Reactive Jamming and Attack Mitigation over Cross-Technology Communication Links

GONGLONG CHEN and WEI DONG, College of Computer Science, Zhejiang University, Alibaba-Zhejiang University Joint Institute of Frontier Technologies

Recently, Cross-Technology Communication (CTC), allowing the direct communication among heterogeneous devices with incompatible physical layers, has attracted much research attention. Many efficient CTC protocols have been proposed to demonstrate its promise in IoT applications. However, the applications built upon CTC will be significantly impaired when CTC suffers from malicious attacks such as jamming or sniffing. In this article, we implement a reactive jamming system, JamCloak, that can attack most existing CTC protocols. To this end, we first propose a taxonomy of the existing CTC protocols. Then based on the taxonomy, we extract essential features to train a CTC detection model, and estimate the parameters that can efficiently jam CTC links. Experimental results show that JamCloak consistently achieves 94.7% of classification accuracy on average in both Line-of-Sight and Non-Line-of-Sight scenarios. We also apply JamCloak to attack three existing CTC protocols: WiZig, Esense and EMF. Results show that JamCloak can significantly reduce PDR (packet delivery ratio) by 80.8% on average in practical environments. In the meantime, JamCloak's jamming gain is more than 1.78× higher than the existing reactive jammer. In addition, we propose a practical countermeasure against reactive jamming attacks over CTC links like JamCloak. Results show that our approach significantly improves the jamming detection accuracy by 91.2% on average than the existing approach, and effectively decreases the reduction in packet delivery ratio to 1.7%.

CCS Concepts: • **Networks** → *Cross-layer protocols*; • **Security and privacy** → *Mobile and wireless security*;

Additional Key Words and Phrases: Cross-technology communication, jamming attack

ACM Reference format:
Gonglong Chen and Wei Dong. 2020. Reactive Jamming and Attack Mitigation over Cross-Technology Communication Links. *ACM Trans. Sen. Netw.* 17, 1, Article 4 (November 2020), 25 pages.
https://doi.org/10.1145/3418210

1 INTRODUCTION

According to Gartner, a well-known IT research and advisory company, the number of IoT (Internet of Things) devices will reach 20.4 billion by 2020 [15]. These IoT devices are envisioned to employ highly heterogeneous wireless technologies such as WiFi, ZigBee, Bluetooth, and so on, causing difficulties in directly interconnecting these devices due to completely different PHY-layer

This work is supported by the National Science Foundation of China under Grant No. 61772465, and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant No. LR19F020001.
Authors' addresses: G. Chen and W. Dong (corresponding author), College of Computer Science, Zhejiang University, Alibaba-Zhejiang University Joint Institute of Frontier Technologies, No. 38 Zheda Road, West Lake District, Hangzhou, Zhejiang, China; emails: {desword, dongw}@zju.edu.cn.

1550-4859/2020/11-ART4 $15.00
https://doi.org/10.1145/3418210

technologies. Traditional approaches require dedicated gateways with multiple radios. The main drawbacks include (1) additional costs, (2) additional traffic flow into and from gateways, and (3) possible congestion or collision near the gateways.

An attractive approach is to allow Cross-Technology Communication (CTC), i.e., the direct communication among these heterogeneous devices without a centralized gateway. CTC has attracted much research attention and many efficient CTC protocols have been proposed in recent years [3, 8, 9, 19, 22, 24, 25, 28, 35, 44–46, 48]. Previous studies have demonstrated the promise of CTC in many IoT applications. For example, using CTC, the Cross-Technology Interference (CTI) problem [5, 6] can be much more effectively solved by sharing the spectrum in a TDMA fashion [3]. Or achieving efficient concurrent transmissions for different wireless protocols [8]. Or enabling real-time patient monitoring by combining the ubiquitous deployed WiFi with the low power Bluetooth (BLE) [35]. CTC, like other wireless communications, could suffer from malicious attacks such as jamming [31] or sniffing [1]. Under attacks, the applications built upon CTC will be significantly impaired, e.g., missing urgent events or leaking private information. Therefore, to guarantee the reliable and effective communications over CTC links, it is very important to study the CTC security (e.g., by exploring the feasibility of performing powerful attacks).

In this study, we focus on the jamming attack on CTC and its countermeasures, which is imperative to the practical applications of CTC. With regard to jamming attacks, we focus on the reactive jamming attack, i.e., it starts jamming only when a network activity is observed [14], to achieve a powerful jamming attack. As opposed to reactive jamming attacks, proactive jamming attacks continuously sends packets or random bits on the channel and thus could be easily detected [14]. For example, in a CSMA network, the carrier sensing time distribution under normal conditions is known and can be acquired either theoretically or empirically. Monitoring for deviations from the benign distribution can be used for detecting proactive jamming, but not for reactive jamming, this is because reactive jamming does not occupy the channel [31]. In this article, we are interested in the following two questions: (1) Can we design a powerful and generic reactive jamming attack system over CTC links? and (2) What countermeasures could be taken to secure the CTC system?

Answering the above questions, however, faces several practical challenges. First, the existing reactive jamming approach cannot be directly applied to CTC due to the totally different modulation schemes. For example, FreeBee [35] modulates CTC bits by shifting the timing of periodic beacon frames and different timing patterns are demodulated accordingly. Existing reactive jamming attacks can only jam the ongoing packets (e.g., jamming WiFi packets using WiFiJamer [37]) but not the timing patterns, therefore these attacks are easily bypassed by FreeBee. Second, existing CTC protocols have used a variety of modulation schemes due to different application scenarios. For example, WiZig [19] can achieve relatively high throughput by mapping different energy levels to CTC bits, but it can only be used in stationary scenarios (e.g., monitoring smart home applications [3]) due to the fixed energy level mapping relationship. FreeBee [35] can be used in both mobile and stationary scenarios, but it is only suitable for non-delay sensitive applications due to its low throughput. It is of importance to design an attacking system against as many CTC protocols as possible in different scenarios. However, existing CTC protocols are very different from each other (e.g., modulating timing patterns or energy levels), it is thus challenging to devise a generic reactive jamming system against most CTC protocols.

To address the above challenges, we implement a reactive jamming system, JamCloak, that can attack most existing CTC protocols. JamCloak consists of two components: detecting CTC activities and performing jamming attacks. JamCloak detects CTC activities by classifying the CTC traffic from normal traffic. To this end, we first propose a taxonomy of the existing CTC protocols. We observe that in an energy sensing-based receiver, there are three possible energy characteristics that can be detected: the intensity of the energy, the duration of the energy, and the gap between

the energy. We thus classify the existing CTC protocols into three categories: energy level-based protocol [9, 19], packet length-based protocol [3, 46], and packet reorder-based protocol [8, 24, 35, 44]. Then based on the taxonomy, the features of each category are extracted according to the observation: The existing CTC protocols construct the energy characteristics with a large difference in normal traffic, guaranteeing to ensure the CTC information can be demodulated efficiently. For example, Esense [3] modulates CTC information using the packet length that is unusual in normal traffic. Then the deviations from the normal distribution can be used to detect packet length-based protocols. Based upon this observation, we extract essential features and train a decision tree model to classify the CTC traffic from the normal traffic. In this way, the CTC activities are thus detected. To perform jamming attacks, we need to design jamming signals that can effectively attack the specific CTC protocol. JamCloak utilizes k-means to estimate the signal patterns and then transmits jamming signals.

To counteract reactive jamming attacks over CTC links like JamCloak, we propose an effective reactive jamming detection and mitigation approach. The existing approach either uses the signal strength or location consistency checks to detect reactive jamming. However, from our experimental results we find that existing detection metrics (i.e., signal strength or location) cannot effectively detect CTC reactive jamming attacks. Because most existing CTC protocols rely on the traffic pattern to convey information and therefore its performance (e.g., packet delivery ratio) is sensitive to the background traffic density. We thus propose a new metric that involves in both signal strength and background traffic density to effectively detect reactive jamming attacks. Finally, our countermeasure will not incur extra overhead, because we perform the mitigation approach only when jamming attacks are detected.

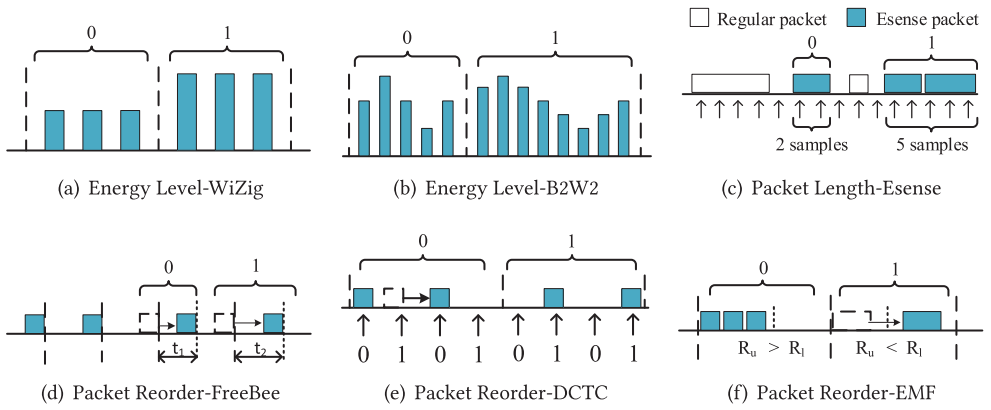
We summarize the contributions of this work as follows:

- To the best of our knowledge, we propose the first taxonomy of the existing CTC protocols based on the energy characteristics.
- We implement a reactive jamming system, JamCloak, that can attack most existing CTC protocols. Results show that JamCloak can consistently achieve higher than 94.7% of classification accuracy for a wide SNR range in both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) scenarios. Extensive experiment results show that JamCloak can significantly reduce the packet delivery ratio by 80.8% on average in practical environments. In the meantime, JamCloak's jamming gain is more than 1.78× higher than the existing reactive jammer.
- We propose a practical countermeasure against reactive jamming attacks over CTC links like JamCloak. Results show that our approach consistently improves the jamming detection accuracy by 91.2% on average than the existing approach, and effectively decreases the reduction in packet delivery ratio to 1.7%.

