AuthCTC: Defending Against Waveform Emulation Attack in Heterogeneous IoT Environments

Sihan Yu
Clemson University
Clemson, SC, USA
sihany@g.clemson.edu

Xiaonan Zhang
Clemson University
Clemson, SC, USA
xiaonaz@g.clemson.edu

Pei Huang
Clemson University
Clemson, SC, USA
peih@clemson.edu

Linke Guo
Clemson University
Clemson, SC, USA
linkeg@clemson.edu

Long Cheng
Clemson University
Clemson, SC, USA
lcheng2@clemson.edu

Kuangching Wang
Clemson University
Clemson, SC, USA
kwang@clemson.edu

ABSTRACT

Widely deployed IoT devices have raised serious concerns for the spectrum shortage and the cost of multi-protocol gateway deployment. Recent emerging Cross-Technology Communication (CTC) technique can alleviate this issue by enabling direct communication among heterogeneous wireless devices, such as WiFi, Bluetooth, and ZigBee on 2.4 GHz. However, this new paradigm also brings security risks, where an attacker can use CTC to launch wireless attacks against IoT devices. Due to limited computational capability and different wireless protocols being used, many IoT devices are unable to use computationally-intensive cryptographic approaches for security enhancement. Therefore, without proper detection methods, IoT devices cannot distinguish signal sources before executing command signals. In this paper, we first demonstrate a new defined physical layer attack in the CTC scenario, named as waveform emulation attack, where a WiFi device can overhear and emulate the ZigBee waveform to attack ZigBee IoT devices. Then, to defend against this new attack, we propose a physical layer defensive mechanism, named as *AuthCTC*, to verify the legitimacy of CTC signals. Specifically, at the sender side, an authorization code is embedded into the packet preamble by leveraging the dynamically changed cyclic prefix. A WiFi-based detector is used to verify the authorization code at the receiver side. Extensive simulations and experiments using off-the-shelf devices are conducted to demonstrate both the feasibility of the attack and the effectiveness of our defensive mechanism.

CCS CONCEPTS

• **Security and privacy → Authorization**; • **Networks → Mobile and wireless security**.

ASIA CCS '20, October 5–9, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-6750-9/20/10…$15.00
https://doi.org/10.1145/3320269.3384726

KEYWORDS

Cross-Technology Communication; Waveform Emulation Attack; Physical Layer Security

ACM Reference Format:
Sihan Yu, Xiaonan Zhang, Pei Huang, Linke Guo, Long Cheng, and Kuangching Wang. 2020. AuthCTC: Defending Against Waveform Emulation Attack in Heterogeneous IoT Environments. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20), October 5–9, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3320269.3384726

1 INTRODUCTION

The wide deployment of the Internet of Things (IoT) has resulted in serious problems in terms of wireless spectrum scarcity and device coexistence [37]. A large number of end IoT devices, although using different wireless protocols, still interfere with each other in the already-crowded industrial, scientific and medical (ISM) band. To tackle this issue, Cross-Technology Communication (CTC) provides a viable solution, which enables direct communication among devices adopting different wireless protocols including WiFi, Bluetooth, and ZigBee [24]. In contrast to existing indirect methods such as deploying a multi-protocol gateway, CTC is able to reduce the cost of gateway deployment and avoid repeated data transmission with different wireless protocols. However, the development of CTC also brings potential challenges to the security of IoT devices. For example, in a designated CTC application scenario, a ZigBee smart lock is allowed to receive commands (`LOCKING/UNLOCKING`) from an authorized ZigBee gateway and some other WiFi devices (e.g., smartphone or tablet) for enhancing the efficiency of spectrum utilization. Meanwhile, all of these commands have the same content since they perform the same function. Then, it is very hard for the smart lock to differentiate whether or not the command comes from an authorized source. As a result, this new communication paradigm provides opportunities for a WiFi-based attacker to maliciously control a broader range of IoT devices, such as smart locks, smart outlets, and security cameras, all of which are controlling critical functionalities in the future smart home. Therefore, how to differentiate whether the command comes from a valid gateway, a legitimate CTC device, or an illegitimate CTC device becomes a challenging issue. Given that most IoT devices have limited computational capabilities, accomplishing such a task is nearly infeasible.

In this work, we present a new physical-layer attack in the heterogeneous IoT environment with a focus on the CTC between WiFi and ZigBee protocol, named as **Waveform Emulation Attack (WEA)**. Specifically, a WiFi-based attacker is able to eavesdrop on the communication channel between a ZigBee gateway and a ZigBee end device, then, emulates the eavesdropped signal to attack the ZigBee end device. Different to traditional replay attacks, the newly proposed WEA has the following uniquenesses: (1) from the perspective of attackers, the replay attack is launched by homogeneous devices whereas WEA is launched by heterogeneous devices; (2) from the perspective of defenders, traditional defensive schemes intend to prevent replays whereas the WEA defender allows replays but wants to check the legitimacy of the signal source for ensuring the authenticity of signals. Additionally, attacking ZigBee devices with WiFi devices has the following advantages: (1) the attacker has a stronger camouflage ability, where he can disguise himself as a passerby with a commonly-used smartphone; (2) the attacker can launch an attack at a farther distance because of the longer transmission range and stronger penetration capability of WiFi signals. However, in terms of defensive approaches, existing cryptographic methods (e.g., AES-128 [29]) may not work to prevent WEA since most of the cryptographic methods used in wireless protocols are applied in the higher layer, where the objects being processed are hexadecimal symbol sequences. Since both the emulated waveform and the original waveform will be decoded into the same sequence at the physical layer, they have no difference when they are transmitted to the higher layer.

To defend against WEA, we propose a physical layer defensive mechanism, named as *AuthCTC*. Our idea is to embed an authorization code at the sender side, which can be verified at the receiver side with a WiFi-based detector. The embedded authorization code will dynamically change over time, making attackers unable to predict or re-use the overheard code for attacking purposes. The main contributions of this paper are as follows:

- We define and demonstrate the existence of a new physical layer attack in the heterogeneous IoT environment, where current security mechanisms are unable to thwart.
- We propose a novel detection method to prevent the WEA. Without using higher-layer cryptographic approaches, the defensive mechanism is implemented in the physical layer to achieve high efficiency and low cost.
- Different to existing schemes [49] that simply regard CTC signal as malicious attacks, our work prevents illegitimate CTC without sacrificing the benefits of legitimate CTC.
- We perform extensive experiments on both the USRP platform and a self-designed prototype to validate the existence of the WEA and further demonstrate the effectiveness of the defense strategy.

The rest of this paper is organized as follows: Section 2 introduces the motivation of proposing *AuthCTC*. Section 3 and Section 4 describe the process of WEA and *AuthCTC* respectively. Section 5 demonstrates the attacking performance and the effectiveness of defensive mechanism through extensive experiments. Section 6 discusses related works about CTC and physical layer security. Section 7 concludes the paper.

2 MOTIVATION

2.1 New Challenges Brought by CTC

Cross-Technology Communication (CTC) [5, 7, 9, 10, 13–15, 18–20, 22, 24, 25, 40, 45, 53, 54] enables two heterogeneous devices to communicate directly without the help of a multi-protocol gateway, which enhances the interoperation of different wireless protocols and the efficiency of spectrum utilization. However, CTC also brings new security risks to IoT devices, e.g., an end device will face potential attacks from many different types of wireless devices. If CTC is allowed and serves as a normal operation, the content being transmitted tends to be relatively simple, because we cannot expect a device knows well about the security mechanisms deployed on other heterogeneous devices. For example, if we use a smartphone to directly control a ZigBee smart lock, then, we should tell the `LOCKING/UNLOCKING` command to the smartphone and allow the replay, because the smartphone does not know the secret key of ZigBee cryptosystem and cannot generate a new encrypted command by itself. Since an illegitimate CTC user can overhear and replay the command as well, ZigBee end devices may receive signals from ZigBee gateway, legitimate CTC users, and illegitimate CTC users. Therefore, how to differentiate the legitimacy of received signals becomes a challenging problem.

2.2 Existing Security Mechanisms in IoT

In recent years, many security mechanisms (e.g. [16, 27, 35, 39]) adopt machine learning methods to achieve anomaly detection. Aegis [35] observes different user activities and usage patterns and builds a contextual model to differentiate malicious and benign behavior. Hafeez *et al.* [16] propose a traffic morphing technique that shapes network traffic thus making the adversary more difficult to identify IoT devices and their activities. HomeSnitch [27] presents a framework for classifying IoT device communication by semantic behaviors (e.g. heartbeat, motion detection), which can help identify previously unseen devices and behaviors. However, this kind of classification method (i.e. classify the user behavior into malicious and benign) is invalid when faced with WEA, because WEA completely mimics the user's behavior, the classifier will inevitably classify the mimic communication traffic into benign behavior.

Other classic security mechanisms adopted in ZigBee IoT devices are mainly the cryptographic methods [36], such as AES-128 [29]. However, the use of cryptographic methods has two main disadvantages: (1) It is hard to differentiate the source of received packets when they have the same content. Although some techniques (e.g., digital signature) can achieve the sender verification, they depend on the uniqueness of timestamp or sequence number to prevent replay attack, i.e., any signed ciphertext can only be used for once, and the second time usage will be regarded as a replay attack. However, the CTC scenario typically allows replay, where the signed ciphertext can be replayed by various legitimate CTC devices so that the non-repudiation property of the digital signature is lost. As a result, this kind of method cannot be used to differentiate the source of packets. (2) It limits the wider adoption of CTC. Specifically, some cryptographic methods have the property of defending against replay attacks, so the encrypted data field of packets will be different for each time. As shown in Fig. 1,

these two packets are the smart bulb's "TURNING ON" command we overheard from a ZigBee gateway in different time slots. They have the same function but with different data fields. However, WiFi devices cannot generate this kind of encrypted packets, because they do not know the secret key used in ZigBee systems. Secret keys and their generating mechanism are the secrets of ZigBee device manufacturer, they directly determine which devices can join their network so that devices produced by other manufacturers may be excluded from their network. However, WiFi devices (e.g. smartphone) can be owned by anyone. If the ZigBee device manufacturers allow arbitrary WiFi devices to know their secret key and join their network, their security attributes will no longer exist. Since the WiFi device cannot generate a new encrypted CTC packet or replay an old packet (due to the usage of cryptographic method), CTC completely loses its functionality.

