAM constellation points, which intrinsically introduces errors and makes difference further between these two signals. Therefore, the WiFi attacker has to uttermost diminish the difference brought by the FFT and QAM quantization to achieve the goal of controlling the ZigBee device.

2) Choosing Frequency Points after FFT: At the WiFi transmitter, the time-domain waveform $x(n)$ after 64-point IFFT is expressed as,

$$x(n) = \frac{1}{K}\sum_{k=1}^{K} X(k)e^{-j2\pi kn/N}, \quad n = 1, 2, \cdots, N \tag{1}$$

where $X(k)$ is the frequency component corresponding subcarrier $e^{j2\pi kn/N}$. $N = K = 64$. From (1), we see that the waveform in the time domain is actually composed by the K frequency components with the subcarriers in the frequency domain. The weight $X(k)$ represents the importance of the subcarrier $e^{j2\pi kn/N}$ to the waveform. Since only 7 subcarriers can be used to emulate the ZigBee signal, we choose the largest frequency components to decrease the difference between the original and the emulated ZigBee signals.

In practice, the WiFi attacker cannot choose the frequency components for each coming $\frac{1}{4}$ ZigBee signal due to the complexity. Since the central frequency and the bandwidth of the coming ZigBee signal is fixed, the distribution of $X(k), k = 1, 2, \cdots, K$ is similar for each unit waveform. Thus, the WiFi attacker only determines the subcarrier indexes k in which the frequency components are kept. A two-step algorithm is proposed to decide the index, the *coarse estimation* and *detailed estimation*. We describe it based on the example in Table. I, where we list the frequency components of each coming $1/4$ ZigBee signal in each column. Note that we ignore the frequency component with the subcarrier indexes $8-54$. In the coarse estimation, the WiFi attacker highlights all the frequency components above the threshold (set as 3 in the example), marked as red in Table. I. In the detailed estimation, the WiFi attacker determines the 7 subcarrier indexes, at which the most highlighted frequency components locate. In the final, the subcarriers with $1-4$ and $62-64$ indexes are chosen. The frequency components on these subcarrier indexes are sent into the QAM quantization.

TABLE I: Frequency Points of ZigBee Waveform

Index	1	2	3	4	5	6
1	19.8135	14.4096	14.9512	40.0943	19.8135	14.4096
2	14.2990	50.3424	44.0796	27.5399	14.2990	50.3424
3	11.1025	28.8303	23.1920	14.1483	11.1025	28.8303
4	8.3671	12.1972	14.9302	17.9765	8.3671	12.1972
5	5.6639	1.4931	5.5869	2.2252	5.6639	1.4931
6	3.0938	1.6792	3.5464	2.5908	3.0938	1.6792
7	1.0538	2.1977	1.4703	2.8351	1.0538	2.1977
...	...	...	...	...	...	...
55	1.1616	0.1748	2.5695	1.4498	1.1616	0.1748
56	0.8171	1.0029	3.2787	0.9751	0.8171	1.0029
57	0.6807	0.6807	3.0777	0.6807	0.6807	0.6807
58	1.6783	0.7128	4.6410	0.8608	1.6783	0.7128
59	2.6743	2.0764	5.2603	4.1972	2.6743	2.0764
60	2.9140	3.0542	5.9928	2.7222	2.9140	3.0542
61	1.5631	4.4502	14.0955	3.4206	1.5631	4.4502
62	4.3057	7.1549	11.4675	13.7336	4.3057	7.1549
63	39.2439	7.8455	8.4652	22.6196	39.2439	7.8455
64	40.7812	14.1395	22.7630	20.6058	40.7812	14.1395

3) Quantizing Chosen Points: According to the Parseval's theorem related to the FFT/IFFT, the energy of the waveform in the time domain is equaled to that in the frequency domain after Fourier transform. Taking the linear property, we have the following equation for the errors introduced by the QAM quantization on the chosen frequency points,

$$\int_{t=-\frac{T}{2}}^{t=\frac{T}{2}} |\hat{x(t)} - x(t)|^2 dt = T \sum_k |\hat{X(k)} - X(k)|^2 \tag{2}$$

where $x(t)$ denotes the ZigBee time domain signal composed of 7 subcarriers. The emulated ZigBee signal is denoted as $\hat{x(t)}$, and $X(k)$, $\hat{X(k)}$ are their corresponding FFT points.

The difference-energy equation (2)shows that minimizing the signal distortion in the time-domain under energy metric is equivalent to minimizing the total deviation of frequency components after QAM quantization. Therefore, QAM quantization is to choose the closest QAM constellation point in term of Euclidean distance to each of the chosen frequency points. However, the WiFi attacker just knows the 64 QAM structure as follows,

$$X(k) = \alpha \left(X_I(k) + jX_Q(k) \right) \tag{3}$$

where $X_I(k), X_Q(k) \in \{-7, -5, -3, -1, +1, +3, +5, +7\}$ are the real and imagine part of the complex symbol $X(k)$, and α is used to scale the constellation. The attacker has to choose a scalar for and QAM constellation first before quantizing the chosen frequency points. The QAM quantization becomes an optimization problem with the variable α, where the objective is to minimize the total Euclidean distance between the chosen frequency points and the QAM constellation points. Specifically, the optimization problem is formulated as follows,

$$\begin{aligned} \min_{\alpha} \quad & \sum_k \left(\hat{X_I(k)} - \alpha X_I(k) \right)^2 + \left(\hat{X_Q(k)} - \alpha X_Q(k) \right)^2 \\ \text{s.t.} \quad & \alpha \geq 0 \end{aligned} \tag{4}$$

in which $\hat{X_I(k)}$ and $\hat{X_Q(k)}$ are the known real and imaging parts of the chosen frequency point $\hat{X(k)}$. Since the $X_I(k)$ and $X_Q(k)$ depend on the discrete values, the WiFi attacker employs a numerical global research method to obtain the value of the scaler, followed by finding the QAM constellation for each frequency point.