The rest of this article is organized as follows. Section 2 presents the taxonomy of the existing CTC approach. Section 3 gives an overview of JamCloak. Section 4 and Section 5 show the key component of JamCloak. Section 6 presents the evaluation results. Section 7 discusses a practical countermeasure against reactive jamming attacks over CTC links. Section 8 introduces the related work and finally, Section 9 concludes this article.

2 TAXONOMY OF CTC PROTOCOLS

To achieve direct communication among heterogeneous devices with incompatible physical layers, sensing the energy patterns on the channel is a promising way that can be supported by many COTS devices. Based on the energy characteristics, we classify the existing CTC protocols into three categories: energy level based [9, 19], packet length based [3, 46], and packet reorder based [8, 24, 35, 44].



(a) Energy Level-WiZig (b) Energy Level-B2W2 (c) Packet Length-Esense

(d) Packet Reorder-FreeBee (e) Packet Reorder-DCTC (f) Packet Reorder-EMF

Fig. 1. Taxonomy of existing CTC protocols.

2.1 Energy Level-based Protocol

By changing the transmission power according to a certain pattern, heterogeneous devices can sense the pattern of the energy level+ and then demodulate information accordingly. For example, WiZig [19] regulates the transmission power of each packet with two different energy levels to modulate CTC bit "0" and "1." The bit error rate (BER) can be reduced by repeating each CTC bit multiple times as shown in Figure 1(a). The energy level pattern of this approach is similar to the square wave. In addition to the square wave, one can also modulate the energy level into other waveform, such as sine wave. B2W2 [9] encodes CTC bits by directly adjusting the adjacent packets' transmission power to form a sine wave as shown in Figure 1(b). CTC bits "0" and "1" are distinguished by changing the frequency of the sine wave.

2.2 Packet Length-based Protocol

According to the existing work [3], the majority of packet length follows a certain distribution (e.g., a bimodal distribution that either small packets corresponding to the ACKs, beacons and management frames. Or they are around 1500-byte packet corresponding to the MTU). This observation leaves an opportunity to modulate CTC bits by transmitting packets with un-regular packet length. For example, Esense [3] and HoWIES [46] encode CTC bits by mapping them to an appropriate alphabet set of packet length. The packet length that does not normally occur is assigned into the alphabet set, such that the modulated packets can be distinguished from regular packets and the BER is reduced. To further enlarge the size of alphabet set, one can construct a merged packet that exceeds the maximum packet length (e.g., leveraging A-MPDU standard [16]), while following the current IEEE 802.11 standard [44].

2.3 Packet Reorder-based Protocol

In normal WiFi traffic, the transmission gap between data packets does not exhibit periodicity due to the IEEE 802.11 standard, such as short inter-frame space (SIFS), DCF inter-frame space (DIFS), and random backoff time. We can thus modulate CTC bits by reordering packet transmission time to construct a periodic pattern that can be demodulated by the receiver. For example, FreeBee [35] modulates one CTC bit by shifting the timing of periodic beacon frames as shown in Figure 1(d). CTC bits "0" and "1" are distinguished by changing the shifting time. DCTC [24] encodes CTC bits by first setting the critical time points within a synchronized time windows, and then shifting data packets to the certain critical time points that have alternating labels to indicate CTC bits "0"

and "1" as shown in Figure 1(e). EMF [8] modulates CTC bits by slightly shifting the packets order to form a unique pattern. Specifically, as shown in Figure 1(f), within the two synchronized time window, the left part with larger packet occupancy ratio denotes CTC bit "0" and vice visa.

2.4 Summary

To improve CTC throughput while reducing BER, a common feature of existing CTC protocols is constructing an un-regular energy characteristic that is distinguished from normal traffic (e.g., un-regular energy levels, packet length and packet transmission gaps). So, in principle, the CTC traffic can be detected by monitoring the deviations from normal traffic, which also poses a big threat to the existing CTC protocols.

Recently, many high throughput CTC protocols have been proposed [7, 25, 28, 29]. They key idea of these protocols is that they emulate different wireless protocols at the PHY-layer, and the desired bits are selected at the application layer, the CTC receiver can then decode the CTC information without hardware modifications. For example, WEBee [28] provides an analog emulation approach to let WiFi signals to emulate the wave of ZigBee signals, achieving up to 250kbps data rate. WIDE [17] provides a digital emulation approach to improve the communication reliability, e.g., increase the PDR from 41.7% to 86.2%. However, these high throughput CTC protocols still utilize the original PHY-layer modulation scheme to deliver CTC information. Therefore, existing jammers can already attack them well, such as WiFiJam [37].

There are also some CTC protocols [18] that JamCloak may not attack them well due to the coarse-grained information we utilized, i.e., RSSI. However, we found that the key idea of this kind of CTC protocols can also be classified into the energy level or packet reorder protocols. Because they actually modulate the energy level in the grained of Channel State Indicator (CSI) to deliver CTC information. Therefore, by upgrading JamCloak with fine-grained CSI information, i.e., at the first step of signal preprocess as shown in Figure 2, JamCloak can also be used to attack this kind of CTC link. We would like to enhance JamCloak to support sampling the fine-grained CSI information in the future.

In the rest of this article, we will detail how to conduct a powerful reactive jamming that can attack most existing CTC protocols.

3 CHALLENGES AND SYSTEM OVERVIEW

3.1 Objectives and Challenges

We study the CTC security by exploring the feasibility of performing reactive jamming attack over CTC links. JamCloak only performs jamming attack when there are CTC activities. To maximize the jamming gain [31], JamCloak should significantly reduce CTC link quality with only a small amount of jamming signals. To achieve above goals, we have to address the following challenges:

C1. How to effectively detect CTC activities without knowing the settings of the victims? A key characteristic of reactive jamming attack is performing attacks only when CTC activities are detected. To detect CTC activities from normal traffic, it is important to achieve an effective classification of CTC protocols. However, it is challenging especially without the prior knowledge of network settings of victims. For example, to classify the packet length-based protocols, a possible way is to compare the packet length distribution between the normal traffic and the received traffic. However, the packet length distribution of normal traffic varies depending on WiFi data rates (e.g., higher WiFi data rates would shorten the sampled length of packets and the distribution is thus changed). To address this challenge, we transform the problem of calculating the difference into a problem of inferring the normal traffic statistics. Then it is possible to extract essential features

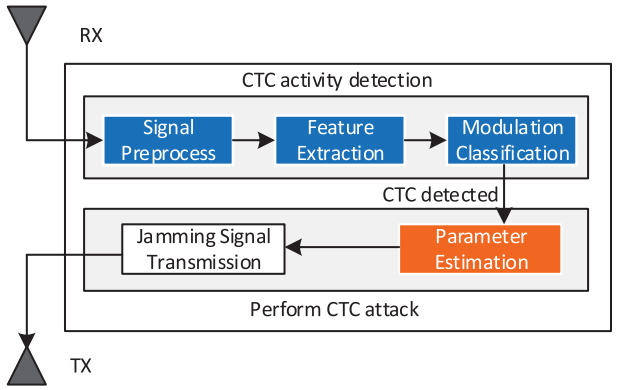


Fig. 2. Overview of JamCloak.

without the knowledge of the victims. For example, the high-frequency packet length of the normal traffic is stable according to prior works [3], we can thus extract this feature to detect Esense [3].

C2. How to reduce the CTC link quality using a small amount of jamming signals? To maximize the jamming gain, the jammer should reduce CTC link quality using as small amount as possible jamming signals. To this end, the jammer needs to estimate the parameters of CTC modulation and then produces an appropriate noise pattern to disrupt the decoding of the victims, or a faked message that can be decoded by the victims. However, it is challenging even with the known CTC categories due to the parameter similarities among the intra-class modulations. For example, to estimate the critical time points of packet reorder protocol, DCTC [24], a possible way is to extract the periodic time points using the fast folding algorithm [35]. However, since EMF [8] moves packets into a sub-window to modulate CTC bits, the folding results will also show a strong periodicity on the positions of reordered packets. Therefore, it is challenging to determine which intra-class modulation should be used for the estimated parameters. To address this challenge, we utilize k-means to further classify the intra-class modulation schemes.

3.2 System Overview

JamCloak is designed as a passive attacker that targets at the CTC links from WiFi to ZigBee. Its high level system architecture is shown in Figure 2. There are two core steps in JamCloak: detecting CTC activity and performing jamming attacks.

(1) Detecting CTC activity. To detect the CTC activities, JamCloak continuously samples the received signal strength (i.e., RSSI) and preprocesses them for feature extraction. Then the extracted features are fed into the off-line trained decision tree model to classify CTC traffic from normal traffic. JamCloak only conveys the classification results to the next step when CTC activities are detected. Otherwise it goes back to the first step to sample RSSI.

(2) Performing CTC jamming attacks. Once the CTC activities are detected, JamCloak will estimate the parameters of the detected CTC protocol. After that, JamCloak transmits jamming signals over CTC links according to the estimated parameters.

As shown in Figure 3, it presents how CTC protocols work in real wireless context. We use the energy level-based CTC protocol WiZig as the illustrative example.