MAC Payload												
FCF	Destination	Source	Radius	Seq Num	Relay Count	Relay Index	Security Control Field	Frame Counter	Extended Source	Key Seq Num	Data	MIC
0806	a34a	0000	1e	11	00	00	28	0d7f2100	b36efc0e006f0d00	00	e8da3d6f30e37fae312d331ca39e	ca8d8fdf

MAC Payload												
FCF	Destination	Source	Radius	Seq Num	Relay Count	Relay Index	Security Control Field	Frame Counter	Extended Source	Key Seq Num	Data	MIC
0806	a34a	0000	1e	1b	00	00	28	147f2100	b36efc0e006f0d00	00	a0e9700ad742d372b53e52cff539	4e7cc368

FCF: Frame Control Field MIC: Message Integrity Code

Figure 1: Encrypted ZigBee Packets

As a result, it is highly desirable that a security mechanism can identify the source of packets even if they have the identical content so that we can verify the legitimacy of received packets instead of completely excluding them with the cryptographic methods. As one of the few works in this field, Zhang *et al.* [49] propose a physical layer detection method that uses constellation higher-order statistic analysis to differentiate whether the received signal is sent from a ZigBee device or a WiFi device. However, this work regards any WiFi devices as potential attackers, which limits the usage of CTC. Taking a step further, we ask, when CTC is allowed, how to differentiate the legitimacy of received signals?

2.3 Adversarial Model

We decide to construct an attacking scenario according to the aforementioned security risks. In our adversarial model, the attacker uses a WiFi-based device with 64-QAM modulation. With our proposed mechanism in Sec. 3, the attacker can eavesdrop on, decode (i.e. demodulate ZigBee waveform into ZigBee symbol) and emulate arbitrary ZigBee packets. The eavesdropped ZigBee commands can be either in plaintext or ciphertext, the attacker cannot and doesn't need to decrypt the ciphertext. The ZigBee system allows commands to be replayed. Noise and other interference signals (e.g., WiFi) are allowed because the attacker can differentiate whether the eavesdropped signal is a ZigBee signal or not.

In order to acquire a valid command, the attacker can analyze the functionalities of eavesdropped packets based on the user's activities, the traffic patterns and the information of packets' headers (e.g. the source and destination addresses, which are in plaintext even if a certain cryptographic method is used). For a multi-device scenario, there are also some machine learning methods which can be used to analyze the functionalities of packets, such as [1, 48].

2.4 Design Intuition of AuthCTC

To defend against WEA, we deploy a WiFi-based device in proximity of IoT devices as the detector. If a legitimate CTC device wants to send a packet, it can embed an authorization code in the preamble of the packet. The detector can detect the authorization code at the receiver side. If the received signal has the correct authorization code, it is regarded as a legitimate CTC signal; If the received signal doesn't have the correct authorization code, it is regarded as an illegitimate CTC signal and the detector will give an alarm to tell the user that your IoT devices are being controlled by attackers. In our scenario, we do not consider the "insider attack" (i.e. a legitimate CTC device may be controlled by an adversary), as long as the CTC device is legitimate, all the packets send by it are regarded as legitimate packets. In our design, the authorization code will change over time, so that an attacker cannot replay the previous code or predict the next code.

3 WAVEFORM EMULATION ATTACK

3.1 Overview

We first present the overview of WEA, then provide a detailed description of each component. The process of WEA mainly contains three steps: starting point detection, decoding, and signal emulation, in which the first two steps correspond to the eavesdropping process and the last step corresponds to the attacking process. Fig. 2 demonstrates the workflow of the standard WiFi physical layer [3], where we mainly highlight the modules that are relevant to our design and omit some irrelevant modules. We overwrite these functions with the above three modules. In particular, the starting point detection is to find the beginning of an eavesdropped ZigBee signal, the decoding is to convert received waveform to symbol sequence, and the signal emulation is to send ZigBee waveform with WiFi hardware.

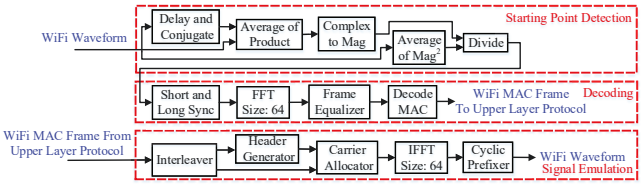
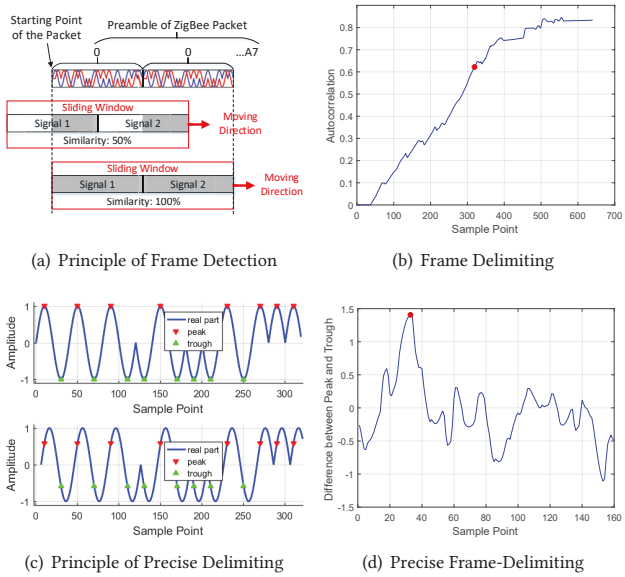


Figure 2: Workflow of WiFi Physical Layer

3.2 Starting Point Detection

3.2.1 Existing Detection Methods. Since we use WiFi devices (which have a broader bandwidth than ZigBee) to eavesdrop on ZigBee signals, the obtained signals may include ZigBee signals, WiFi signals, and noise. Hence, the first step is the frame detection, i.e., determining whether a received signal is a ZigBee signal and where is the starting point. Using WiFi devices to delimit ZigBee frames is a new challenge. Existing works [3, 6, 14] exploit the repetitive pattern of preamble (i.e., the ZigBee preamble is "00000000A7") to delimit frames. Fig. 3(a) shows their principle of frame delimiting, which measures the similarity of two waveforms (i.e. signal 1 and signal 2), also known as autocorrelation coefficient (ACC).



(a) Principle of Frame Detection (b) Frame Delimiting

(c) Principle of Precise Delimiting (d) Precise Frame-Delimiting

Figure 3: Signal Discrimination

The sliding window moves from the left to right, when it finds the similarity (i.e. the area of gray parts in Fig. 3(a)) of two waveforms is higher than a threshold, it regards the current position as the starting point of the packet. However, our empirical study reveals that the result of this method is not accurate enough, because the increase in similarity is a slow process and it is very difficult to set a generic threshold. We conduct an experiment to find the starting point of a frame based on this method, as shown in Fig. 3(b). It can be seen that the ACC increases slowly and finally reaches to about 0.83. However, in fact, the frame begins at the red point, whose ACC is 0.62. Thus, it is very difficult to set a threshold to find the starting point based on this kind of figure.

3.2.2 Precise Delimiting Scheme. To address the above problem, we design a precise frame delimiting scheme. The first step is using a rough estimation method to calculate the ACC. Once the ACC is greater than 0.5, we then perform the second step, i.e., precise delimiting. In the first step, we calculate the ACC as follows,

$$ACC_i = \frac{\sum_{k=0}^{\lfloor 319/n \rfloor} S_{i+nk} S^*_{i+320+nk}}{\sum_{k=0}^{\lfloor 319/n \rfloor} S_{i+320+nk} S^*_{i+320+nk}} \quad (1)$$

where S_i is the i-th sample point, S^* is the complex conjugate of S, 320 (or 0-319) is the length of an emulated symbol, and n is the step size. By adjusting n, we can make a tradeoff between computational cost and detection precision. The function of the denominator is to normalize the numerator.

In the second step, we compare the current waveform, i.e., signal 1 in Fig. 3(a), with ZigBee symbol "0". If they match very well, we conclude that the current position of the sliding window is the beginning of a ZigBee frame. The upper part of Fig. 3(c) shows the principle of precise delimiting. The real part of the ZigBee symbol "0" has 8 peaks and 8 troughs, the difference between peaks and troughs is expected to be large. However, if the sliding window shifts to the left or right, the difference will not be so significant, as shown in the lower part of Fig. 3(c). Based on this observation, we calculate the starting point of the frame as follows,

$$i = \arg\max_i \left\{ w_i | w_i = \Re \left(\sum_{k=1}^{8} (S_{P_k+i} - S_{T_k+i}) \right), i \in [0, 79] \right\} \quad (2)$$

where P_k is the k-th element of set P, T_k is the k-th element of set T. P and T are the indexes of peak points and trough points, each of which contains 8 elements. The range of i is adjustable. Here we set $i \in [0, 79]$ because we find that when ACC is greater than 0.5, the starting point often appears within the following 40 sample points. So we set a relatively wide range that includes 80 sample points to ensure the starting point appears in the range. Fig. 3(d) shows an experiment result of our method, in which we successfully find the red point as the exact starting point of the ZigBee frame.