4) Carrier Allocation and Cyclic Prefix: Since the preprocessing in the WiFi transmitter is invertible, the source bits of the emulated signal can be easily obtained given the quantized QAM constellation points. Hence, we directly get into the pilot/null subcarrier insertion step when emulating the ZigBee signals, which is actually a subcarrier allocation process among data, pilot, and the null points. A common subcarrier allocation scheme is to put 48 data points into subcarriers $[-26, -22]$, $[-20, -8]$, $[-6, -1]$, $[1, 6]$, $[8, 20]$, and $[22, 26]$, respectively, and allocate subcarriers -21, -7, 7 and 21 to the pilot points. Since the WiFi signal transmitted in the pilot/null subcarrier cannot be controlled by software, the WiFi attacker has to put the quantized frequency points into the data subcarriers. Because the WiFi attacker knows the central frequency of the ZigBee receiver, it can set its central frequency to achieve the above goal. Taking the ZigBee 17 channel as an example, it works at the central frequency 2435MHz. The WiFi attacker sets its central frequency at 2440MHz, under which the data subcarriers $[-20, -8]$ carry the information of the Zigbee signal. information.

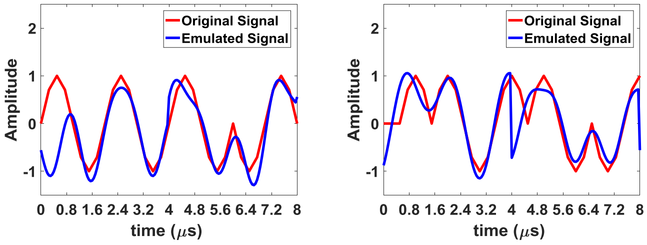
Because the cyclic prefix must be added to the beginning of each WiFi symbol, the beginning parts and the end parts of the emulated signal remains the same whereas the ZigBee signal does not have such a repetition. Nevertheless, the emulated signal can still pass the ZigBee receiver detection and decoding since DSSS demodulation is able to tolerate a certain number of errors.

D. Emulation Attack Simulation

As an initial validation, we simulate the CTC waveform emulation attack on the USRP N210 devices [19].

1) Simulation Process: Our simulation process follows the attacking process: channel listening and emulation attack. We first create ZigBee waveform using a ZigBee transmitter with 2MHz bands and 4MHz sampling rate. Given the assumption that the WiFi attacker synchronizes the ZigBee waveform perfectly, we interpolate the ZigBee waveform with parameter 5 creating 80 points in each WiFi symbol duration. Then, we put the last 64 points into FFT, and choose the frequency points at the location $1-4$ and $62-64$, which are sent into the QAM quantization with an optimized scaler $\alpha = \sqrt{26}$. The preprocessing is ignored and the produced QAM constellation points are sent into 64-point IFFT. We add the last 16 points of the IFFT output to the beginning as the cyclic prefix. A new 80-point emulated ZigBee signal is formed, which is actually a WiFi signal with the sample rate 20MHz and will be sent to the ZigBee receiver.

2) Simulation Result: Fig. 5 plots the In-Phase and Quadrature waveform of both the original and emulated ZigBee signals, respectively. We can see that the WiFi attacker can perfectly emulate each quarter segment of ZigBee waveform using one WiFi symbol except for the first $0.8\mu s$.



(a) In-Phase

(b) Quadrature

Fig. 5: Emulated Signal Comparison

Meanwhile, we test whether the emulated ZigBee waveform can pass the detection and demodulation process at the ZigBee receiver. In the experiment, although $0.8\mu s$ waveform in the emulated ZigBee signal is totally different than the original one, the receiver still decodes the emulated signal, which demonstrates the effectiveness of the emulation attack.

To see whether the emulated ZigBee signal can survive in the noise environment, we conduct the receiving test at the ZigBee receiver. Additive White Gaussian Noise (AWGN) is added to the emulated ZigBee signal. In each signal-to-noise ratio (SNR), we perform 1000 transmissions from the WiFi attacker to the ZigBee receiver. The successful rate is listed in Table. II, which shows that the WiFi attacker can totally control the ZigBee devices by launching our proposed emulation attack in high SNR.

TABLE II: Emulation Attack Performance Under AWGN

SNR	7dB	9dB	11dB	13dB	15dB	17dB
Successful Rate	42.4%	69.2%	87.4%	93.3%	97.2%	100%

V. Defensive Mechanism Design

In the previous section, the WiFi attacker is able to fool the ZigBee receiver to believe that the received signal is from the authentic ZigBee transmitter or the gateway. At ZigBee receiver side, it seems there is no way to differentiate between the signal from the WiFi attacker and that from the ZigBee transmitter. Existing schemes built upon higher-layer protocols are not able to thwart the proposed waveform emulation attack.