For the CTC protocol, the CTC sender needs to change to energy level of WiFi packets to transmit CTC bits. At the CTC receiver side (i.e., the ZigBee device as shown in Figure 3), different energy levels are sampled from the channel and decoded to CTC bits correspondingly. At the WiFi

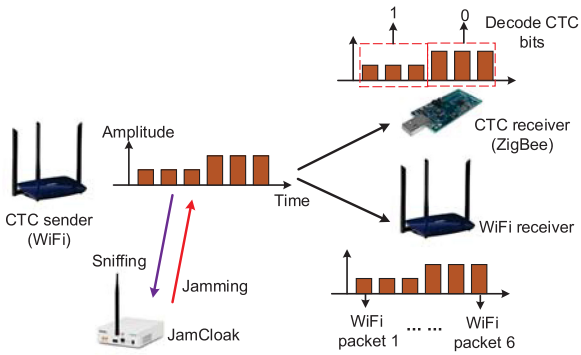


Fig. 3. Scenarios of CTC protocols and JamCloak.

receiver side, it can directly decode the WiFi packets normally. For other kind of CTC protocols, the procedures are similar.

When the packets from the legacy sender and the CTC sender are interleaved, the CTC decoding may fail due to the noise introduced by the legacy packets. The legacy packets may not contain the expected CTC info like the energy levels, which leads to an inaccuracy of CTC decoding.

However, it is possible to mitigate the impacts caused by the legacy packets by introducing extra robust mechanisms. In WiZig, the CTC sender transmits multiple packets with the same energy level to convey the same CTC information in a sliding window. The CTC receiver decodes the CTC information from the sliding window by extracting the energy levels that are delivered by a specific number of packets. For example, when the sliding window contains seven packets, and the WiZig receiver treats the energy level appears larger than four times as the conveyed CTC information. Otherwise, WiZig receiver recognizes that the CTC information is decoding incorrectly. In the above settings, fewer than two legacy packets are allowed in the sliding window. And the decoding threshold can be set according to the link quality that is stated in WiZig.

For JamCloak, as shown in Figure 3, it continuously sniffers the traffic on the air, and performs the jamming attack once the CTC traffic is detected. As we stated in Section 7, to mitigate this kind of attacks, we propose a new approach to detect the jamming. When the attacks are detected, the CTC sender and receiver will hop to another channel to continue the communications.

4 CTC ACTIVITY DETECTION

To perform a reactive jamming attack over CTC links, determining whether the CTC is active or not is an important step. Thus, efficiently classifying the CTC protocols is the core step to capture CTC activities. Specifically, there are three steps to classify the CTC protocols: preprocessing signal, extracting feature, and constructing CTC protocol classification model.

4.1 Signal Sampling and Preprocessing

We utilize USRP N210 to sample RSSI series at high frequency about 1 MHz [32]. Formally, the sampled RSSI sequence can be expressed as *I[]*. Before these sampled signals can be fed into the feature extraction module, JamCloak needs to preprocess the signal as follows:

4.1.1 Extracting Packet Level Energy. The energy level-based protocols modulate CTC bits at packet level. That is, changing the transmission power of packets to convey CTC information. However, the sampled RSSI sequence consists of not only the energy pattern of packet level, but also in-packet level. It would have negative impacts on the classification of the energy level-based protocols, especially when WiFi traffic are modulated by energy level-like modulation (e.g., QAM).

Thus JamCloak first needs to extract the packet level energy sequence. The high RSSI sampling rate ensures every WiFi packet can be distinguished (e.g., SIFS is 10 μs and DIFS is 50 or 28 μs [16]), we thus treat the consecutive RSSI sequence that is larger than the threshold *pktThd* (e.g., packet detection threshold) as one packet. The sets of WiFi packet begin (B) and end (E) positions are as follows:

$$\begin{aligned} B &= \{b | I[b-1] - I_n < pktThd, I[b] - I_n \geq pktThd\}, \\ E &= \{e | I[e] - I_n \geq pktThd, I[e+1] - I_n < pktThd\}, \end{aligned} \tag{1}$$

where I_n denotes the noise floor. For a given transmission, the maximum amplitude within a packet is constant [13], we thus extract the maximum RSSI value within a packet to denote the packet level energy. Then the energy of ith packet can be denoted as $max(I[B[i]], I[E[i]])$. The sets of WiFi packet level energy are as follows:

$$SE = \{max(I[B[i]], I[E[i]]) | 0 \leq i \leq len(B)\}. \tag{2}$$

4.1.2 Extracting Packet Length. To analyze the packet length-based CTC protocols, we need extract packet length data as follows:

$$SL = \{E[i] - B[i] | 0 \leq i \leq len(B)\}. \tag{3}$$

We merge adjacent packet length if their arrival time interval is less than 50 μs according to 802.11 standard [16].

4.1.3 Extracting Packet Interval. For packet reorder-based CTC protocols, the interval among data packets has been changed and is different from normal traffic. Thus we need to extract packet interval data:

$$ST = \{E[i] - B[i-1] | 1 \leq i \leq len(B)\}. \tag{4}$$

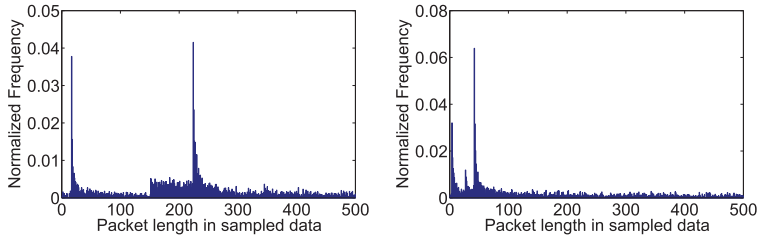
4.2 Feature Extraction

In this section, we will detail how to extract essential features to classify CTC protocols.

4.2.1 Feature for Distinguishing Energy Level-based Protocol. Now that we have obtained the energy changes of packet level, the energy level-based protocols can be identified using the variance of the packet level energy sequence, since the CTC traffic will have a greater variance than normal traffic. To reduce the effects of noise, we can filter noise using the low-pass filtering-based approach [20]. We choose the simple moving average (SMA) approach due to its simplicity and effectiveness. We thus extract the SMA of packet level energy sequence for distinguishing energy level protocols: $FE = \overline{SE}$.

4.2.2 Feature for Distinguishing Packet Length-based Protocol. To identify the packet length-based protocols, a straightforward way is computing the distance of packet length distribution between the received traffic and the normal traffic, since the packet length of CTC traffic is different from normal traffic such that the CTC receiver can demodulate CTC bits via packet length. However, the packet length distribution of normal traffic varies depending on WiFi data rates as shown in Figure 4. For the same packet length that is transmitted at the sender side, the sampled length at the receiver side is changed with WiFi data rates changing (e.g., sampled length for 1,500-bytes packet changes from 220 samples to 40 samples when WiFi data rates changes from 54 to 300 Mpbs). The deviations from normal traffic are thus challenging to be used for distinguishing packet length-based protocols.

To identify packet length-based protocols, we can extract the number of models in the packet length distribution as the feature. The intuition is that the number of models is up to two for normal traffic. Because packet length distribution of normal traffic can only be bimodal or unimodal [3]



(a) 54 Mbps WiFi data rate. (b) 300 Mbps WiFi data rate.

Fig. 4. Packet length distribution of normal WiFi traffic (static data rates).

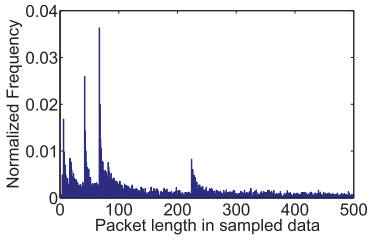


Fig. 5. Packet length distribution of normal WiFi traffic with rate adaptation (54, 180, and 300 Mbps).

considering different WiFi data rates (as shown in Figure 4). In other words, if the number of models is greater than two then we determine current traffic is modulated by packet length-based protocols.

It works well when the WiFi data rate is static [4]. However, there are link rate adaptation mechanisms in WiFi to improve the throughput [26], and the number of models may not be a suitable feature in this scenario. For example, there are six models when WiFi AP can adjust its data rate among 54, 180, and 300 Mbps (as shown in Figure 5), indicating the higher false positive when applying the feature in Reference [4].

Then how to identify the packet length-based protocols reliably under scenarios with static or dynamic data rates? We found that in rate adaptation scenarios, to reduce the overhead of synchronizing the modulation information (e.g., mapping from the packet length to the data bits) between the sender and the receiver, the sender often takes the responsibility to figure out the packet size and the rate at which to send this packet, such that the receiver can decode CTC information from the sampled packet length without synchronization [3]. For example, to convey data bits "01," which is denoted by 110 samples, the WiFi sender needs to transmit 700-byte packets with 54 Mbps while 1,400-byte packets with 108 Mbps. Similarly with other CTC protocols, to reduce the bit error rate, the sampled packet length in CTC traffic should be greatly distinguishable with the length of normal traffic.

Based on above observations, we can conclude that the set of sampled packet length in normal traffic is stable even with dynamic data rates. Therefore, the number of remained packet length after eliminating the packet length in normal traffic can then be extracted as the feature to classify the packet length-based protocol.

Algorithm 1 shows how to efficiently obtain the feature. Note that the model estimation algorithm in Reference [4] cannot be applied, because it can only estimate the number of models but not the set of modulated packet length. To identify the packet length-based protocols reliably in scenarios with static and dynamic data rates, we propose a detection approach based on z-score metric [36]. As shown in Algorithm 1, there are three input parameters, i.e., the packet length

ALGORITHM 1: Estimating the number of modulated packet length.