3.3 Decoding

Before launching a WEA, decoding is necessary. The eavesdropped signal may include multiple ZigBee packets that have different sources and destinations. The attacker has to pick the useful packets and replay them to launch an attack. The main principle of decoding is to compare the received signal with 16 standard ZigBee symbols to find which is the most similar one. In order to reduce the cost of comparison, we extract the key feature of the 16 standard ZigBee symbols to form a table. Fig. 4 shows how to extract the feature of a ZigBee symbol. When a ZigBee symbol is received by a WiFi device, it will be truncated into 4 pieces, where each piece includes 80 sample points. Then, the first 1/5 of each piece is removed because it is considered as the cyclic prefix (CP). Finally, each piece will be converted to the frequency domain by FFT. Each WiFi symbol corresponds to 7 frequency-domain points so that a ZigBee symbol can be simplified to 28 frequency-domain points. Through the above processes, we can derive the frequency-domain data of 16 ZigBee symbols and then compare them with that of the received signal. We use the Pearson Correlation Coefficient (PCC) [42] to measure the similarity of two sequences. Given two sequences $X = \{x_1, x_2, ..., x_K\}$ and $Y = \{y_1, y_2, ..., y_K\}$, PCC is defined as:

$$P_{XY} = \frac{\sum_{k=1}^{K} (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{K} (x_k - \bar{x})^2} \sqrt{\sum_{k=1}^{K} (y_k - \bar{y})^2}} \quad (3)$$

where $\bar{x} = (\sum_{k=1}^{K} x_k)/K$ is the sample mean and analogous for $\bar{y}$. The P_{XY} ranges from -1 to 1, where "1" or "−1" indicates that X and Y have a perfect positive/negative linear correlation. "0" indicates that there is no linear correlation. The reason for using PCC for comparison is that it is invariant under separate changes in location and scale of the two variables. That is, when we transform X and Y to $a + bX$ and $c + dY$, where $b, d > 0$, the P_{XY} does not change.

Decoding the ZigBee signal by frequency-domain data comparison can significantly reduce the computation complexity. We only need to compare the similarity of two sequences whose lengths are 28. In addition, it can make better use of the WiFi hardware resources, because the FFT function is executed by the hardware and has a very fast operation speed. On the contrary, if we decode the ZigBee signal with raw sample points, we have to compare the similarity of two sequences whose lengths are 320 whereas

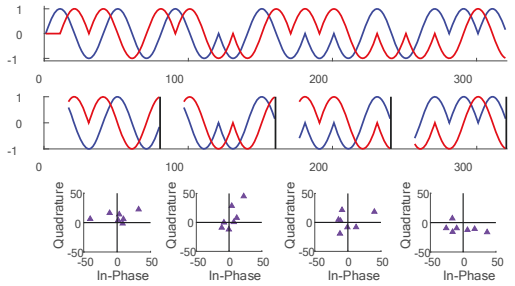


Figure 4: Decoding Process

the hardware resource of FFT is idle. This will lead to that some modules are too busy to finish the task whereas other modules have nothing to do.

We also do some experiments to verify whether the 16 ZigBee symbols are distinguishable with 28 frequency-domain data. We use both PCC and Dynamic Time Wrapping (DTW) to evaluate the distinguishability. DTW is good at measuring similarity between two sequences which may vary in speed or have different lengths. For DTW, a small value means two sequences are very similar to each other. We only take the imaginary parts of the 28 data into consideration, because we find that the real parts of ZigBee symbols are indistinguishable. Specifically, the real parts of the following ZigBee symbol pairs {(0, 8), (1, 9), (2, A), (3, B), (4, C), (5, D), (6, E), (7, F)} are the same.

Fig. 5 shows the experiment results, in which white grids indicate that two symbols are very similar to each other whereas black grids mean two symbols are very different from each other. Both of these two results demonstrate that ZigBee symbols can be distinguished by 28 frequency-domain data.

symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	-0.1756	-0.1533	-0.4894	0.1321	0.0486	-0.1533	-0.2802	-0.3104	-0.1382	0.1771	0.2503	0.296	-0.3624	0.1771	-0.2428
1	-0.1756	1	-0.2802	0.0798	0.0486	-0.1455	-0.4894	0.0798	-0.1382	-0.0798	-0.2428	0.3146	-0.3624	0.108	0.2503	0.3146
2	-0.1533	-0.2802	1	-0.1756	-0.1533	-0.4894	0.1321	0.0486	0.1771	-0.2428	-0.3104	-0.1382	0.1771	0.2503	0.296	-0.3624
3	-0.4894	0.0798	-0.1756	1	-0.2802	0.0798	0.0486	-0.1455	0.2503	0.3146	-0.1382	-0.0798	-0.2428	0.3146	-0.3624	0.108
4	0.1321	0.0486	-0.1533	-0.2802	1	-0.1756	-0.1533	-0.4894	0.296	-0.3624	0.1771	-0.2428	-0.3104	-0.1382	0.1771	0.2503
5	0.0486	-0.1455	-0.4894	0.0798	-0.1756	1	-0.2802	0.0798	-0.3624	0.108	0.2503	0.3146	-0.1382	-0.0798	-0.2428	0.3146
6	-0.1533	-0.4894	0.1321	0.0486	-0.1533	-0.2802	1	-0.1756	0.1771	0.2503	0.296	-0.3624	0.1771	-0.2428	-0.3104	-0.1382
7	-0.2802	0.0798	0.0486	-0.1455	-0.4894	0.0798	-0.1756	1	-0.2428	0.3146	-0.3624	0.108	0.2503	0.3146	-0.1382	-0.0798
8	-0.3104	-0.1382	0.1771	0.2503	0.296	-0.3624	0.1771	-0.2428	1	-0.1756	-0.1533	-0.4894	0.1321	0.0486	-0.1533	-0.2802
9	-0.1382	-0.0798	-0.2428	0.3146	-0.3624	0.108	0.2503	0.3146	-0.1756	1	-0.2802	0.0798	0.0486	-0.1455	-0.4894	0.0798
A	0.1771	-0.2428	-0.3104	-0.1382	0.1771	0.2503	0.296	-0.3624	-0.1533	-0.2802	1	-0.1756	-0.1533	-0.4894	0.1321	0.0486
B	0.2503	0.3146	-0.1382	-0.0798	-0.2428	0.3146	-0.3624	0.108	-0.4894	0.0798	-0.1756	1	-0.2802	0.0798	0.0486	-0.1455
C	0.296	-0.3624	0.1771	-0.2428	-0.3104	-0.1382	0.1771	0.2503	0.1321	0.0486	-0.1533	-0.2802	1	-0.1756	-0.1533	-0.4894
D	-0.3624	0.108	0.2503	0.3146	-0.1382	-0.0798	-0.2428	0.3146	0.0486	-0.1455	-0.4894	0.0798	-0.1756	1	-0.2802	0.0798
E	0.1771	0.2503	0.296	-0.3624	0.1771	-0.2428	-0.3104	-0.1382	-0.1533	-0.4894	0.1321	0.0486	-0.1533	-0.2802	1	-0.1756
F	-0.2428	0.3146	-0.3624	0.108	0.2503	0.3146	-0.1382	-0.0798	-0.2802	0.0798	0.0486	-0.1455	-0.4894	0.0798	-0.1756	1

(a) Distinguishability Verification With PCC

symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	60	36	62	64	64	34	54	64	60	52	46	68	62	58	56
1	60	0	58	30	54	52	54	30	58	60	48	46	58	56	50	48
2	36	58	0	54	48	60	64	60	56	54	74	54	50	54	74	68
3	62	30	54	0	64	42	58	44	54	48	46	50	56	46	62	60
4	64	54	48	64	0	66	40	58	70	60	56	60	70	48	52	50
5	64	52	60	42	66	0	54	36	58	54	54	50	56	56	62	56
6	34	54	64	58	40	54	0	58	54	48	68	56	62	58	68	54
7	54	30	60	44	58	36	58	0	52	46	52	52	52	52	52	54
8	64	58	56	54	70	58	54	52	0	58	36	62	64	56	46	70
9	60	60	54	48	60	54	48	46	58	0	68	30	60	52	64	46
A	52	48	74	46	56	54	68	52	36	68	0	60	36	56	62	58
B	46	46	54	50	60	50	56	52	62	30	60	0	66	26	54	50
C	68	58	50	56	70	56	62	52	64	60	36	66	0	58	46	60
D	62	56	54	46	48	56	58	52	56	52	56	26	58	0	64	32
E	58	50	74	62	52	62	68	52	46	64	62	54	46	64	0	60
F	56	48	68	60	50	56	54	54	70	46	58	50	60	32	60	0

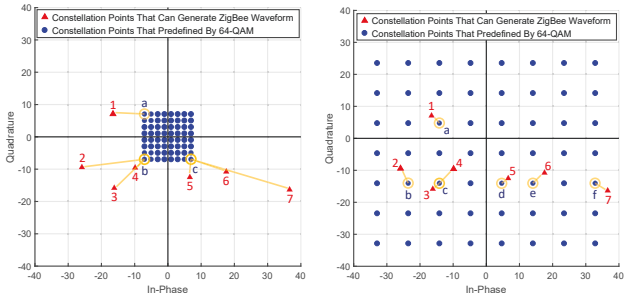
(b) Distinguishability Verification With DTW

Figure 5: Distinguishability Verification

So far, the attacker can understand and capture the useful content from the eavesdropped ZigBee packets to launch an attack.