A. Defensive Strategy Analysis

Our intuition on defending the CTC waveform emulation attack is to find differences between the ordinary ZigBee signal and the WiFi signal containing the ZigBee packet. Although the emulated waveform is close to the original ZigBee waveform, different transmission schemes must leave enough "footprints", which paves way for detection.

1) Warm-up Solutions: We analyze the possible defensive strategies by scrutinizing the information flow from Fig. 2. At first glance, cyclic prefix sheds light for us. In each emulated ZigBee waveform segment, the beginning and the end segment are the same. If the ZigBee receiver detects the repetition, it could potentially conclude that the suspicious signal comes from the WiFi attacker. However, this methodology is not reliable. In practice, the signal received by the ZigBee device suffers from the AWGN and even fading effect, which results in the situation that the ZigBee device fails to find the above repetition. We also consider using the output of OQPSK demodulation, which is the signal frequency related to the sample rate, for identifying the authentic ZigBee transmitter. However, the sampling rates for both the ZigBee signal and the emulated signal are the same at the ZigBee receiver side, and thus it is infeasible to differentiate the attacker. Last but not least, in the DSSS demodulation, the hard decision is deployed to decode the chip sequence from the chip samples. Since there are intrinsic errors between the ZigBee and the emulated signals, the chip sequence from these two signals must different, which may be a good candidate for detection. Unfortunately, since DSSS demodulation can tolerate a certain number of errors on chip sequences for decoding, both of the emulated signal and ZigBee signal can be decoded as the same ZigBee symbol even if the received chip sequences are different.

2) Constellation Analysis: The QAM quantization motivates us to differentiate the received signal in the view of the constellation. If the signal comes from an actual ZigBee transmitter, it has the QPSK constellation in the time domain; if not, the signal has the 64-QAM constellation in the frequency domain. Without transforming to the frequency domain, the constellation analysis can be easily done in the time domain. Compared to the actual ZigBee signal, the emulated signal has much larger offsets coming from the quantization errors and the FFT process (i.e., losing non-overlapping high-frequency components), both of which serve as the basis for detecting the waveform emulation attack.

To identify the emulated ZigBee signal, we first get complex symbols from the received time-domain waveform. Considering the DSSS decoding in Fig. 2, every 32 float values are collected, which are then determined as binary 0 or 1 chip and mapped into one ZigBee symbol according to the predefined symbol-to-chip spreading relationship in the DSSS process. At the ZigBee transmitter, the output of DSSS is OQPSK modulated, in which we can use the input of the DSSS demodulation to construct a new QPSK constellation diagram. Specifically, we divide those input as odd and even parts, where odd parts are put to the real axis and even parts being put to the imaginary axis. Therefore, the defensive strategy becomes a simplified constellation recognition problem. In particular, we carry out the digital modulation classification [11] to determine whether the newly constructed constellation diagram belongs to a QPSK structure or not.

In what follows, we mainly consider two scenarios for emulation attack detection. In the ideal scenario, the received signal only suffers AWGN at the ZigBee receiver side. In the practical scenario, the frequency/phase offset happens at the received signal due to the complex channel condition.

B. Emulation Attack Detection under Ideal Scenario

Higher-order statistic is a common and easy method used in the digital modulation classification problem, which can efficiently characterize the shape of the distribution of the noisy baseband samples. Given the newly constructed constellation diagram, we focus on the fourth-order cumulant characteristics.

1) Preliminaries: For a complex-value random variable x, its second-order moments are defined in the following two ways based on the placement of conjugation,

$$C_{20} = E[x^2], \quad C_{21} = E[|x|^2] \tag{5}$$

As for the fourth-order moments and cumulants, they can be defined in three different ways,

$$\begin{aligned} C_{40} &= \mathtt{cum}\,(x, x, x, x) \\ C_{41} &= \mathtt{cum}\,(x, x, x, x^*) \\ C_{42} &= \mathtt{cum}\,(x, x, x^*, x^*) \end{aligned} \tag{6}$$

where x^* represents the conjugate the random variable x, and for zero-mean random variables w, x, y, and z,

$$\begin{aligned} \mathtt{cum}(w, x, y, z) = E(wxyz) - E(wx)E(yz) - \\ E(wy)E(xz) - E(wz)E(xy) \end{aligned} \tag{7}$$

2) Sample Estimation: According to [20], we are able to use the collected complex sample $d_i, i = 1, 2, \cdots, D$ output from the Clock Recovery to estimate (5) and (6) as follows,

$$\widetilde{C}_{20} = \frac{1}{D}\sum_{i=1}^{D} d_i^2, \quad \widetilde{C}_{21} = \frac{1}{D}\sum_{i=1}^{D} |d_i|^2 \tag{8}$$

where $\widetilde{\cdot}$ denotes the sample average. Considering the fourth-order cumulant estimation using complex samples, we have,

$$\begin{aligned} \widetilde{C}_{40} &= \frac{1}{D}\sum_{i=1}^{D} d_i^4 - 3\widetilde{C}_{20}^2 \\ \widetilde{C}_{41} &= \frac{1}{D}\sum_{i=1}^{D} d_i^3 d_i^* - 3\widetilde{C}_{20}\widetilde{C}_{21} \\ \widetilde{C}_{42} &= \frac{1}{D}\sum_{i=1}^{D} |d_i^4| - |\widetilde{C}_{20}|^2 - 2\widetilde{C}_{21}^2 \end{aligned} \tag{9}$$