Input: Packet length sequence SL[], normalized histogram of packet length NH[], normal packet length set RS[].
Output: The number of modulated packet length.

```
1  Smooth factor α; Window size W;
2  Peak set PS = [];
3  for i=W+1: i < len(NH): i++ do
4      meanNH = mean(NH[i-W:i-1]);
5      stdNH = std(NH[i-W:i-1]);
6      if NH[i]- meanNH> stdNH then
7          /*Peak value detected, save the packet length.*/;
8          PS.append(SL[i]);
9          /*Smooth current peak value for latter detection.*/;
10         NH[i] = α*NH[i]+(1-α)*NH[i-1];
11 /*Remove the packet length belonging to normal packet length set*/;
12 for i=0: i < len(PS): i++ do
13     if PS[i] belongs to RS then
14         drop PS[i];
15 return k=len(PS);
```

sequence SL, the normalized histogram of packet length NH and the normal packet length set RS. Among them, the set RS stores the sampled packet length under normal WiFi traffic with various data rate, e.g., from 1 to 54 Mbps [3]. To extract the modulated packet length, the basic idea is to detect the packet length with bursty value (peak value) in histogram. Then, to effectively detect peak value in histogram, the algorithm utilizes the intuition that when the differences between the current value and the average value over a moving window M (with length W) exceeds a threshold, then the current value is identified as a peak. In Algorithm 1, the threshold is set to the standard deviation over the moving window M. The peak value would highly increase the average value over the window M, leading to a undesirable increased threshold. To compensate the impact of the peak value, we smooth the value with the factor α (as shown in line 10). Finally, we remove the detected packet length belonging to the normal packet length, and return the number of packet length that is possibly used for the CTC modulation. Then feature is set to $FL = k$.

4.2.3 Feature for Distinguishing Packet Reorder-based Protocol. We leverage the intuition that for packet reorder-based protocols, the interval between WiFi packets is changed and different from normal traffic. According to existing work that the packet interval of normal WiFi traffic would follow the Pareto model [21], which cannot be fitted by the modulated CTC traffic.

Then we use the Kolmogorov-Smirnov Test (K-S test) of 0.95 significance to evaluate the goodness-of-fit of fitting the received packet interval ST using Pareto model. We divide the trace into W equal sized windows and record the number P of passing K-S test. Then the feature for distinguishing packet reorder-based protocols can be represented as the passing rate of K-S test: $FT = P/W$.

4.3 Classification Model Construction

After extracting all essential features for each CTC protocol, we then construct the classification model using C4.5 algorithm [10].

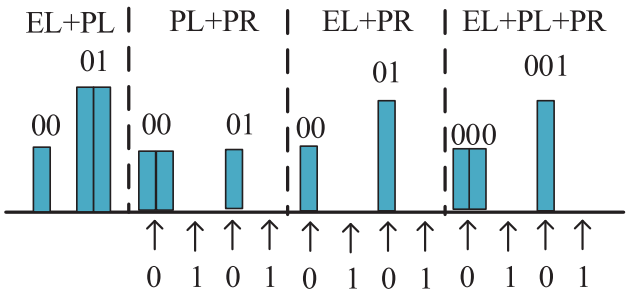


Fig. 6. New CTC protocols. EL: Energy Level, PL: Packet Length, PR: Packet Reorder.

To validate the generality of our model, we also design four new CTC protocols-based on the existing CTC protocols. As shown in Figure 6, these CTC protocols are described below: (1) Energy level and packet length ($EL + PL$): It conveys two bits of CTC information by changing the energy level and length of each packet. (2) Packet length and packet reorder ($PL + PR$): It conveys two bits of CTC information by varying the packet length at the critical time points (similar with CMorse [44]). (3) Energy level and packet reorder ($EL + PR$): it conveys two bits of CTC information by varying the transmission power of packets at the critical time points. (4) Combination of three ($EL + PL + PR$): It conveys three bits of CTC information by changing the energy level and length of each packet at the critical time points. We also construct the classification model for all above seven categories using C4.5 algorithm [10]. In Section 6, we will detail the evaluation setup and results show that our model achieves high accuracy not only in existing CTC protocols but also in new CTC protocols.

5 CTC MODULATION PARAMETERS ESTIMATION

To maximize the jamming gain [31], the jammer should reduce CTC links quality using as small amount as possible jamming signals. To this end, the jammer needs to estimate the parameters of CTC modulation and then produces an appropriate noise pattern to disrupt the decoding procedure. However, it is challenging to estimate the parameters even with the known CTC modulation scheme due to the parameter similarity among the intra-class modulation.

5.1 Estimating Parameters of Energy Level Modulation

We assume that the possible modulated energy level patterns are known (e.g., square wave for WiZig [19] and sine wave for B2W2 [9]), but the parameters of the specific pattern are unknown. This assumption is reasonable, because it is similar with the current QAM or ASK modulation that the way of modulation is known but the parameters (e.g., 16-QAM or 64-QAM) vary with different systems.

Then how to estimate the parameters given a specific signal pattern? We first segment the received signal based on the signal pattern and then utilize clustering analysis to estimate the parameters. The intuition is that for the energy level modulation patterns that are used to convey useful CTC information, they are repeated many times and the energy level patterns modulated by the same parameter will show a strong correlation. In other words, it is possible to cluster the segmented energy level patterns into multiple sets that have strong correlation. Then for the first k sets containing the most elements, the energy-level patterns they contain will most likely be the patterns currently used to modulate CTC information.

To this end, there are three steps to estimate the parameters: (1) segmenting the energy level patterns, (2) clustering, and (3) extracting parameters.

5.1.1 Segmenting Energy Level Pattern. According to the given energy level patterns, we can generate the segmentation rule set $R[]$ for each pattern. Suppose there are m possible patterns, we let $R[i]$ denote the segmentation rule for B2W2 modulation, $segThd$ denotes the threshold to determine the start and end of the segment. Then the rule $R[i]$ can be defined as follows:

$$R[i].split(a,b) = \begin{cases} beg, & \text{if } a < segThd,\ b \geq segThd, \\ end, & \text{if } a \geq segThd,\ b < segThd. \end{cases} \tag{5}$$

Then the positions sets of segmented signal begin ($B_s[i]$) and end ($E_s[i]$) for the ith energy level pattern are as follows:

$$\begin{aligned} B_s[i] &= \{b | R[i].split(SE[b-1], SE[b]) == beg\}, \\ E_s[i] &= \{e | R[i].split(SE[e], SE[e+1]) == end\}. \end{aligned} \tag{6}$$

Then we have the segmented signal set $S_g = \{S_g[1], S_g[2], \ldots, S_g[n]\}$, where $S_g[i] = SE[B_s[i] : E_s[i]]$.

5.1.2 Clustering Based on Correlation Distance. Then we use the k-means that replaces the distance indicator from Euclidean distance to the correlation distance (e.g., Pearson Correlation [2]) to cluster the segmented patterns. For every possible energy level patterns, the resulted cluster set is $C_g = \{C_{g1}[1], \ldots, C_{g1}[z], C_{g2}[1], \ldots, C_{gm}[z]\}$, where $C_{gx}[y]$ means the yth cluster for the xth energy level pattern. We only retain the first r (e.g., four in our system) clusters that have most elements among all possible patterns, because as long as one energy level pattern that is most likely to be successfully decoded by CTC, then the jamming attack is realized. We let the most common segmented signal of each remained cluster to form the final vector $FC = \{S_g[1], S_g[2], \ldots, S_g[r]\}$.

5.1.3 Extracting Parameters. To minimize the jamming signal duration, we select the segmented signal $S_g[min]$ with the minimal number of WiFi packets l from FC. Then JamCloak transmits l jamming signals with normal WiFi pakcet-on-air time each (e.g., 0.5 ms [47]) at the power in $S_g[min]$.

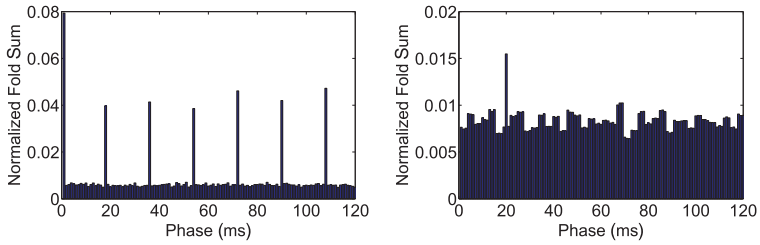
5.2 Estimating Parameters of Packet Length Modulation

Based on the obtained cluster set C from Section 4.2.2, we have the candidate set of modulated packet length. To extract the CTC modulated packet length, we first filter out the packet length of normal traffic by the following rules: the cluster that contains the shortest packet length, and the cluster that contains one of the packet that occurs merging in Section 4.1.2. Then we choose the shortest packet length pl from the remaining cluster set. JamCloak transmits jamming signals with pl ms each at the maximum power.

5.3 Estimating Parameters of Packet Reorder Modulation

We find that the folding peaks perform different distribution between DCTC [24] or FreeBee [35] and EMF [8]. As shown in Figure 7, it can be seen that EMF often causes a cluster of folding peaks because of their dense packet reordering, and DCTC or FreeBee often shows a sparse distribution of folding peaks because of their critical transmission time points.