3.4 Signal Emulation

In WEA, the key challenge is how to emulate the ZigBee signal as perfectly as possible. The WiFi 64-QAM cannot generate ZigBee waveform in a perfect way because its constellation points are discrete and predefined, for which the quantization process will use the nearest predefined point to approximately represent the point we want to use. According to Parseval's theorem [41], minimizing the signal distortion in the time domain is equivalent to minimizing the total deviation of constellation points in the frequency domain. In addition, we find that the size of the 64-QAM constellation diagram also influences the quantized result. As shown in Fig. 6(a), if the size is not suitable, we can only use a fraction of constellation points (a to c) to represent the points we want to use (point 1 to 7). However, if the size is suitable, as shown in Fig. 6(b), we can maximize the use of constellation points (a to f) and minimize the total error.



(a) Quantized with Unfit Constellation Diagram (b) Quantized with Suitable-sized Constellation Diagram

Figure 6: The Quantized FFT Points

Suppose α is the size factor of 64-QAM constellation diagram, i.e., the real parts and imaginary parts of constellation points are chosen from $\mathbf{A} = \{-7\alpha, -5\alpha, -3\alpha, -\alpha, \alpha, 3\alpha, 5\alpha, 7\alpha\}$, then (4) and (5) describe how to minimize the quantization error.

$$E(\alpha) = \sum_{i=1}^{P} (\Re(p_i) - \Re(N(\alpha, p_i)))^2 + (\Im(p_i) - \Im(N(\alpha, p_i)))^2 \quad (4)$$

$$\alpha = \arg\min_{\alpha} E(\alpha) \quad (5)$$

in which (4) is the sum of squared error, (5) is to get the size factor that can minimize (4), $N(\alpha, p_i)$ is to find the nearest constellation point (p_j) to point p_i from 64 predefined constellation points whose size factor is α (i.e. (6)), $\Re$ and $\Im$ is to get the real part and imaginary part, and P is the total number of points.

$$p_j = \arg\min_{p_j} \sqrt{(\Re(p_j) - \Re(p_i))^2 + (\Im(p_j) - \Im(p_i))^2} \quad (6)$$
$$\Re(p_j), \Im(p_j) \in \mathbf{A}$$

Since the second derivative of (4) (i.e., (7), in which $N'(\alpha, p_i)$ is a number instead of an expression) is greater than 0, (4) is a convex function and has the global minimum. Thus, the first derivative of (4) (i.e., (8)) is a monotonic increasing function. We use binary

search to find its zero-crossing point, which is equivalent to find the minimum value of the primitive function.

$$E''(\alpha) = 2\sum_{i=1}^{P} \Re^2(N'(\alpha, p_i)) + \Im^2(N'(\alpha, p_i)) \tag{7}$$

$$\begin{aligned} E'(\alpha) = -2\sum_{i=1}^{P} &[\Re(p_i) - \Re(N(\alpha, p_i))] \times \Re(N'(\alpha, p_i)) \\ &+[\Im(p_i) - \Im(N(\alpha, p_i))] \times \Im(N'(\alpha, p_i)) \end{aligned} \tag{8}$$

Algorithm 1 shows the process of finding α. The most time-consuming part is line 3, which calculates the total error of P points. Its time complexity is $O(P)$. The while-loop executes $\log_2 \left\lceil \frac{high-low}{a} \right\rceil$ times, whose time complexity is $O(-\log_2 a)$, in which a is the result accuracy we want to achieve (e.g., $a = 10^{-3}$ if round to 3 decimal places). Thus, the overall time complexity is $O(-P\log_2 a)$.

Algorithm 1: Finding α

Input: result accuracy $a = 10^{-3}$
FFT points p_i
lower limit of α *low*
upper limit of α *high*
Output: optimum size of constellation diagram α

1 **while** $high - low > a$ **do**
2 $\quad \alpha = \frac{high+low}{2}$;
3 $\quad E'(\alpha) = -2\sum_{i=1}^{P} [\Re(p_i) - \Re(N(\alpha, p_i))] \times \Re(N'(\alpha, p_i)) + [\Im(p_i) - \Im(N(\alpha, p_i))] \times \Im(N'(\alpha, p_i))$;
4 $\quad$ **if** $E'(\alpha) > 0$ **then**
5 $\quad\quad high = \alpha$;
6 $\quad$ **else if** $E'(\alpha) < 0$ **then**
7 $\quad\quad low = \alpha$;
8 $\quad$ **else**
9 $\quad\quad$ return α;
10 $\quad$ **end**
11 **end**
12 $\alpha = round(\frac{high+low}{2}, -\lg a)$;
13 return α;

After acquiring the suitable-sized constellation diagram, the attacker can use it to generate the desired waveform.

4 WEA DETECTION

To defeat WEA, we propose a physical layer defensive mechanism named as *AuthCTC*, which uses a one-way hash chain [21] as authorization codes. These codes are only known by both the legitimate CTC device and the detector. Each time, an authorization code is embedded into the preamble of a CTC packet and sent by the legitimate CTC device. If the detector finds that the received authorization code is correct, the packet will be regarded as a legitimate CTC packet.

For the scenario that there are more than one legitimate CTC devices, each of them should possess a unique hash chain. The detector should possess all chains and work as a central node.

4.1 Authorization Code Generation

To generate a chain of hash values, the detector selects a random number n_γ and recursively computes $n_i = h(n_{i+1}||\text{ID})$, $\forall i \in [0, \gamma - 1]$, where $h(\cdot)$ denotes the cryptographic one-way hash function such as SHA-1, and ID denotes the identification number (e.g., hardware address) of a legitimate CTC device. Next, the detector sends n_γ to the legitimate CTC device in a secure way (e.g., input it manually by the user. This step is feasible because the user only needs to input a seed, just like the process of Bluetooth pairing). Finally, the legitimate CTC device recursively computes $\{n_1, ..., n_{\gamma-1}\}$ in the same way and uses n_i as the authorization code of the i-th transmission. Because the order of hash value generation and hash value usage are different, even the attacker can overhear the current hash value, he/she cannot derive the next available hash value.

As mentioned above, the user needs to reenter a new seed periodically. The value of γ depends on how long the period is. Suppose a user uses 50 authorization codes per day (e.g. he/she turns on/off a ZigBee bulb 25 times), meanwhile, he/she wants to reenter a new seed each year (which is not very often), then, the γ should be $50 \times 365 = 18250$. Suppose the length of an authorization code is 40 bits, then, it will take up $18250 \times 40 = 730Kb$ storage space. A larger γ can reduce the user's unnecessary trouble whereas a smaller γ can enhance the security since the content of the hash chain will be changed before being cracked.

For the problem of synchronization, i.e. how can the communication pair knows which one authorization code is using currently, we can just reserve a few bits to represent the sequence number of the authorization code when embedding them into the packet so that the detector can find the corresponding value of the authorization code according to the sequence number and compare it with the actually received value.

4.2 Authorization Code Encoding

Some papers [51, 52] adopt special modulation schemes to embed authorization codes at the sender side and detect them at the receiver side. However, in our scenario, due to the usage of commercial WiFi devices, the modulation scheme cannot be changed. Jin *et al.* [21] propose a method to prevent unauthorized dynamic spectrum access, they embed spectrum permits into WiFi symbols by changing the lengths of CPs. Similar work can also be found in [46], which embed authorization codes into ZigBee preambles to detect unauthorized devices in the IoT environment. Motivated by the above works, we decide to embed authorization codes into ZigBee preambles by dynamically changing the CP lengths according to the content of the authorization codes. Specifically, a ZigBee preamble "`00000000A7`" includes 10 ZigBee symbols, each ZigBee symbol is emulated by 4 WiFi symbols. Thus, a ZigBee preamble includes 40 WiFi CPs.

Here, we first give an example with specific numbers, then formulate our approach. In normal circumstances, the CP is composed of 16 sample points in the IEEE 802.11g. We define the CP length as a variable value that can be chosen from set A={10, 12, 14, 16, 18, 20, 22}. Then, two CPs that have a sum of 32 form a pair so that we can get 4 CP pairs including (10, 22), (12, 20), (14, 18), (16, 16). In particular, the objective here is to ensure that in the macroscopic view, the CP's length doesn't change, so that the packet length doesn't

change as well. Finally, these 4 CP pairs are mapped to 00, 01, 10, 11 four bit-pairs. When we want to embed an authorization code into the preamble, we can divide the bit sequence of authorization code into bit-pairs and then map them to CP pairs. Fig. 7 shows 4 WiFi symbol pairs with different CP pairs. In each symbol pair, the part with the same color represents the same symbol.

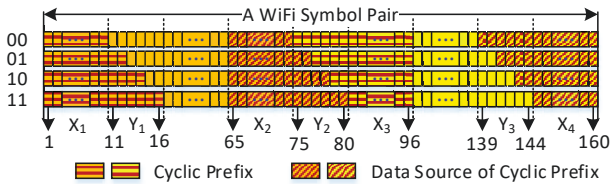


Figure 7: Authorization Code Encoding/Decoding

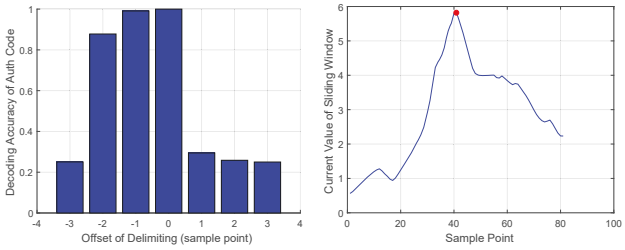
A key issue may be: whether the CP length can be modified arbitrarily on commercial devices. In fact, Schulz *et al.* [32–34] have developed a smartphone APP to implement the Software-Defined Radio (SDR) function on smartphones. It supports various kinds of WiFi devices, which can be found in [31]. By using this APP, we can achieve the goal of adjusting CP length on commercial WiFi devices.

4.3 Authorization Code Decoding

Since both the detector and the CTC user are WiFi-based devices, the detector can receive the CTC signal without any modification in the preamble and header of the WiFi packet. But the detector still has to know where is the starting point of the emulated ZigBee signal, as discussed in Sec. 4.3.1.