In (8), the sample estimates of the second-order cumulants include the effect of the noise random variable. Thus, a local estimate of its variance has to be obtained and subtracted from $\widetilde{C}_{20}$ and $\widetilde{C}_{21}$. In addition, such a noise effect affects the estimate of the fourth-order cumulants according to (9). However, the constellations are not necessarily normalized after decoding at the ZigBee receiver in practice. To deal with the problem, the fourth-order cumulant estimates are usually normalized as $\widehat{C}_{4q} = \widetilde{C}_{4q}/\widetilde{C}_{21}^2$, where $q = 0, 1, 2$. The final normalized fourth-order cumulant estimates are then compared with the corresponding theoretical cumulants in order to decide the constellation type, which are shown in Table. III .

TABLE III: Theoretical Cumulants for $C_{21} = 1$

Modulation	C_{20}	C_{40}	C_{42}
BPSK	1	-2.0000	-2.0000
QPSK	0	1.0000	-1.0000
PSK(>4)	0	0.0000	-1.0000
4-PAM	1	-1.3600	-1.3600
8-PAM	1	-1.2381	-1.2381
16-PAM	1	-1.2094	-1.2094
16-QAM	0	-0.6800	-0.6800
64-QAM	0	-0.6190	-0.6190
256-QAM	0	-0.6047	-0.6047

3) Defensive Strategy: As shown in Table. III, both C_{40} and C_{42} are used to decide constellation types among PSK, PAM and QAM. Specific to our defensive strategy, since the reconstructed constellation is known to be QPSK modulation, we mainly compare how far the estimated fourth-order cumulants are to the theoretical values by using the received chips.

We first define a Voronoi tessellation [21] of the feature space as $\mathbf{v} \triangleq [|C_{40}|, C_{42}]^T$, where C_{40} and C_{42} are the theoretical values as listed in Table. III. Similarly, our estimated fourth-order cumulants $|\widehat{C}_{40}|$ and $\widehat{C}_{42}$ compose a new vector $\phi = \left[|\widehat{C}_{40}|, \widehat{C}_{42}\right]^T$. The Euclidean distance D_E is used to be the distance measure metric between the Voronoi tessellation $\mathbf{v}$ and our estimated vector ϕ, where $D_E = ||\phi - \mathbf{v}||_{\mathbf{2}}$. We decide whether the received signal is transmitted by the ZigBee transmitter or the WiFi attacker by deploying the hypothesis testing. Specifically, we have,

$$\begin{cases} H_0 : \text{From the ZigBee Transmitter} \\ H_1 : \text{From the WiFi attacker} \end{cases} \tag{10}$$

If the signal comes from the WiFi attacker, the error brought the FFT and QAM quantization puts a negative effect to the decision of the constructed constellation type. Here, we introduce a threshold Q to help us make the decision,

$$D_E^2 \underset{H_0}{\overset{H_1}{\gtrless}} Q \tag{11}$$

We will give the value of Q according to our experiments.

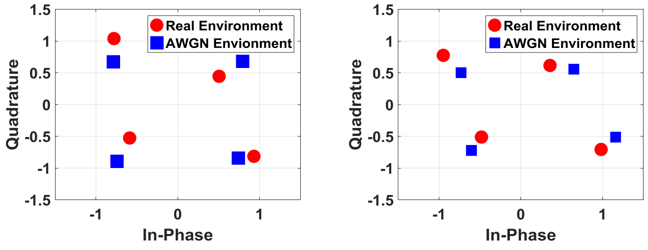
C. Emulation Attack Detection under Real Scenario

We first give an example of the newly constructed constellation in both AWGN and real environments as shown in

Fig. 6. Given the chip samples, we deploy k-means clustering algorithm [22] to help find the constructed constellation points. Denote the chip samples as a set $S_c = \{s_{c1}, s_{c2}, \cdots, s_{cC}\}$, where C is the number of chip samples, k-means clustering algorithms aim at partitioning all the chip samples into 4 sets $\mathbf{S} = \{S_1, S_2, S_3, S_4\}$ so as to minimize the within-cluster sum of squares. Mathematically, its objective is to find:

$$\underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{4} \sum_{S_c \in S_i} ||S_c - \mu_i||^2 \tag{12}$$

where μ_i is the mean of points in S_i.



(a) ZigBee Signal (b) Emulated Signal

Fig. 6: Constellation Diagram Comparison

From Fig. 6, we can see that the new constellation in the real environment has an obvious phase offset compared to that in AWGN environment, which further proves that the existence of significant phase offset in the real environment. Facing the phase offset effect, we reconsider the higher-order statistics deployed in the constellation recognition for the AWGN environment. Denote the frequency offset and the phase offset as Δf and θ, respectively. According to [23], C_{40} is scaled by $e^{j(\Delta f+\theta)}$. In order to avoid the frequency and phase offset, we consider the estimate of the absolute value of C_{40} instead of the Voronoi tessellation in the defensive strategy in the real environment.