Based on the above observation, we use k-means to cluster the folding peaks data and extract the parameters according to the number of elements in the resulted clusters. Algorithm 2 shows the details of estimating the average number of elements AveE in the resulted clusters. If AveE is larger than the threshold $ptThd$, then it is highly probable that the modulation is EMF, because more phase values are put into one cluster and it reflects the clustered folding peak feature of EMF. Otherwise, the modulation is FreeBee or DCTC. We first filter the noise folding data the same as Algorithm 1, then transform two dimension histogram data into one dimension data (e.g.,



(a) Folding results for DCTC. (b) Folding results for EMF.

Fig. 7. Normalized folding sum histogram.

ALGORITHM 2: Estimating the average number of elements in folding peaks cluster

Input: Fast folding phase histogram PH[].
Output: Average number of elements AveE, cluster CT.

```
for i=1: i < len(PH): i++ do
    if PH[i]/sum(PH) ≤ noiseThd then
        PH[i] = 0; /*filter noise phase.*/
OneD = [];
for i=0; i < len(PH); i++ do
    /*generate 1 × PH[i] data for each phase.*/
    OneD.append(repmat(i, 1, PH[i]));
find best k-means cluster set CT using Elbow method;
SumE = 0;
for i=0 ; i < len(CT) ; i++ do
    /* Drop the elements that are lower than borderThd. Collect the number of rest elements into
    SumE.*/
    isBorderC = len(CT[i])/range < borderThd;
    if isBorderC == True then
        drop CT[i] from CT;
    else
        /*remove duplicated elements.*/;
        CT[i] = unique(CT[i]);
        SumE += len(CT[i]);
return AveE = SumE/len(CT), CT;
```

repeating the phase by the folding count). Finally, the average number of elements AveE and the resulted clusters CT are returned.

However, we now only know the sub-modulation category but still do not know the modulation parameters, such as the sub-window size of EMF [8]. To ensure the successful jamming attack while reducing the transmission time of jamming signal, we utilize a conservative strategy to estimate the parameters. First, the most common phase of each cluster in CT is extracted to form the phase vector $H = \{ps_1, ps_2, \ldots, ps_{len(CT)}\}$. Then, for DCTC or FreeBee (e.g., AveE $\leq$ ptThd), JamCloak continuously transmits jamming signals with random packet intervals selected from the phase vector H. For EMF (e.g., AveE > ptThd), given the largest phase psm in H, JamCloak transmits jamming signals with $psm/2$ ms each at the maximum power. In this way, jamming signals carrying benign but undesirable phases are transmitted to degrade the ability of decoding CTC information.

5.4 Timely Reactive Jamming

To successfully perform reactive jamming, we need to complete the attack within the CTC packet on-air time. We thus simplify the whole jamming procedure as follows: (1) Only extract the feature of previous detected modulation scheme within an enough sliding window size, and (2) directly select an appropriate attack parameter from the already estimated range of parameters. To ensure that the jamming time of JamCloak is sufficient to destroy CTC packets in a wide range of SNR, we let JamCloak attack last for a relatively long time (e.g., more than half of the estimated time) in the case of small SNR.

5.5 Jamming PHY Emulation-based CTC protocols

Recently, many high throughput PHY emulation-based CTC protocols are proposed [7, 25, 28, 29]. Based on such high-throughput technologies, the application scenarios of CTC protocols can be broadened. Therefore, it is also necessary to discover the security of the PHY emulation-based CTC protocols. To this end, we upgrade JamCloak to attack this kind of CTC protocols.

The key idea of PHY emulation-based CTC protocols is to utilize the modulation similarity between the various radio technologies to transmit CTC information. For example, the frequency-shift keying can be transformed into an equation of phase-shift keying. Then by carefully selecting the CTC bits at the application layer, the CTC receiver can decode the information relying on the built-in error recovery function. Therefore, they still utilize the original modulation scheme to transit information, e.g., WEBee [28] still utilizes the WiFi OFDM symbols to deliver CTC bits to the ZigBee receiver. We can thus incorporate existing jammers in JamCloak to attack the PHY emulation-based CTC protocols, e.g., WiFiJam [37].

Specifically, JamCloak still performs the packet-level CTC traffic sniffing. When no packet-level traffics are detected, JamCloak switches to WiFiJam to jam the WiFi links. The WiFi packets can then be corrupted severely. As a result, the receiver side can hardly decode the CTC information due to the corrupted WiFi packets.

6 EVALUATION

In this section, we present a thorough evaluation of JamCloak performance. In the following, we first introduce our experimental methodology, and then discuss experiment results in detail.

6.1 Experimental Methodology

We implement a prototype of JamCloak on the USRP N210/ GNURadio platform [32] due to its wide range of transmission power and high RSSI sampling rate (e.g., 1 MHz). To implement existing CTC protocols, we use a USRP and a TelosB node, a commercial ZigBee platform, as a CTC transceiver pair at a distance of 1.2m as shown in Figure 8. We use the USRP platform to transmit WiFi packets following IEEE 802.11 standards to the TelosB node. The WiFi data rate is set to 54 Mbps, which is a common setting in current AP. Note that we assume the rate adaptation function in WiFi is off, because for CTC protocols using packet length-based modulation, the dynamically changed data rate will violate the packet length mapping relationship and thus degrade the performance of packet length-based CTC protocols. We select overlapped channels (i.e., 802.11 channel 11 and 802.15.4 channel 21) to construct the CTC channel.

To generate different SNR range (i.e., -2 dB ~ -10 dB), we use another USRP [32] to generate the Gaussian noise with different power. Two scenarios are evaluated as follows:

LoS: As shown in Figure 8, the LoS scenario is a hallway that is 10.2 m long and 1.2 m wide. We fix the distance between JamCloak and the CTC transceiver pair to 7.2 m, and the noise generator is placed between them.

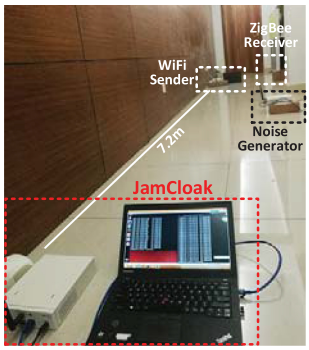


Fig. 8. Line-of-Sight

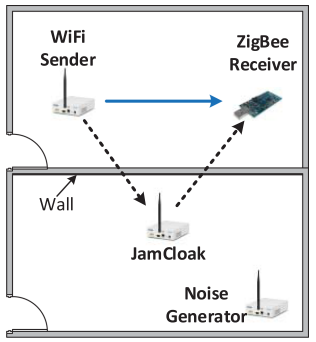


Fig. 9. None-Line-of-Sight.

Table 1. Modulation Parameter Settings

CTC Protocol	WiFi Packet Energy Level	WiFi Packet Length	Critical Time Interval	CTC Symbol Duration (1 bit)
WiZig/EL	3dBm, 12dBm	normal	-	6~7ms
Esense/PL	10dBm	2 pkts, 3 pkts	-	-
EMF/PR	10dBm	normal	-	4~5ms
EL+PL	3dBm, 12dBm	2 pkts, 3 pkts	-	6~7ms
PL+PR	10dBm	2 pkts, 3 pkts	2ms	11~12ms
EL+PR	3dBm, 12dBm	normal	2ms	11~12ms
EL+PL+PR	3dBm, 12dBm	2 pkts, 3 pkts	2ms	11~12ms

EL: Energy Level, PL: Packet Length, PR: Packet Reorder.

NLoS: As shown in Figure 9, in this scenario, the jammer and the CTC transceiver pair are deployed in two adjacent rooms that are separated by a wall.

To evaluate the accuracy and the generality of the classification model in JamCloak, we implement three existing CTC protocols (i.e., WiZig [19], Esense [3], and EMF [8]) that represent three categories, and four new CTC protocols as described in Section 4. The symbol error rate of all above seven CTC protocols is lower than 1% with the parameter settings in Table 1. The "x pkts" means Esense modulates CTC bits with x merged maximum packet length (i.e., 1,500 bytes). We also evaluate the impact of rate adaptation on the detection accuracy of packet length-based protocols, e.g., the WiFi data rate switching frequency and the number of mixed data rate. The WiFi data rate is switched changed from 0 to 45 Hz and the data rate is changed from 10 to 54 Mbps. The number of mixed data rate is varied within 10 possible data rates with the fixed switch frequency 5 Hz. We conduct the above experiments in LoS scenario with SNR = 10 dB. Results are discussed in Section 6.2.

To validate the jamming effectiveness of JamCloak, we also apply JamCloak to attack above three existing CTC protocols (i.e., WiZig [19], Esense [3] and EMF [8]). We use the metric PDR (packet delivery ratio) to evaluate the jamming impact on CTC links. We use eight consecutive CTC bits "0" and "1," respectively, to indicate the beginning and end of the CTC packet. The CTC packet payload is set to eight bits (i.e., 24 bits total length), which is a common setting in coordination or events monitoring applications [35, 45]. There is no retransmission in the CTC link and a packet is labeled as corrupted once there is one erroneous CTC bit. We also compare JamCloak with an existing WiFi reactive jammer [37] (e.g., WiFiJam for short) in terms of the jamming gain [31].