4.3.1 High-Precision Frame Delimiting. When receiving emulated signals, the detector also faces the problem of frame delimiting. This frame delimiting is different from what we discussed in Sec. 3.2. The previous one is used on the user side (e.g., smartphone) to detect ZigBee signals, this one is used on the detector side to verify emulated signals. They have two main differences: (1) The ZigBee waveform is predefined so that we can use the received waveform to compare with the predefined waveform. However, the emulated waveform is indeterminate due to the existence of cyclic prefix, especially in this work, where the cyclic prefix is variable for embedding authorization code. (2) The frame delimiting on the detector side has a higher precision requirement than that on the user side. On the user side, even if the delimiting has an offset of 5 sample points, it will not impact the decoding results obviously. However, on the detector side, an offset of 1-2 sample points will lead to a large difference in the decoding results of the authorization code. Fig. 8(a) shows the consequence of delimiting offset. If the delimiting has an offset of 1 sample point, the decoding accuracy of authorization code drops from 100% to 29%. Therefore, we need to design a high-precision frame delimiting method to delimit the beginning of emulated ZigBee frames. The detailed process is described as follows.

When a signal is detected, i.e. the amplitude of the received waveform exceeds a determined threshold, the frame delimiting algorithm is activated. We use a sliding window (length: 160 sample points) to check whether the current position is the starting point of



(a) Decoding Accuracy (b) Frame Delimiting on Detector

Figure 8: Precision Delimiting

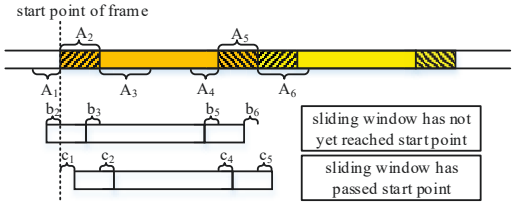


Figure 9: Principle of Sliding Window

the frame. The sliding window should be placed at the position that prior to the header of the received packet and then searches towards the trailer of the received packet. Each time the sliding window moves to a new position, we use (9) to calculate the window's value, which represents the probability that the current position is the starting point of a frame. After getting a series of values (e.g. 80 values, we suppose the starting point will appear among the nearby 80 sample points when a signal is detected), we choose the index of the maximum value as the starting point of the frame.

$$w = \frac{\sum_{k=-l_1}^{0} |S_k - S_{k+64}| + \sum_{k=97}^{97+l_2} |S_k - S_{k+64}|}{\sum_{k=1}^{10} |S_k - S_{k+64}| + \sum_{k=81}^{96} |S_k - S_{k+64}|} \tag{9}$$

In (9), $l_1 \in [1, 48]$ and $l_2 \in [1, 42]$ are adjustable values for trading off between computational costs and delimiting accuracy. A larger l_1 or l_2 means we take more sample points into consideration when calculating the value of sliding window, which will result in a more accurate result as well as more computational costs. A negative subscript means the sample point is at the tail of the last symbol pair, while a positive subscript greater than 160 means the sample point is at the head of the next symbol pair. Since the waveforms represented by the numerator are different from each other whereas the waveforms represented by the denominator are very similar to each other, the value of (9) should be very large. However, if the sliding window has not yet reached or has passed the starting point, the numerator will get smaller whereas the denominator will get larger. As a result, the fraction will get smaller. Therefore, only the starting point has the maximum value.

We use Fig. 9 as an example to explain the principle of (9). Because of $A_2 = A_5$, $A_1 \neq A_4$, $A_3 \neq A_6$, the value of $\frac{|A_1 - A_4| + |A_3 - A_6|}{|A_2 - A_5|}$ (this expression has the similar form with (9)) should be very large.

However, if the sliding window has not yet reached the starting point, the denominator will get larger due to $b_2 \neq b_5$, the numerator will get smaller due to $b_3 = b_6$. As a result, the fraction will get smaller. Similarly, if the sliding window has passed the starting point, the denominator will get larger due to $c_2 \neq c_5$, the numerator will get smaller due to $c_1 = c_4$. As a result, the fraction will also get smaller. Therefore, only when the sliding window exactly arrives at the starting point of the frame, (9) has the maximum value. Fig. 8(b) shows an experiment result of the variation of sliding window values with different window positions. The red point denotes the starting point of the frame, which demonstrates the high precision of our method.

4.3.2 Authorization Code Extraction. After finding the starting point, every 160 sample points are grouped into a symbol pair. Then, we extract the authorization code by checking the CP length of each symbol pair. The distinguishability of four cases lie in Y_1, Y_2 and Y_3 (as shown in Fig. 7), we define the following equations to distinguish the four cases,

$$\text{case 00:} \quad \sum_{k=75}^{80} |D_k^+| / \sum_{k=75}^{80} |D_k^-| \tag{10}$$

$$\text{case 01:} \left(\sum_{k=75}^{76} |D_k^-| + \sum_{k=77}^{80} |D_k^+| \right) / \left(\sum_{k=75}^{76} |D_k^+| + \sum_{k=77}^{80} |D_k^-| \right) \tag{11}$$

$$\text{case 10:} \left(\sum_{k=75}^{78} |D_k^-| + \sum_{k=79}^{80} |D_k^+| \right) / \left(\sum_{k=75}^{78} |D_k^+| + \sum_{k=79}^{80} |D_k^-| \right) \tag{12}$$

$$\text{case 11:} \quad \sum_{k=75}^{80} |D_k^-| / \sum_{k=75}^{80} |D_k^+| \tag{13}$$

in which $D_k^+ = S_k - S_{k+64}$, $D_k^- = S_k - S_{k-64}$, S_k denotes the k-th sample point of a symbol pair. For each received symbol pair, we calculate the value of these four equations. Which one has the minimum value among these four equations, the embedded bits of authorization code are just the corresponding case. From (10)–(13), we can find that the values of sample points in the numerator are almost identical to each other, so their difference will be close to zero. However, the value of the denominator is relatively large, so the value of the whole equation will be close to 0 if the received symbol pair fits the case. In this way, we achieve the purpose of authorization code decoding.

4.4 Detection Scheme Analysis

The previous subsection introduces the encoding/decoding process with a set of given numbers in the parameter setting. In this subsection, we extend the above analysis into a general case and further discuss the pros and cons of the parameter selection.

4.4.1 Formulation. The number of available CP pairs are adjustable in the general case. For example, we can define 8 cases of symbol pairs which with CP lengths {(9,23), (10,22),..., (16,16)} to denote 8 authorization codes 000-111. In this case, the CP lengths are chosen from set A={9, 10, ..., 23}, which has 15 available values with an interval of 1.

Suppose we define C cases of symbol pairs, in which the CP length are chosen from set A={$L_{cp}-(C-1)\Delta$, $L_{cp}-(C-2)\Delta$, ..., $L_{cp}+(C-1)\Delta$}, the delimiting process can be expressed as follows,

$$i = \arg\max_i \{w_i | i \in [0, 79]\} \tag{14}$$

$$w_i = \frac{\sum_{k=-l_1+i}^{i} |S_k - S_{k+L_f}| + \sum_{k=L_s+L_{cp}+1+i}^{L_s+L_{cp}+1+l_2+i} |S_k - S_{k+L_f}|}{\sum_{k=1+i}^{L_{cp}-(C-1)\Delta+i} |S_k - S_{k+L_f}| + \sum_{k=L_s+1+i}^{L_s+L_{cp}+i} |S_k - S_{k+L_f}|} \tag{15}$$

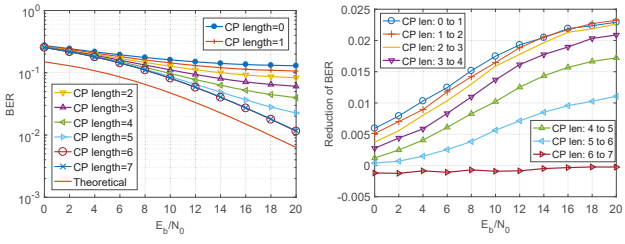
in which $L_{cp} = 16$ is the normal CP length in IEEE 802.11g, Δ is the interval of elements in set A. $L_s = 80$ is the length of WiFi symbol, $L_f = 64$ is the size of FFT operation, $l_1 \in [1, L_s - 2L_{cp}]$, $l_2 \in [1, L_s - L_{cp} - (C-1)\Delta]$. For decoding, we have C equations to estimate which case it is. The c-th equation is defined as (16):

$$\frac{\sum_{k=L_s-(C-1)\Delta+1}^{L_s+(c-C)\Delta} |S_k - S_{k-L_f}| + \sum_{k=L_s+(c-C)\Delta+1}^{L_s} |S_k - S_{k+L_f}|}{\sum_{k=L_s-(C-1)\Delta+1}^{L_s+(c-C)\Delta} |S_k - S_{k+L_f}| + \sum_{k=L_s+(c-C)\Delta+1}^{L_s} |S_k - S_{k-L_f}|} \tag{16}$$

4.4.2 Parameter Selection. The value of C and Δ cannot be too large, otherwise, they will lead to insufficient CP length and cause severe inter-symbol interference and a high BER. On the other hand, the CP cannot be too long, which will slow down the data rate. Thus, the CP length is usually designed as two to four times the root-mean-square of delay spread. We use the following equation [12] to evaluate the BER of 64-QAM OFDM system in Rayleigh fading channel,

$$P_b = \frac{\alpha_M}{2} \left(1 - \sqrt{\frac{0.5\beta_M \gamma_b}{1 + 0.5\beta_M \gamma_b}} \right) \tag{17}$$

in which $\alpha_M = \frac{4}{\log_2 M}$, $\beta_M = \frac{3\log_2 M}{M-1}$, and $\gamma_b = E_b/N_0$ is the SNR per bit. We also conduct a simulation in frequency selective fading channel with 7 taps to measure the appropriate range of CP length. Fig. 10(a) shows the change of BER with different CP lengths and SNR. The BER decreases with the increase of CP length, but when the CP length reaches 6, the BER no longer decreases. Thus, the two curves (CP length=6 and 7) are overlapped with each other. Fig. 10(b) shows a more explicit decrease of BER when the CP length changes. If the value is large, the increase of CP length has a significant effect on decreasing BER. It can be found that when CP length increases from 6 to 7, the decrease of BER is nearly 0, which means this process is meaningless in reducing BER.