VI. Performance Evaluation

We build a prototype to further demonstrate the effectiveness of ZigBee waveform emulation attack and the proposed defensive strategy in both the simulation and real environment respectively. In the end, a thorough complexity analysis is conducted on both the attack and defensive approach.

A. Simulation Settings

We construct two complete communication links including APP layer, MAC layer, and PHY layer from the ZigBee transmitter to the ZigBee receiver and from the WiFi attacker to the ZigBee receiver. We assume that the WiFi attacker has the knowledge about the signal waveform sent by the ZigBee transmitter. The WiFi attacker follows the signal processing as explained in Section IV. Besides, we add another function in the ZigBee receiver to achieve the defensive strategy as described in Section V.

An additive white Gaussian noise (AWGN) with the noise variance σ^2 is transmitted along with the original ZigBee signal and the WiFi emulation signal. respectively. The power of the transmitted signal is normalized and we define the signal-to-noise ratio SNR as $SNR = \frac{1}{\sigma^2}$. For each communication link, the transmission and reception process repeat 100 times. We collect the physical-layer data in the first 50 times to calculate the threshold in (11) at the ZigBee receiver. The rest of the physical-layer data is used in the hypothesis testing show the effectiveness of the proposed defensive strategy.

B. Experimental Results

1) Performance of Waveform Emulation Attack:: We denote the text from ’00000’ to ’00099’ as the input of the APP layer. The ZigBee transmitter sends its waveform directly to the ZigBee receiver. The WiFi attacker emulates its waveform and then transmits the emulated one to the ZigBee receiver. We demonstrate the chip-level performance in Fig.7, which shows the Hamming distance distribution of the received chips. When the signal comes from the ZigBee transmitter, the received chip sequences are exactly the same with the predefined chip sequences as shown in the upper figure in Fig.7. Illustrated in the lower figure, there are 4 to 8 error chips between each chip sequence and the predefined one when the emulated ZigBee signal is received. Since DSSS has the error resilience, the sequences with error chips could be decoded as the correct ZigBee symbols with a feasible threshold. In our simulation, all of the emulated signals are decoded correctly with a feasible threshold of 10. Such observation further testifies that the WiFi attacker could control the ZigBee device by deploying the principle of the DSSS explained in IV-A.

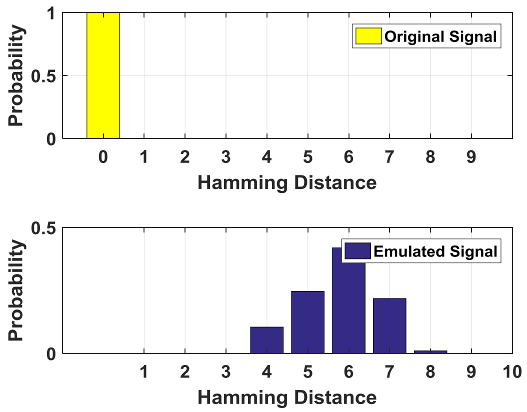
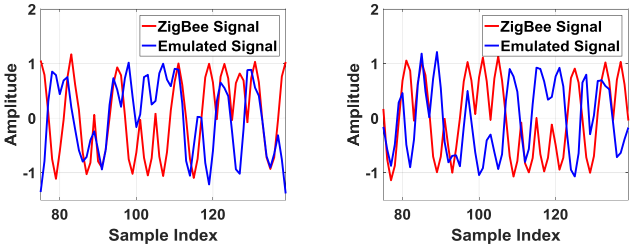


Fig. 7: Hamming Distance Distribution Comparison

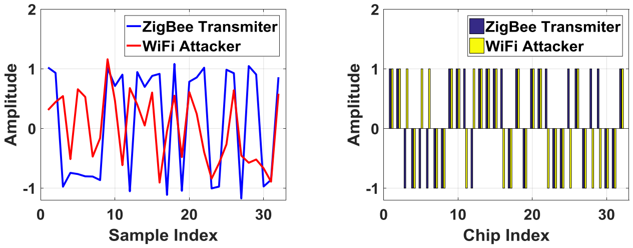
2) Performance of Warmup Strategy: We first show the defensive approach performance of the warmup strategy in Sec.V-A1. For the experiment, we choose high SNR to avoid the noise effect. Fig.8 shows the received In-Phase and Quadrature waveform at SNR = 17dB respectively. We can hardly find the repeated segment from the waveform. Thus, we can hardly identify the emulation attacker by comparing the beginning and the end segment of the received signal.



(a) In-Phase (b) Quadrature

Fig. 8: Waveform Comparison

In Fig.9a, we demonstrate the output of the OQPSK demodulation process, which shows the signal frequency in relation to the sample rate. It is obvious that the trends of these two waveforms are the same, and thus we cannot use the output from the OQPSK demodulation to distinguish the transmitter. In addition, we show the chip sequence performance after hard decision in DSSS demodulation in Fig. 9b. Although the chip sequence performance under the ZigBee and emulated signal cases are totally different, the ZigBee receiver can obtain the same ZigBee symbol. Thus, we cannot distinguish the transmitter or attacker from these chip sequence.