Table 2. Classification Accuracy in LoS Scenario

	WiZig	Esense	EMF	EL+PL	PL+PR	EL+PR	EL+PL+PR	Normal
WiZig	0.95	0	0	0.01	0	0	0.01	0.01
Esense	0	0.97	0.01	0	0	0	0	0
EMF	0	0	0.96	0	0	0	0	0
EL+PL	0	0	0	0.92	0.01	0	0.03	0
PL+PR	0.02	0.03	0	0	0.91	0	0.05	0
EL+PR	0.01	0	0.03	0	0.02	0.92	0.01	0
EL+PL+PR	0.01	0	0	0.07	0.06	0.08	0.9	0
Normal	0.01	0	0	0	0	0	0	0.99

	WiZig	Esense	EMF	EL+PL	PL+PR	EL+PR	EL+PL+PR	Normal
WiZig	0.99	0	0	0	0	0	0	0
Esense	0	0.99	0	0	0	0	0	0
EMF	0	0	1	0	0	0	0	0
EL+PL	0	0	0	0.99	0	0.01	0	0
PL+PR	0	0.01	0	0	0.98	0	0.01	0
EL+PR	0.01	0	0	0	0	0.99	0	0
EL+PL+PR	0	0	0	0.01	0.02	0	0.99	0
Normal	0	0	0	0	0	0	0	1

Left: SNR = −2dB. Right: SNR = 10 dB.

Table 3. Classification Accuracy in NLoS Scenario

	WiZig	Esense	EMF	EL+PL	PL+PR	EL+PR	EL+PL+PR	Normal
WiZig	0.91	0	0	0	0	0	0	0
Esense	0	0.9	0	0	0.01	0	0	0.05
EMF	0	0.01	0.9	0	0.03	0	0.01	0.03
EL+PL	0.01	0	0	0.89	0	0.11	0.04	0
PL+PR	0.01	0	0.05	0	0.88	0	0.01	0.02
EL+PR	0	0	0.02	0	0.06	0.88	0.04	0
EL+PL+PR	0	0.09	0.03	0.11	0	0.01	0.87	0
Normal	0.07	0	0	0	0.02	0	0.03	0.9

	WiZig	Esense	EMF	EL+PL	PL+PR	EL+PR	EL+PL+PR	Normal
WiZig	0.94	0	0	0	0	0	0	0.03
Esense	0	0.94	0	0	0	0	0	0
EMF	0	0	0.95	0	0	0	0	0.02
EL+PL	0	0	0	0.92	0.03	0.04	0.02	0.01
PL+PR	0	0.06	0	0	0.92	0	0.04	0.01
EL+PR	0.04	0	0.02	0	0	0.92	0.03	0.03
EL+PL+PR	0	0	0.03	0.08	0.05	0.04	0.91	0
Normal	0.02	0	0	0	0	0	0.01	0.9

Left: SNR = −2dB. Right: SNR = 10 dB.

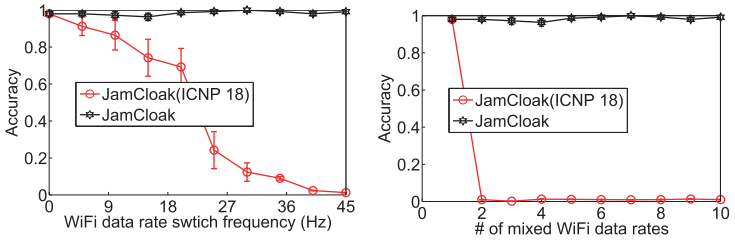
We select WiFiJam as the comparison target because of the following reasons. First, as we stated in Section 3 that JamCloak targets at one direction of the cross-technology communication link, i.e., from WiFi to ZigBee, therefore, JamCloak actually aims at analyzing WiFi traffic and then performs the attack. The most related works may be the jamming approaches that also aim at analyzing WiFi traffic, such as WiFiJam [37]. Considering the similarity of the functionality and the analysis target, WiFiJam is selected. Second, WiFiJam is an open-source work [38], which greatly reduces the overhead of reproducing the results and conducting the comprehensive evaluation.

The jamming gain is defined according to the existing work [31]: the inverse ratio of the time of jamming used to achieve a desired effect with the jammer under consideration to the time of jamming that is used to achieve the same effect with the constant jammer. Let t^c_{jam} and t^r_{jam} denote the jamming time for constant jammer and JamCloak, respectively. Then the jamming gain is defined as $10log_{10}t^c_{jam}/t^r_{jam}$. Note that the jamming gain is used to evaluate the relative jamming time of JamCloak compared with the constant jammer. Therefore, the design goal of JamCloak is to perform jamming with as little time as possible. It can reduce the probability of being detected, which is also stated in References [31, 43]. We let WiFiJam transmits a short burst once it observes WiFi activities. Results are discussed in Section 6.3.

We also evaluate impact of the distance on the jamming performance, considering various WiFi transmission power and the number of obstacle walls. In an about 30-m corridor, we utilize multiple 2m × 3m planks as walls to conduct the experiments. The WiFi transmission power varies from 5 to 20 dBm, and the number of obstacles are from one to three. The experiments are conducted with SNR = 10 dB. Results are shown in Section 6.4.

6.2 Classification Accuracy

We evaluate the CTC classification model of JamCloak in terms of the accuracy and the generality under different SNR. We apply our model to classify the traffic that contains both new CTC protocols and existing CTC protocols to validate the generality. As shown in the confusion matrix Tables 2 and 3, in both LoS and NLoS scenarios, JamCloak consistently achieves high classification accuracy (e.g., 94.7% on average for existing CTC protocols and 92.4% on average for new CTC protocols detailed in Section 4) for a wide SNR range (e.g., from −2 to 10 dB). We only show the



(a) Impact of rate switch frequency. (b) Impact of the number of mixed data rate.

Fig. 10. Impact of rate adaptation on detection accuracy.

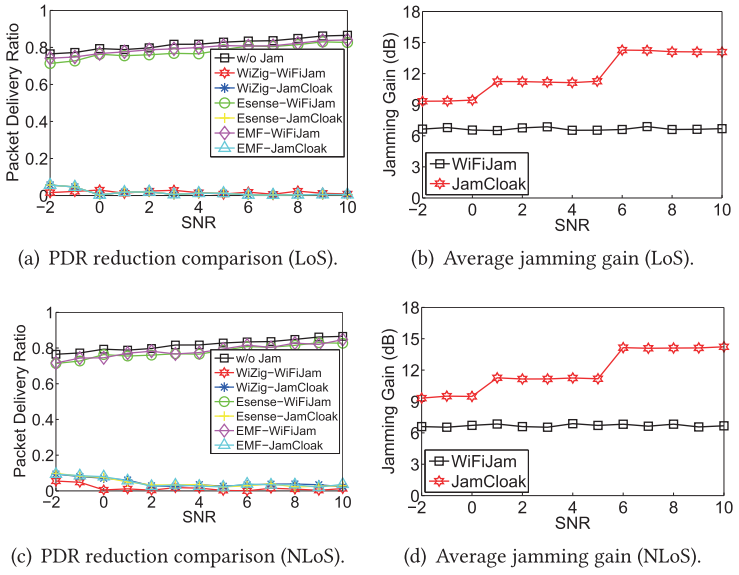
confusion matrix in SNR with −2 and 10 dB due to space limitation. We note that when the SNR is low (e.g., −2 dB), the False Positives (FP) in the CTC protocol classification associated with energy level are slightly increased by 4.7%. Because the low SNR makes the variation of the energy level larger, resulting in a more ambiguous boundary between the normal traffic and the energy level modulated traffic. The FP are thus increased. We also note that the False Negatives in NLoS scenario are larger than in LoS scenario (e.g., 2.4%), this might be less CTC features are captured in NLoS scenario than LoS scenario due to the wall, resulting in a slight reduction in accuracy.

Figure 10 presents the impact of rate adaptation on the detection accuracy of packet length-based protocols. As shown in Figure 10(a), JamCloak consistently achieves high accuracy (e.g., more than 95%) when the rate switch frequency varies from 0 to 45 Hz compared with JamCloak (ICNP 18). We can see that JamCloak achieves the most accuracy improvements (e.g., 97.2%) at larger switch frequency (e.g., 45 Hz), because it is highly possible that more than two models will be detected by JamCloak (ICNP 18) even under normal WiFi traffic, resulting in larger false positive for the detection of packet length-based protocols. Figure 10(b) shows that JamCloak works well even when there are ten mixed data rates, e.g., 98.3% accuracy on average. Note that the accuracy of JamCloak (ICNP 18) decreases rapidly (i.e., from 98.1% to 1.3%) when the number of mixed data rates is bigger than two because more than two models are detected, resulting in larger false detection of packet length-based protocols.

6.3 Proof-of-concept for Reactive Jamming Attack

Jamming packet-level CTC protocols that transmit from WiFi to ZigBee. We compare JamCloak with the existing reactive jammer [37], WiFiJam, in terms of the effects of jamming attack over three existing CTC protocols (i.e., WiZig [19], Esense [3], and EMF [8]) in different scenarios. As shown in Figure 11, in both LoS and NLoS scenarios, JamCloak significantly reduces the PDR by 80.8% on average. In the meantime, JamCloak's jamming gain is more than 1.78× higher than WiFiJam. We note that the jamming effects caused by WiFiJam to Esense and EMF are subtle (e.g., only 3.2% and 2.6% of PDR reduction on average, respectively). This is because the short bursts fail to significantly change the Esense modulated packet length, nor does it change the energy occupancy of the EMF modulated window. The reduction of PDR in LoS scenario is more than in NLoS scenario (e.g., 3.1% on average) for JamCloak, because with the absence of the wall, JamCloak can detect and jam more CTC packets. We also find that jamming gain of JamCloak increases with the SNR. Because JamCloak estimates the parameters of jamming signal more accurately at higher SNR, thus more efficient attacks are enabled using less jamming signals.