(a) Variation Tendency of BER (b) Reduction of BER w/ CP changes

Figure 10: The Impact of CP Length on BER

This simulation demonstrates that 6 can be an appropriate threshold value, as long as the CP length is not less than 6, it will not affect the BER of 64-QAM OFDM system.

In addition, we also study whether the dynamic CP length will cause chip errors when a ZigBee device decodes the emulated signal. We set $C = 2, 4, 8, 16$, where different C values mean the number of available CP pairs and the range of CP lengths are different. Experiment results show that the average chip error rates are: 7.3/32, 5.3/32, 8.7/32, 10.4/32, respectively, indicating $C = 4$ has the lowest chip error rate. Note that as long as the number of wrong chips is smaller than the fault-tolerant threshold of DSSS, the symbol can be decoded correctly.

4.4.3 Overhead Discussion. In the aspect of overhead, the most noticeable problem may be whether the encoding and decoding processes can be done in real-time. Actually, they do not need to be real-time. First, these two processes are performed on WiFi devices (i.e. smartphone and detector) instead of IoT end devices. WiFi devices usually have sufficient capability to complete the above calculation task. Second, the authorization code is only embedded into the preamble of a ZigBee packet instead of the whole packet, so the sender doesn't have to wait for the whole packet to be constructed. Once the previous ZigBee command has been sent, the sender can begin to prepare the preamble of the next command. When the next command needs to be sent, the sender can concatenate the preamble and the payload (i.e. the next command), then send it.

5 EXPERIMENT AND EVALUATION

In this section, we thoroughly investigate the feasibility of WEA, the performance of the attacking process, as well as the effectiveness of the defensive mechanism.

5.1 Feasibility of WEA

We conduct an experiment with a commercial off-the-shelf Osram smart bulb [2] to demonstrate the feasibility of WEA. As shown in Fig. 11(a), we use a WiFi-based USRP to turn on the ZigBee bulb (the "`TURNING ON`" command has been overheard in advance) and repeat this experiment in different positions, as given in Fig. 11(d), where S is the location of the sender, R1-R5 are the locations of the receiver.

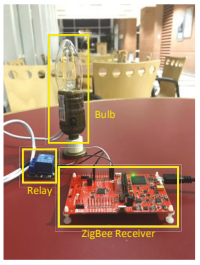
We find that the emulated signal has a significant advantage in the attacking range. In all non-line-of-sight (NLOS) cases (R1, R2, R3, R5), the emulated signal can turn on the light whereas the ZigBee signal cannot. This is because the signal power of ZigBee devices is usually lower than that of WiFi devices. In the line-of-sight (LOS) case (R4), both the emulated signal and the ZigBee signal can turn on the light. Fig. 11(e) shows the symbol error rate (SER) and packet error rate (PER) at each location, it can be seen that the SER and PER of ZigBee signal are significantly higher than that of the emulated signal.

We also implement the WEA on LG Nexus 5 and do the indoor and outdoor experiments. Fig. 11(b) shows the structure of our equipment. It consists of a ZigBee launchPad CC26x2R, a relay, and a 110V light bulb. When the "`TURNING ON`" command is detected by the launchpad, it triggers a high level to the I/O output D100, which enables the relay to turn on the bulb. The smartphone works in the central frequency 2412MHz whereas CC26x2R centered on 2407MHz. For the indoor experiment, we measure the SER and PER under different distances. The results are shown in Table 1. For the outdoor experiment, the smartphone can even turn on the light bulb at a distance of 100m (as shown in Fig. 11(c)). Both of the experiment results indicate that WiFi has a better performance in launching WEA.

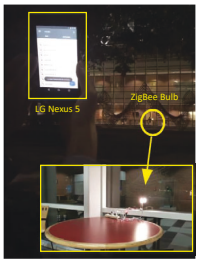
Table 1: SER and PER of WEA by Smartphone

Distance	5m	10m	15m	20m
SER(WiFi)	0.55%	0.4%	0.52%	1.23%
PER(WiFi)	0.75%	1.8%	4.1%	4.8%
SER(ZigBee)	0.51%	0.44%	1.34%	2.31%
PER(ZigBee)	1.1%	1.7%	6%	15.2%

(a) WEA on Smart Bulb

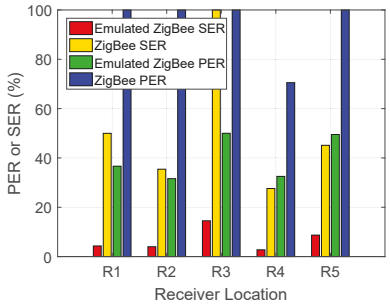


(b) Prototype Demonstration



(c) Outdoor Experiment

(d) Building Map



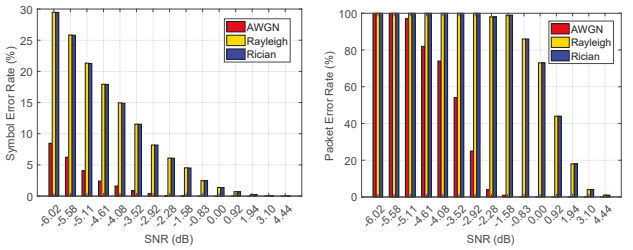
(e) Performance Comparison

Figure 11: WEA Feasibility Demonstration

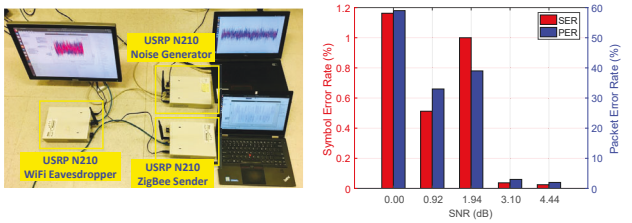
5.2 WEA Performance

In this subsection, we carry out two experiments, where the first one evaluates whether the attacker can decode the eavesdropped signal correctly, and the second one evaluates whether the attacker can emulate the eavesdropped signal accurately. We conduct both simulation experiments and field experiments. Simulation experiments are based on GNU Radio and field experiments are based on USRP.

5.2.1 Signal Eavesdropping. We focus on the impact of different environmental factors, such as channel model, distance, transmission power, and SNR. Each time the ZigBee device sends 100 packets, each of which includes 64 symbols. Meanwhile, we measure the SER and PER at the eavesdropper side.



(a) SER (b) PER



(c) Field Experiment Scenario (d) Error Rate Vs. SNR

Figure 12: Eavesdropping Performance

For the simulation, we simulate different channel models in GNU radio, in which the eavesdropper moves at a speed of 1m/s to simulate the walking process in frequency selective fading model (Rayleigh and Rician). Fig. 12(a) and Fig. 12(b) show the SER and PER results respectively. From Fig. 12(a), we can find that the SER decreases steadily with the increase of SNR. With the same SNR, the AWGN channel model usually has a lower SER. From Fig. 12(b), we can find that the PER also decreases with the increase of SNR, and the decrease occurred in frequency selective fading channel is later than that in the AWGN channel. This is due to the ZigBee device has a relatively low transmission rate so that it is easier to be affected by the movement.

For the field experiment, we evaluate three factors' impact on decoding accuracy, including the distance, transmission power, and SNR. We use three USRPs as the sender, receiver and noise generator respectively, as given in Fig. 12(c). We carry out the experiment by changing the distance between the sender and receiver from 1m to 10m and adjusting the transmission power from -20dBm to -70dBm. All ZigBee signal can be decoded correctly, which demonstrates the accuracy and effectiveness of the delimiting and decoding mechanisms proposed in Sec. 3.2 and 3.3. In Fig. 12(d), we show the variation of communication effect with different SNR. When the SNR is greater than 3.1dB, the SER and PER remain at '0' or very small. When the SNR is smaller than 1.94dB, SER and PER begin to increase significantly. The above experiments demonstrate the accuracy of the proposed eavesdropping process.

5.2.2 Attacking Performance. We carry out similar experiments and evaluate the same factors to demonstrate the attacking performance. Specifically, we let attacker send 100 emulated packets, each of which contains 60 symbols, and measure the SER and PER at the victim side.

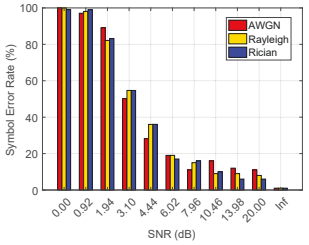
For the simulation, attacker moves at a speed of 1m/s in frequency selective fading model. From Fig. 13(a) and Fig. 13(b), it can be seen that the SER and PER decrease with the increase of SNR. Different to Fig. 12(a) and Fig. 12(b), the decrease of these three models are synchronized, which means the emulated signal has a good performance in frequency selective fading channel, because it is a WiFi-based signal.