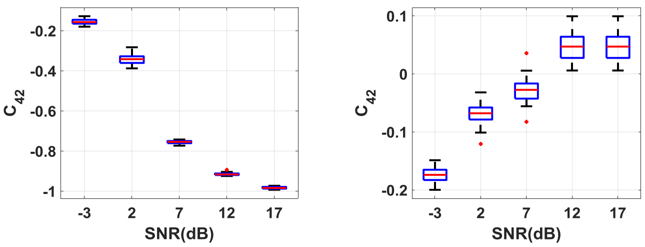


(a) Phase (b) Chip Amplitude

Fig. 9: Received Signal Comparison

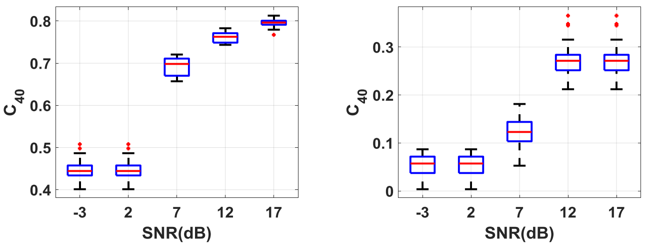
3) Performance of Constellation-based Approach: To demonstrate the effectiveness of our proposed constellation-based defensive strategy, we conduct experiments at different SNRs to evaluate the fourth-order cumulant C_{42} performance of signals from the ZigBee transmitter and the WiFi attacker, respectively.

As shown in Fig. 10, we mainly compare the value of C_{42}, where more approaching to the theoretical value -1 will be categorized as authentic ZigBee transmitters. In Fig.10a, it shows the C_{42} performance of the actual ZigBee signal. With the increase of SNR, the value of C_{42} will be much closer to -1. However, the C_{42} value of emulated signals are far from the theoretical value and keeps on changing to an opposite way. Due to the errors in the QAM quantization and the information lost on the non-overlapped frequency, the newly constructed constellation under the emulated signal intrinsically has an offset to the QPSK constellation. As the SNR becomes lower, the noise with larger variance decreases such offset on the contrary. Therefore, the trends of the C_{42} under the emulated signal and ZigBee signal cases are opposite. Such observation validates the effectiveness of our proposed defensive strategy. The fourth-order cumulant C_{40} performance is also demonstrated in Fig. 11. The calculation methods are the same with the C_{42}. Comparing the value of C_{40} under the ZigBee signal in Fig.11a and the emulated signal case in Fig.11b, the C_{40} value under the ZigBee signal case is more close to the theoretical value 1 than that under the emulated signal. However, the ZigBee receiver cannot distinguish the transmitter using the above trends because it cannot get the C_{42} and/or C_{40} performance of the received signal at different SNRs at once. Therefore, the predetermined threshold decision is needed for WiFi attacker detection.



(a) ZigBee Signal (b) Emulated Signal

Fig. 10: C_{42} Performance



(a) ZigBee Signal (b) Emulated Signal

Fig. 11: C_{40} Performance

4) Effectiveness of Threshold Decision: When receiving a signal, the ZigBee cannot know the transmitter except for calculating the value of $|\widehat{C}_{40}|$ and $\widehat{C}_{42}$. For detection purpose, it needs a threshold to decide whether the signal is from the ZigBee transmitter and the WiFi attacker. Note that we have demonstrated that the packet reception rate is low at the SNR below 7dB when the signal is coming from the WiFi attacker in Table. II. Thus, we reconsider the fourth-order estimation performance at the SNR above 7dB. Instead of the Euclidean distance, we first calculate average Euclidean distance square using the first 50 signal samples under both the ZigBee signal and emulated signal at each SNR, which is listed in the Table. IV. We observe that there is a large gap between the ZigBee signal and emulate signal, which make our decision on the threshold easier. To find out the specific threshold value,

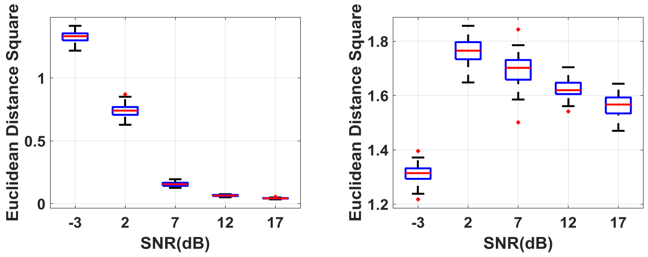
$$\begin{aligned} {D_E}^2 = ||\phi - \mathbf{v}||_2^2 &= (\widehat{C}_{40} - C_{40})^2 + (\widehat{C}_{42} - C_{42})^2 \\ &= (\widehat{C}_{40} - 1)^2 + (\widehat{C}_{42} + 1)^2. \end{aligned}$$

So, we decide the threshold of $\widehat{C}_{42}$ as -0.5 and $\widehat{C}_{40}$ as 0.5. Therefore, the final threshold Q in (11) becomes 0.5.

TABLE IV: Averaged Euclidean Distance Square (${D_E}^2$)

SNR	7dB	12dB	17dB
ZigBee Signal	0.1546	0.0642	0.0421
Emulated Signal	1.7140	1.6238	1.5536

The average of the Euclidean distance square over 100 ZigBee signal samples and 100 emulated signal samples in Fig. 12. We observe that the maximum ${D_E}^2$ is below 0.5 at the SNR above 7dB for the ZigBee signal while the minimum ${D_E}^2$ is above 0.5 for the emulated signal at the corresponding SNR. Since the WiFi attacker can fool the ZigBee devices at the SNR above 7dB, the ZigBee receiver can distinguish the ZigBee signal and the emulated signal effectively by using our proposed defensive strategy while receiving the message.