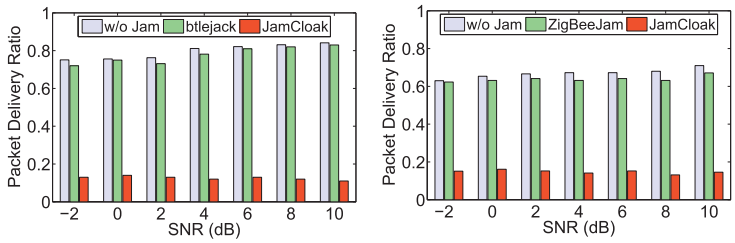
Jamming packet-level CTC protocols that transmit from ZigBee/BLE to WiFi. For the directions from ZigBee/BLE to WiFi, we utilize the basic theory of Esense, i.e., varying the packet length of ZigBee/BLE to transmit CTC information WiFi. At the WiFi side, it continuously samples the channel energy to detect potential CTC information. As shown in existing packet traces, there



(a) PDR reduction comparison (LoS). (b) Average jamming gain (LoS).

(c) PDR reduction comparison (NLoS). (d) Average jamming gain (NLoS).

Fig. 11. Proof-of-concept for jamming attack.



(a) Jamming CTC links from BLE to WiFi. (b) Jamming CTC links from ZigBee to WiFi.

Fig. 12. Performance of jamming CTC links over other directions (i.e., BLE/ZigBee to WiFi).

are also packet length bias in BLE [27] and ZigBee [34]. Therefore, it is possible to construct packet length set that is rarely seen in normal traffic to deliver CTC information. In JamCloak, we only need to add new training sets that labelled with the new CTC directions to obtain the classification model. The jamming parameters can be estimated similarly.

Figure 12 shows the results of performing jamming attacks on other directions (i.e., BLE/ZigBee to WiFi). We compare JamCloak with two existing jammers, i.e., the ZigBee jammer (ZigBeeJam for short) [41] and the BLE jammer btlejack [40]. The experiments are conducted in LoS scenario. Results show that for both directions from ZigBee or BLE, JamCloak can achieve better jamming performance than existing jammers in various environments.

Jamming PHY emulation-based CTC protocol. We also evaluate the performance of jamming the PHY emulation-based CTC protocol, WEBee [28], by JamCloak integrated with WiFiJam. Figure 13 shows that JamCloak can reduce the packet delivery ratio to lower than 10% in various environments. It indicates that JamCloak can successfully judge whether there are packet-level CTC protocols, and effectively switch to WiFiJam to perform the jamming attacks.

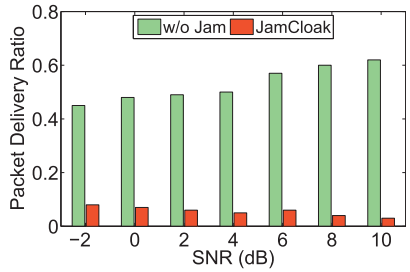
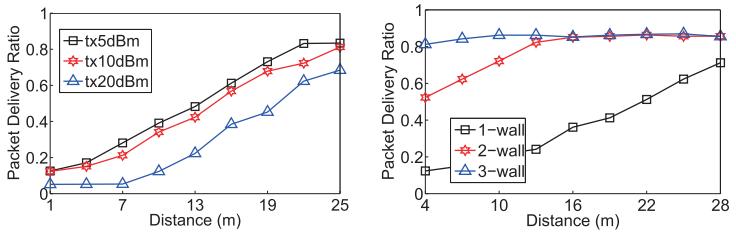


Fig. 13. Performance of jamming PHY emulation-based CTC protocol.



(a) Impact of WiFi transmission power (no wall). (b) Impact of the number of walls (transmission power=20dBm).

Fig. 14. Impact of distance on jamming performance.

Table 4. Reaction Delay

CTC Protocol	Packet In-air Time t_{packet}	Sampling Time t_{sample}	Detection Time t_{detect}	Jamming Time t_{jam}	Total Jamming
WiZig	155.9 ms	21 ms	<1ms	7~21 ms	28~42ms
Esense	1568.6 ms	100 ms	25 ms	-	125ms+pktlen
EMF	115.3 ms	21 ms	5 ms	5~15 ms	31~41ms

6.4 Impact of Distance on Jamming Performance

Figure 14 presents the impact of distance and the number of walls on the jamming performance. Figure 14(a) shows that JamCloak can achieve high jamming performance within 10 m when the WiFi transmission power is 20 dBm. We can see that the jamming efficiency decreases rapidly when the WiFi transmission power changes from 20 to 5 dBm, because JamCloak cannot detect CTC signals transmitted from the WiFi sender and jamming attacks are not triggered at all. Figure 14(b) presents that JamCloak can effectively corrupt jamming CTC packets with less than one wall within 6m. The received signal strength of WiFi drops sharply with more blocked walls. It causes JamCloak not to detect the weak CTC signals, and the effect of jamming is not obvious.

6.5 Reaction Delay for Performing Jamming Attack

We evaluate the reaction delay for performing reactive jamming in all scenarios and the results are averaged as shown in Table 4. We can find that for all existing three implemented CTC protocols, JamCloak can capture CTC packets and perform attacks in time (e.g., the total jamming time is lower than CTC packet on-air time t_{packet}). To perform a reactive jamming attack, JamCloak captures enough samples within the window size t_{sample} and then detects CTC activities (with the required time t_{detect}) and sends a short yet sufficient jamming burst (i.e., t_{jam}^{min}) to destroy the

packet, all while the CTC packet is being transmitted (i.e., t_{packet}). Therefore, JamCloak performs an effective reactive jamming by meeting the following time requirement:

$$t_{sample} + t_{detect} + t_{jam}^{min} \leq t_{packet}. \tag{7}$$

In Table 4, the packet on-air time is inferred based on the CTC packet, but not the WiFi packet. As stated in Section 6.1, we use eight consecutive CTC bits "0" and "1," respectively, to indicate the beginning and end of the CTC packet. The CTC packet payload is set to eight bits (i.e., 24 bits total CTC packet length), which is a common setting in the coordination coexistence or event monitoring applications. Then, the CTC packet on-air time is estimated according to the settings in Table 1. For example, WiZig utilizes two energy level (i.e., 3 and 12 dBm) to delivery one CTC bit. The CTC bit is repeated several times to achieve reliable CTC transmissions (e.g., $6 \sim 7$ times according to WiZig), which results in a roughly $6 \sim 7$ ms symbol duration time. Considering that the CTC packet is 24-bit long, the on-air time is roughly calculated as $(6 \sim 7) \times 24 = 144–168$ ms. The reported packet on-air time (i.e., 155.9ms) in Table 4 is averaged from real experiments and falls in this scope.

6.6 Impact of CTC Packet Size

In practical scenarios, JamCloak can achieve effective jamming attacks over a wide range of CTC packet size. As we have analyzed in Section 6.5, in the case of CTC packet size of 24 bits, JamCloak meets the time constraint and achieves effective reactive jamming attacks. When the CTC packet size is reduced to eight bits, JamCloak is still able to achieve an effective attack. For example, the CTC packet on-air time of WiZig, Esense, and EMF becomes 51.9, 522.8, and 38.4 ms, respectively, which is larger than the total jamming time of JamCloak (e.g., $28 \sim 42$, 125, and $31 \sim 41$ ms, respectively). However, such a short packet size (including the packet header and tail) is challenging to be used in practice, like the transmission of temperature information in smart home monitoring applications [3].

7 MITIGATION APPROACH

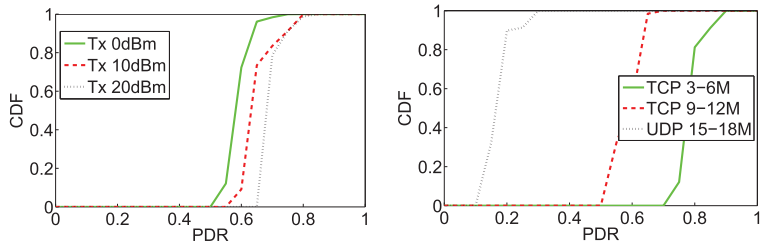
To counteract reactive jamming systems like JamCloak, we discuss a practical countermeasure that involves in jamming detection and mitigation. Our countermeasure will not introduce additional overhead when there is no jamming. Because our countermeasure switches to the anti-jamming mode only when jamming attacks or strong interferences are detected.

7.1 Reactive Jamming Detection

Existing detection approach can be classified into two types: the physical layer approach [30, 39, 43] and the MAC layer approach [31, 33]. The classification is based on the implementation layer of the reactive jamming as we detailed in Section 8.

Note that JamCloak is the physical layer reactive jamming attack, because it performs the jamming attacks at the level of modulation type (e.g., find the optimal jamming signal patterns) without considering the context of transmission packets. We thus only consider the detection approach against physical layer reactive jamming.

Consistency check detection approach [30, 39, 43] has been widely used to detect reactive jamming attacks. There are two type of consistency checks against reactive jamming: the signal strength consistency check and the location consistency check. The basic idea of above two checks is finding the inconsistency between the measured PDR and the target metrics (e.g., the received signal strength or the location). For example, if we measure low PDR and high received packet signal strength or small distances to the neighbors, then it is most likely that the node is jammed. However, above two metrics are ineffective to capture the reactive jamming behaviors



(a) Fixed traffic density (3M) with varying Tx power.

(b) Fixed Tx power (20 dBm) with varying traffic density.

Fig. 15. Jamming detection metrics comparison.

over CTC links, because the PDR of CTC is more sensitive to the background traffic than the RSSI or the location. We conduct experiments to validate this observation as shown in Figure 15. The distance of the WiFi transmitter and the ZigBee receiver is set to 5 m. The background traffic is generated using the tool iperf [12].