For the field experiments, we evaluate the three factors as mentioned in Sec. 5.2.1. We use a TI CC26X2R1 launchpad as the receiver, two USRPs as the sender and noise generator respectively, as given in Fig. 13(c). We vary the distance from 1m to 10m and find that the variation of SER and PER are irregular but the absolute values of them are not very large, as shown in Fig.13(d). This is because 10m is totally in the coverage of signal. Within this range, errors are mostly caused by multi-path effect and random noise. Then, we adjust the transmission power and test the attacking performance with different RSSI (-20dBm∼-90dBm). Fig. 13(e) shows that from -20dBm to -80dBm, the SER and PER do not change obviously, but they increase dramatically at -90dBm, which indicates that at the limit of communication capability, there exists significant performance degradation. Finally, we test the attacking performance with different SNRs. In Fig. 13(f), the SER and PER decrease with the increase of SNR, although some fluctuations may exist due to the random noise. Besides, we find that if PER is significantly greater than the corresponding SER, then, the wrong symbols are often scattered in different packets. If PER is similar to the corresponding SER, then, wrong symbols are more concentrated.

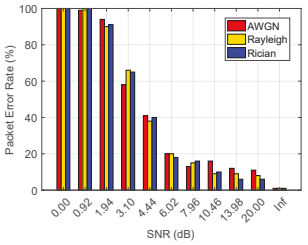
5.3 AuthCTC Performance

In this subsection, we evaluate the defensive performance of *AuthCTC*. As a defense mechanism, if the detector can extract the authorization code accurately from the received packet, it can decide whether the packet is legitimate accurately. Thus, we let the legitimate CTC device send 10000 emulated packets with specified authorization code embedded in their preambles. Then, we measure how many authorization codes can be decoded correctly at the detector side.

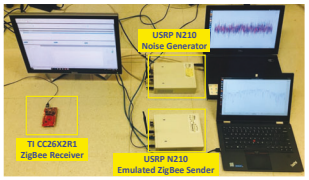
For the simulation, we measure the decoding accuracy with different Δ and C. Fig. 14(a) shows the result of decoding accuracy under different Δ and SNR. The decoding accuracy increases with the increase of SNR. When the SNR is larger than 15dB, the decoding accuracy is higher than 90%. Besides, the decoding accuracy also increases with the increase of Δ, because when the interval of



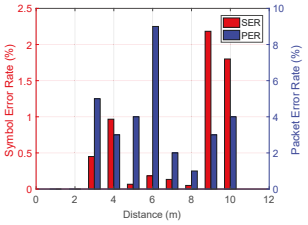
(a) Simulation Results of SER



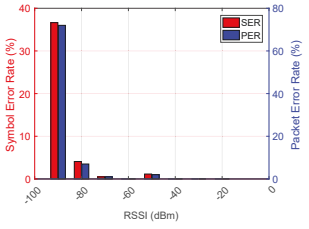
(b) Simulation Results of PER



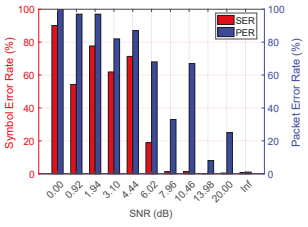
(c) Experiment Scenario



(d) Error Rate Vs. Distances



(e) Error Rate Vs. Signal Strength



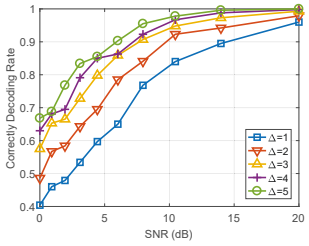
(f) Error Rate Vs. SNR

Figure 13: Attacking Performance

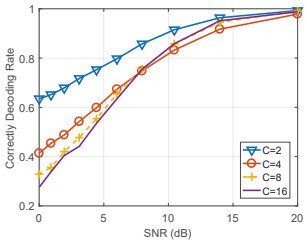
available CP lengths is widened, the distinguishability of different CP lengths is improved. Therefore, the decoding error rate will decrease. Fig. 14(b) shows the decoding accuracy with different C and SNR. In the low SNR part, a small C often has a higher decoding accuracy. This is because when the number of available CP lengths is small, the cases that may cause errors are limited, so the decoding accuracy is higher. However, when the SNR is large, regardless of how many available CP lengths are used, the decoding errors are unlikely to happen. So the C value will not affect the decoding accuracy obviously. From another point of view, the C value should be as large as possible, because when the number of available symbol pairs is large, a symbol pair can represent more bits of the authorization code so that the total length of authorization code can be longer, which will provide a higher level of security, as shown in Fig. 14(d).

For the field experiments, we also measure the impact of Δ and C on decoding accuracy. Fig. 14(e) and Fig. 14(f) show the results, which are similar to the simulation results although there are some fluctuations which are caused by multi-path effect and random noise in the practical environment.

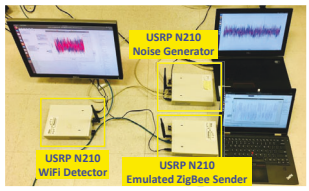
The above experiments demonstrate that our authorization code decoding mechanism (includes the high-precision delimiting method) is valid and accurate. In practical applications, we can set the threshold to 80%, i.e., as long as 80% of the received authorization codes are correct, we can regard the signal as legitimate CTC signal.



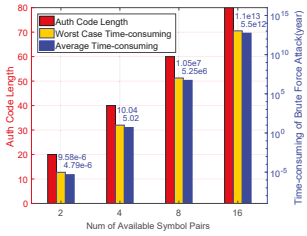
(a) Simulation Results of Decoding Accuracy with different Δ



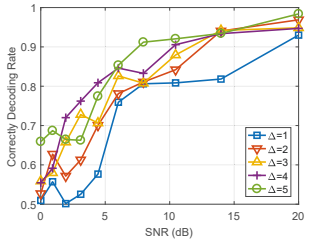
(b) Simulation Results of Decoding Accuracy with different C



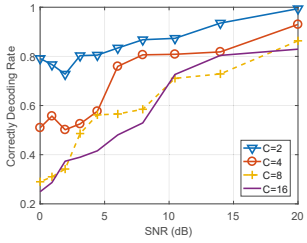
(c) Experiment Scenario



(d) Security Levels



(e) Field Experiment Results of Decoding Accuracy with different Δ



(f) Field Experiment Results of Decoding Accuracy with different C

Figure 14: WEA Detection and *AuthCTC* Performance

6 RELATED WORK

6.1 Cross-Technology Communication

Most of existing CTC schemes focus on improving throughput and shortening the communication delay, which can be classified into two categories: packet level CTC and physical level CTC.

For the packet level CTC, it uses packet level information (e.g., packet duration [5, 53], beacon interval [22], energy pattern [9, 20, 45], energy level [10, 15, 54]) as the minimal unit to construct special pattern that can be detected by other technologies [18, 40]. Esense [5] proposes a WiFi to ZigBee CTC technology by sensing the WiFi packet length at the ZigBee side, in which the packet length is specified and can be distinguished from noise. FreeBee [22] proposes a CTC technology among WiFi, ZigBee and Bluetooth by modulating data into WiFi beacons and shifting the transmission timings of them. C-Morse [45] proposes a WiFi to ZigBee CTC technology by constructing a series of Morse-code-like long and short WiFi packets that can be demodulated at the ZigBee side. B^2W^2 [10] proposes a Bluetooth to WiFi CTC technology by modulating the energy level of Bluetooth packets and demodulated through WiFi CSI at the receiver side. Packet level CTC usually has a low network throughput and large transmission delay.

The recent advances in physical level CTC [7, 13, 19, 24, 25] establish direct physical layer communication via software-based signal emulation. WEBee [24] proposes a WiFi to ZigBee CTC technology by constructing the payload of a WiFi frame elaborately so that the waveform of payload resembles that of ZigBee signals. BlueBee [19] proposes a Bluetooth to ZigBee CTC technology by exploring the opportunities in the signal phase shifts. TwinBee [7] improves the reliability of WiFi to ZigBee CTC by recovering intrinsic errors of signal emulation. LongBee [25] extends the transmission range of WiFi to ZigBee CTC by concentrating the TX power and improving RX sensitivity. WIDE [13] proposes a novel WiFi to ZigBee CTC technology by digital emulation, i.e., decodes symbols by phase shift instead of waveform. Physical level CTC can achieve the maximum transmission rate but also face asymmetric link issues [40], i.e., signal emulation is only applicable for higher-end transmitter to lower-end receiver scenario. This is because powerful radios support sophisticated modulations that can offer higher degrees of freedom in waveform control but not vice versa [18].

6.2 RF Fingerprinting

RF fingerprinting is also used to identify the transmitter. Most radio fingerprinting methods identify a device by considering various physical-layer classification approaches. According to [4], RF features are broadly classified into: (1) channel-specific ones, which characterize the wireless channel, e.g., channel impulse response; (2) transmitter-specific ones, which are independent of the channel, e.g., signal encoding. Channel-specific features have been successfully adopted in robust location distinction [26, 28] by uniquely identifying the channel between the transmitter and the receiver. Transmitter-specific features refer to the RF features that are related to the signal itself [4, 11, 30]. For example, the authors in [4] measure differentiating artifacts of individual wireless frames to achieve accurate NIC identification. In [11], the fingerprinting technique considers the unique features in the radio turn-on transients that appear at the beginning of each transmission. Joint time-frequency Gaborand Gabor-Wigner Transform features are considered in [30] as an approach to extract greater device discriminating information. Besides the above software-based radio fingerprinting methods, the identity can also built on the properties of hardware. However, it is difficult to apply the radio fingerprint techniques on commercial off-the-shelf devices since their unmodifiable features.

6.3 PHY Security Schemes in Heterogeneous Environment

Our identified waveform emulation attack is related to the security schemes using physical layer approaches. The most relevant works are either keyless or key-based approaches [17].

For keyless approaches, typical techniques are to improve the secrecy by making eavesdropper's SINR lower than the receiver's SINR, which include exploiting channel coding [23], channel adaptation [47] and artificial noise injection [43], etc. These approaches focus on avoiding passive eavesdropping instead of recognizing active attacks.