(a) Tested ZigBee Signal (b) Tested Emulated Signal

Fig. 12: Defense Strategy Performance

C. Experimental Settings and Results

We conduct the experiment using the USRP N210 and the commodity device TI LaunchPad CC26x2R1 [24]. The USRP N210 is equipped with AD and DA converters before the RF front ends and UBX-40 daughter boards operating in the 2.4GHz range as transceivers. Its corresponding software toolkit is GNURadio [25]. The LaunchPad CC26x2R1 is part of the micro-controller unit (MCU) platform supporting the IEEE 802.15.4g protocol. In the experiment shown in Fig.13, we deploy one USRP N210 as the ZigBee transmitter and WiFi attacker alternately. The ZigBee transmitter works on the spectrum centered at 2435MHz with the sample rate 4MHz. Whereas the WiFi attacker operates at the center of 2440MHz with the sample rate 20MHz. The power gain of them is set at 0.75. Because the Zigbee receiver begins to decode the sequence only after getting a zero sequence, we add 10 zero points at the beginning of each emulated packet. The other USPR N210 and the launchpad CC26x2R1 play the role of the ZigBee receiver. Both of them is centered at the 2435MHz. The received power gain of the USRP receiver is set as 0.75. Because the Zigbee receiver begins to decode the sequence only after getting a zero sequence, we add 10 "0" at the beginning of each emulated packet. The distance between the transmitter and receivers ranges from 1m to 8m. During the experiment, there are human activities such as walking. We illustrate the value of received signal strength indication (RSSI) at the launchpad CC26x2R1 under different distances in Table. V in Fig.13. RSSI is an indication of the power level being received by the receive radio after the antenna loss [26].

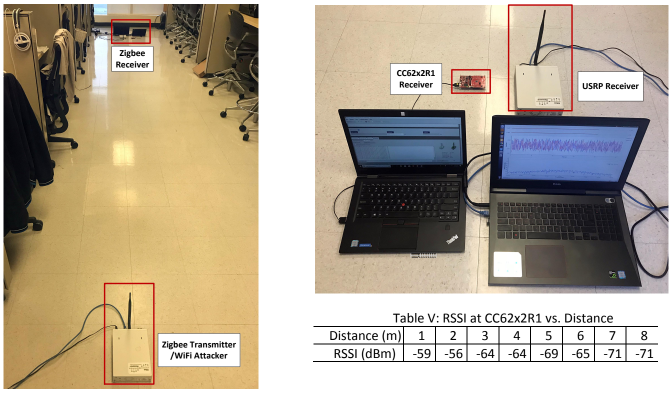
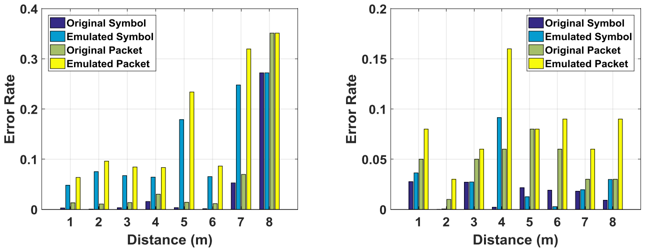


Table V: RSSI at CC62x2R1 vs. Distance

Distance (m)	1	2	3	4	5	6	7	8
RSSI (dBm)	-59	-56	-64	-64	-69	-65	-71	-71

Fig. 13: Experimental Setting

We mainly focus on the performance of the waveform emulation attack in the practical environment. As the same as the simulation, the ZigBee transmitter and the WiFi attacker send the text from '00000' to '00099', respectively, we evaluate the error rates of the packet and symbol at the USRP receiver and CC26x2R1. As shown in Fig.14, the error rates of both the packet and symbol are lower than that of the emulated packet and symbol. This is because the noise and interference in the real scenario enlarge the difference between the emulated and original signal at the ZigBee receiver. Meanwhile, the packet error rate is larger than the symbol error rate because the packet is received correctly only if all the symbols in the packet are exactly received.



(a) Receiver: USRP (b) Receiver: CC26x2R1

Fig. 14: Waveform Emulation Attack Performance

As demonstrated in Fig.14a, the error rates of both the packet and symbol are less than 0.1 for both the emulated and original ZigBee signal when the distance between the transmitter and the USRP receiver is below 5m. When the distance increases, e.g., 7m, the WiFi attacker could not fool the ZigBee device due to the large error rate. At the distance 8m, the USRP receiver cannot decode the original ZigBee signal either. Thus, it is obvious that a WiFi attacker performs a little bit worse than the ZigBee transmitter at the USRP receiver. However, shown in Fig.14b where CC26x2R1 is deployed as the receiver, the error rates of both the emulated packet and symbol are less than 0.1 even if the distance between the

WiFi attacker and the receiver is long, e.g., 8m. Since the commodity ZigBee device has stronger demodulation function than the experimental USRP, we conclude that the proposed waveform emulation attack could effectively fool the ZigBee device even from a long distance.

D. Complexity Analysis

1) Waveform Emulation Attack: The attacking process mainly consists of FFT and QAM quantization. The N-point FFT is done with $\mathcal{O}(N\log(N))$. The coarse estimation after FFT is a binary hard-decision process with $\mathcal{O}(M)$, where M denotes the number of samples. Following with it, we sum up the binary elements in each row and get a final vector, where each element denotes the number of the highlighted signal samples related to the subcarrier index. The detailed estimation is to sort the vector and find the first 7 maximized elements, which has the complexity $\mathcal{O}(n)$, where n is the number of total subcarriers. The QAM quantization includes finding the optimal scalar and mapping the frequency components of the ZigBee Signal to the QAM constellation. Meanwhile, our global search method is based on the mapping process. According to [2], choosing the closest N QAM points in term of total Euclidean distance to each of K FFT points of desired signals is easily done in $\mathcal{O}(K)$.

In general, FFT has a complexity $\mathcal{O}(N\log(N))$. However, N fixed at 64 while others depend on the number of the samples from coming ZigBee waveform. Therefore, the waveform emulation attack can be done easily in $\mathcal{O}(M)$, where M is the number of the coming ZigBee samples.

2) Defensive Approach: The main part of our defensive strategy is to calculate the fourth-order cumulants. According to [20], the fourth-order cumulants estimation can be done in $\mathcal{O}(N)$, where N denotes the complex sample number. Therefore, our proposed defense strategy is easy to be implemented with the order of the sample number.

VII. Conclusion

In this paper, we discovered a new emulation attack built on CTC, where the WiFi device fully controls the ZigBee device directly bypassing the ZigBee gateway. To defend against this attack, we proposed a countermeasure to identify the WiFi attacker by using higher-order statistics to recognize the constellation of the received signal. We perform a thorough evaluation on the USRP platform and the commodity device in both AWGN and real scenario. The experimental results demonstrated the effectiveness of the CTC emulation attack and the defensive strategy.

References

[1] "Iot report how internet of things technology is now reaching mainstream companies and consumers," https://www.businessinsider.com/internet-of-things-report.

[2] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 2–14.

[3] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 245–258.

[4] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali, "Zigfi: Harnessing channel state information for cross-technology communication," in *Proceedings of ACM INFOCOM*, 2018.

[5] X. Zheng, Y. He, and X. Guo, "Stripcomm: Interference-resilient cross-technology communication in coexisting environments," in *IEEE Int. Conf. Comput. Commun.(INFOCOM)*, 2018, pp. 15–19.

[6] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 317–330.

[7] K. Chebrolu and A. Dhekne, "Esense: communication through energy sensing," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 85–96.

[8] X. Zhang and K. G. Shin, "Gap sense: Lightweight coordination of heterogeneous wireless devices," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 3094–3101.

[9] W. Jiang, Z. Yin, S. M. Kim, and T. He, "Transparent cross-technology communication over data traffic," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[10] Z. Yin, W. Jiang, S. M. Kim, and T. He, "C-morse: Cross-technology communication with transparent morse coding," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[11] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed feature-based modulation classification using wireless sensor networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE, 2008, pp. 1–7.

[12] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[13] L. Hong and K. Ho, "Bpsk and qpsk modulation classification with unknown signal level," in *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, vol. 2. IEEE, 2000, pp. 976–980.

[14] ——, "Modulation classification of bpsk and qpsk signals using a two element antenna array receiver," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, vol. 1. IEEE, 2001, pp. 118–122.

[15] C. J. Le Martret and D. Boiteau, "Modulation classification by means of different orders statistical moments," in *MILCOM 97 Proceedings*, vol. 3. IEEE, 1997, pp. 1387–1391.

[16] P. Marchand, C. Le Martret, and J.-L. Lacoume, "Classification of linear modulations by a combination of different orders cyclic cumulants," in *spwhos*. IEEE, 1997, p. 0047.

[17] P. Marchand, J.-L. Lacoume, and C. Le Martret, "Multiple hypothesis modulation classification based on cyclic cumulants of different orders," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 2157–2160.

[18] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[19] "Usrp n210," https://www.ettus.com/product/details/UN210-KIT.

[20] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on communications*, vol. 48, no. 3, pp. 416–429, 2000.

[21] S. Fortune, "Voronoi diagrams and delaunay triangulations," in *Computing in Euclidean geometry*. World Scientific, 1995, pp. 225–265.

[22] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering." in *ICML*, vol. 98. Citeseer, 1998, pp. 91–99.

[23] A. Swami and B. Sadler, "Modulation classification via hierarchical agglomerative cluster analysis," in *Signal Processing Advances in Wireless Communications, First IEEE Signal Processing Workshop on*. IEEE, 1997, pp. 141–144.

[24] "Simplelink cc26x2r1 sdk overview," http://dev.ti.com/tirex/content/simplelink_zigbee_sdk_plugin_1_60_00_14/docs/zigbee_user_guide/html/zigbee/simplelink_cc2652_sdk_overview/simplelink_cc2652_sdk_overview.html.

[25] "Gnu radio," https://www.gnuradio.org/.org.

[26] "Cc2652r simplelink multiprotocol 2.4-ghz wireless mcu," http://www.ti.com/lit/ds/symlink/cc2652r.pdf.