The key-based approach provides secrecy by extracting random keys from the channel of legitimate parties and manipulate them at higher layers. Typical techniques to extract secret key include exploiting channel variation caused by fading [44], relays [38, 50] and subcarriers [8], etc. However, key-based approaches are used among homogeneous devices instead of heterogeneous devices.

To the best of our knowledge, there is no security mechanism that can effectively detect attacks among heterogeneous devices (e.g., WEA), especially in an IoT environment that allows the cross-technology communication.

7 CONCLUSION

In this paper, we propose and implement a novel attack in the cross-technology communication (CTC) environment, named as Waveform Emulation Attack (WEA), where a WiFi device can eavesdrop on the ZigBee communication channel, and then emulate ZigBee signals to control target devices. Then, we develop a physical layer defensive mechanism to defend against this kind of attack. At the sender side, the legitimate CTC device can embed an authorization code into the preamble of the packet by changing the CP length dynamically. At the receiver side, a detector is used to verify the authorization code. Since the authorization code is changing over time, an illegitimate device cannot know the next available authorization code so that its packet cannot pass the verification. Through this mechanism, end ZigBee devices can verify the legitimacy of CTC signals. Experiment results demonstrate that the WEA is feasible and our defense mechanism can defeat it effectively.

ACKNOWLEDGMENTS

We would like to thank Dr. Selcuk Uluagac and other anonymous reviewers for their helpful feedback. The work of L. Guo is partially supported by National Science Foundation (NSF) CNS-1947065, ECCS-1949639, and IIS-1949640. The work of K.C. Wang is partially supported by National Science Foundation (NSF) CNS-1643020.

REFERENCES

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and A Selcuk Uluagac. 2018. Peek-a-Boo: I see your smart home activities, even encrypted! *arXiv preprint arXiv:1808.02741* (2018).

[2] Amazon. [n.d.]. Osram Lightify Smart LED Bulb. https://www.amazon.com/Sylvania-Osram-Lightify-Daylight-Smart/dp/B01NAIV40U

[3] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. 2013. An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio. In *Proceedings of the second workshop on Software radio implementation forum*. ACM, 9–16.

[4] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 116–127.

[5] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: communication through energy sensing. In *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 85–96.

[6] Ruirong Chen and Wei Gao. 2019. Enabling Cross-Technology Coexistence for Extremely Weak Wireless Devices. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 253–261.

[7] Yongrui Chen, Zhijun Li, and Tian He. 2018. TwinBee: Reliable Physical-Layer Cross-Technology Communication with Symbol-Level Coding. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 153–161.

[8] Wei Cheng, Aidong Xu, Yixin Jiang, Hong Wen, Huanhuan Song, Kai Ouyang, and Xiping Zhu. 2017. The realization of key extraction based on USRP and OFDM channel response. In *Communications and Network Security (CNS), 2017 IEEE Conference on*. IEEE, 374–375.

[9] Zicheng Chi, Zhichuan Huang, Yao Yao, Tiantian Xie, Hongyu Sun, and Ting Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[10] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2w2: N-way concurrent communication for iot devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 245–258.

[11] Boris Danev and Srdjan Capkun. 2009. Transient-based identification of wireless sensor nodes. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 25–36.

[12] Andrea Goldsmith. 2005. *Wireless Communications*. Cambridge university press.

[13] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. WIDE: physical-level CTC via digital emulation. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 49–60.

[14] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu, and Yunhao Liu. 2019. Lego-fi: Transmitter-transparent ctc with cross-demapping. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2125–2133.

[15] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. Wizig: Cross-technology energy communication over a noisy channel. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[16] Ibbad Hafeez, Markku Antikainen, and Sasu Tarkoma. 2019. Protecting IoT-environments against Traffic Analysis Attacks with Traffic Morphing. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 196–201.

[17] Jehad M Hamamreh, Haji M Furqan, and Huseyin Arslan. 2018. Classifications and Applications of Physical Layer Security Techniques for Confidentiality: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* (2018).

[18] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. 2018. Achieving Receiver-Side Cross-Technology Communication with Cross-Decoding. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 639–652.

[19] Wenchao Jiang, Ruofeng Liu, Ling Liu, Zhijun Li, and Tian He. 2017. BlueBee: 10,000 x Faster Cross-Technology Communication from Bluetooth to ZigBee. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 486–487.

[20] Wenchao Jiang, Zhimeng Yin, Song Mim Kim, and Tian He. 2017. Transparent cross-technology communication over data traffic. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[21] Xiaocong Jin, Jingchao Sun, Rui Zhang, and Yanchao Zhang. 2015. SafeDSA: Safeguard dynamic spectrum access against fake secondary users. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 304–315.

[22] Song Min Kim and Tian He. 2015. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 317–330.

[23] Demijan Klinc, Jeongseok Ha, Steven W McLaughlin, Joao Barros, and Byung-Jae Kwak. 2011. LDPC codes for the Gaussian wiretap channel. *IEEE Transactions on Information Forensics and Security* 6, 3 (2011), 532–540.

[24] Zhijun Li and Tian He. 2017. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2–14.

[25] Zhijun Li and Tian He. 2018. LongBee: Enabling Long-Range Cross-Technology Communication. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 162–170.

[26] Zang Li, Wenyuan Xu, Rob Miller, and Wade Trappe. 2006. Securing wireless systems via lower layer enforcements. In *Proceedings of the 5th ACM workshop on Wireless security*. ACM, 33–42.

[27] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. 2019. HomeSnitch: behavior transparency and control for smart home IoT devices. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 128–138.

[28] Neal Patwari and Sneha K Kasera. 2007. Robust location distinction using temporal link signatures. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 111–122.

[29] NIST FIPS Pub. 2001. 197: Advanced encryption standard (AES). *Federal information processing standards publication* 197, 441 (2001), 0311.

[30] Donald R Reising, Michael A Temple, and Mark E Oxley. 2012. Gabor-based RF-DNA fingerprinting for classifying 802.16 e WiMAX mobile subscribers. In *2012 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 7–13.

[31] Matthias Schulz. [n.d.]. Nexmon. https://github.com/seemoo-lab/nexmon

[32] Matthias Schulz. 2018. *Teaching Your Wireless Card New Tricks: Smartphone Performance and Security Enhancements Through Wi-Fi Firmware Modifications*. Ph.D. Dissertation. Technische Universität.

[33] Matthias Schulz, Jakob Link, Francesco Gringoli, and Matthias Hollick. 2018. Shadow Wi-Fi: Teaching Smartphones to Transmit Raw Signals and to Extract Channel State Information to Implement Practical Covert Channels over Wi-Fi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 256–268.

[34] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. 2017. Nexmon: The c-based firmware patching framework.

[35] Amit Kumar Sikder, Leonardo Babun, Hidayet Aksu, and A Selcuk Uluagac. 2019. Aegis: a context-aware security framework for smart home systems. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 28–41.

[36] ZigBee Specification. 2012. Document 053474r20. *Zigbee Standards Organization: San Ramon, CA, USA* (2012).

[37] statista. [n.d.]. statista report. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[38] Chan Dai Truyen Thai, Jemin Lee, and Tony QS Quek. 2016. Physical-layer secret key generation with colluding untrusted relays. *IEEE Transactions on Wireless Communications* 15, 2 (2016), 1517–1530.

[39] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A Gunter. 2019. Charting the Attack Surface of Trigger-Action IoT Platforms. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1439–1453.

[40] Shuai Wang, Zhimeng Yin, Zhijun Li, and Tian He. 2018. Networking Support For Physical-Layer Cross-Technology Communication. In *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 259–269.

[41] Eric W. Weisstein. [n.d.]. Parseval's Theorem. http://mathworld.wolfram.com/ParsevalsTheorem.html

[42] Wikipediak. [n.d.]. Pearson Correlation Coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

[43] Qian Xu, Pinyi Ren, Houbing Song, and Qinghe Du. 2017. Security-aware waveforms for enhancing wireless communications privacy in cyber-physical systems via multipath receptions. *IEEE Internet of Things Journal* 4, 6 (2017), 1924–1933.

[44] Chunxuan Ye, Suhas Mathur, Alex Reznik, Yogendra Shah, Wade Trappe, and Narayan B Mandayam. 2010. Information-theoretically secret key generation for fading wireless channels. *IEEE Transactions on Information Forensics and Security* 5, 2 (2010), 240–254.

[45] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. 2017. C-morse: Cross-technology communication with transparent morse coding. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[46] Sihan Yu, Xiaonan Zhang, Pei Huang, and Linke Guo. 2019. Secure Authentication in Cross-Technology Communication for Heterogeneous IoT. In *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 1–2.

[47] Marwan Yusuf and Huseyin Arslan. 2016. Controlled inter-carrier interference for physical layer security in OFDM systems. In *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*. IEEE, 1–5.

[48] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. 2018. Homonit: Monitoring smart home apps from encrypted traffic. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1074–1088.

[49] Xiaonan Zhang, Pei Huang, Linke Guo, and Yuguang Fang. 2019. Hide and Seek: Waveform Emulation Attack and Defense in Cross-Technology Communication. In *Proceedings of the 39th Annual International Conference on Distributed Computing Systems*. IEEE, 1–10.

[50] Xiaonan Zhang, Pei Huang, Linke Guo, and Mo Sha. 2019. Incentivizing relay participation for securing IoT communication. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 1504–1512.

[51] Xiaonan Zhang, Pei Huang, Qi Jia, and Linke Guo. 2018. Cream: Unauthorized secondary user detection in fading environments. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 406–414.

[52] Xiaonan Zhang, Qi Jia, and Linke Guo. 2017. Secure and optimized unauthorized secondary user detection in dynamic spectrum access. In *2017 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.

[53] Yifan Zhang and Qun Li. 2013. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 1366–1374.

[54] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2018. StripComm: Interference-resilient cross-technology communication in coexisting environments. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 171–179.