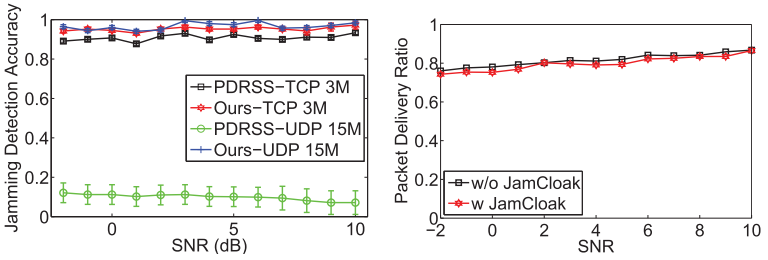
We generate 3 Mbps TCP traffic for obtaining the results in Figure 15(a), and set the transmission power of WiFi to 20 dBm for obtaining the results in Figure 15(b). We can find that under dense background traffic, it is normal that the PDR is relative low even under high transmission power scenarios, causing high false positive ratio to existing jamming detection approaches (i.e., the ratio of determining the normal traffic as jamming attacks).

We thus propose a new metric considering both RSSI and background traffic Signal-to-Traffic-Ratio (STR). Note that other metrics like involving in the distance and the background traffic can be derived similar with the STR. Given the observation window W, we let S denote the average RSSI during the observation window. The ZigBee receiver samples the background traffic density when there is no CTC traffic. We detect WiFi packets using RSSI-based approach and calculate the WiFi packets occupancy ratio T as the background traffic density. The metric STR can be expressed as $STR = \frac{S}{T}$. We separate the states of the CTC transceivers into (1) Normal, (2) Jammed, and (3) BadChannel. Given the STR metric threshold TH_{str} and the PDR threshold TH_{pdr}, the above states can be determined by verifying the rules correspondingly: (1) $STR > TH_{str}, PDR \geq TH_{pdr}$; (2) $STR > TH_{str}, PDR < TH_{pdr}$; and (3) $STR < TH_{str}$. We use the following rules to determine whether the CTC transceivers are currently under jamming attacks: It is normal for the condition (1), and is under jammed for the condition (2). For the condition (3), it is the bad channel.

The relatively larger STR means the channel quality is high and the ZigBee receiver should see a high PDR (i.e., denoted as the Normal state). Otherwise, the ZigBee receiver is under the reactive jamming attack (i.e., denoted as the Jammed state). As for the scenario that the STR is relatively lower, we denote it as the BadChannel state. When the Zigbee receiver detects the current state as non-Normal, it switches to the reactive jamming mitigation mode. Note that for the BadChannel state, we also let the ZigBee receiver try to improve its performance by using the mitigation approach.

7.2 Reactive Jamming Mitigation

To mitigate the strong interference caused by reactive jamming attacks like JamCloak or high density background traffic, we consider using channel hopping approach to evade the jammed or bad channel [31]. Assuming JamCloak can only sample the CTC pattern for a WiFi channel on which its ratio is listening, then the candidate hopping channel sequence is WiFi channel $\{1, 6, 11\}$, and the ZigBee receiver will hop to any one of the overlapped ZigBee channel. To avoid the predictability of this hopping pattern, we use a pre-shared secret between the CTC transceiver.



(a) Jamming detection accuracy. (b) Effectiveness of mitigation.

Fig. 16. Jamming mitigation approach evaluation.

7.3 Evaluation of Mitigation Approach

We evaluate the performance of the proposed countermeasure in terms of the reactive jamming detection accuracy and the improvements of PDR. The setup of this experiment is similar with the previous experiments in Section 6. We only shows the results of LoS scenario due to space limit. Figure 16(a) shows the results of the reactive jamming detection accuracy between the existing signal strength consistency check-based approach PDRSS [30] and our approach. Results show that our detection approach consistently improves the accuracy by 91.2% on average than the existing approach under most scenarios. As for mitigation efficiency, we average the results over all traffic density settings as shown in Figure 16(b). We can find that our proposed countermeasure can effectively decrease the reduction of PDR by 1.7%.

8 RELATED WORKS

Jamming attack. In general, there are two kind of elementary jamming attack: proactive and reactive [31]. Proactive jammer continuously sends random bits or electromagnetic energy on the channel (e.g., constant jammer [31]). To save energy, proactive jammer can also emit signals periodically (e.g., random jammer [31]). Reactive jammer starts jamming only when it observes a network activity occurred. The reactive jammer can be further classified into two sub-class based on the implemented layer: the link layer based (deceptive jammer) and the physical layer based. For the link layer-based jammer, the jammer can decode the information of received packet and only jam the valuable packets like ACK to cause the additional retransmissions [43]. As for the physical layer-based jammer, the jammer decide to jam the channel based the on sampled RSSI (e.g., higher than a threshold) [39]. It is challenging to detect reactive jamming [31], because only limited interference with other nodes is experienced, which minimizes the risk of exposure [41]. However, achieving reactive jamming attack over CTC links is challenging due to the totally different modulation scheme. In this article, we first propose a taxonomy of the existing CTC protocols. Then based on the taxonomy, we extract essential features to train a CTC classification and detection model, and estimate the parameters that can be used to efficiently jam CTC links.

Jamming attack detection. Michael et al. [33] propose a link layer approach to detect reactive jamming attacks over DSSS-based wireless systems. They take advantage of the fact that the first few jamming-free bits are known *a priori* and thus can be used to detect jamming attacks. But, current CTC systems are not DSSS based and the above approach is not applicable. Xu et al. [43] proposed a consistency checks-based approach to detect reactive jamming attacks, and it is further improved in References [30, 39]. It is a PHY-layer-based approach and the basic idea is to find the inconsistency between the measured PDR and the target metrics (e.g., the received signal strength or the location). However, above two target metrics are both not enough to capture the reactive jamming behaviors over CTC links, because the PDR of CTC is sensitive to the background traffic

but not only the RSSI or the location. In this article, we propose a new metric considering both received signal strength and traffic density to improve reactive jamming detection accuracy.

Jamming attack mitigation. Many jamming attack mitigation approach has been proposed [11, 23, 30, 31, 42]. Frequency-Hopping-based approach [23, 30] hops to another channel when jamming attacks are detected. Coding-based approach [31] improves the resilience of jamming by employing forward error correction code. mobile agent-based approach [42] explores the unjammed area and then designs a new routing path to improve network level performance. Convert channel-based approach [11] leverages the packet arrival time to transmit information when under jamming attacks. Note that CTC is a kind of the convert channel and JamCloak targets at this communication links. Therefore, this kind of approach cannot be applied.

9 CONCLUSION

This article presents JamCloak, the first reactive jamming system that can attack most existing CTC protocols. We propose a taxonomy of the existing CTC protocols. Then based on the taxonomy, we extract essential features to train a CTC detection model, and estimate the parameters to efficiently jam CTC links. Extensive results show that JamCloak can significantly reduce the PDR by 80.8% on average in practical environments. In the meantime, JamCloak's jamming gain is more than 1.78× higher than the existing reactive jamming attack. In addition, we propose a practical countermeasure against reactive jamming attack over CTC links like JamCloak. Results show that our approach consistently improves the jamming detection accuracy by 91.2% on average than existing approach, and effectively decreases the reduction in packet delivery ratio to 1.7%.

REFERENCES

[1] Wahhab A. Jun Huang and Guoliang Xing. 2016. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'16)*.

[2] Michael R. Berthold and Frank Höppner. 2016. On clustering time series using euclidean distance and pearson correlation. *arXiv* abs/1601.02213. Retrieved from https://arxiv.org/abs/1601.02213.

[3] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *Proceedings of the ACM Mobile Computing and Networking Conference (MOBICOM'09)*.

[4] Gonglong Chen and Wei Dong. 2018. JamCloak: Reactive jamming attack over cross-technology communication links. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP'18)*.

[5] Gonglong Chen, Wei Dong, Zhiwei Zhao, and Tao Gu. 2017. Towards accurate corruption estimation in zigbee under cross-technology interference. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS'17)*.

[6] G. Chen, W. Dong, Z. Zhao, and T. Gu. 2019. Accurate corruption estimation in zigbee under cross-technology interference. *IEEE Trans. Mobile Comput.* 18, 10 (2019), 2243–2256.

[7] Yongrui Chen and Zhijun Li. 2018. TwinBee: Reliable physical-layer cross-technology communication with symbol-level coding. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*.

[8] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*.

[9] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, and etc. 2016. B2W2: N-way concurrent communication for iot devices. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys'16)*.

[10] R. D. and P. Hart. 2001. *Pattern Classification*. Wiley Interscience.

[11] S. Doro, L. Galluccio, G. Morabito, et al. 2015. Defeating jamming with the power of silence: A game-theoretic analysis. *IEEE Trans. Wireless Commun.* 5, 14 (2015), 2337–2352.

[12] ESnet and Lawrence Berkeley National Laboratory. 2020. iperf. Retrieved from https://iperf.fr/.

[13] Andrea G. 2005. *Wireless Communications*. Cambridge University Press.

[14] Kanika G., Alvin L., and Qing Y. 2014. Jamming and anti-jamming techniques in wireless networks: A survey. *Int. J. Ad Hoc Ubiq. Comput.* 17, 4 (2014), 197–215.

[15] Gartner. 2017. Gartner news. Retrieved from http://www.gartner.com/newsroom/id/3598917.

[16] W. L. W. Group. 2012. IEEE standard part 11: Wireless lan mac and phy specifications. In *IEEE Std 802.11-2012*.

[17] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. WIDE: Physical-level CTC via digital emulation. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN'19)*.
[18] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali. 2020. ZigFi: Harnessing channel state information for cross-technology communication. *IEEE/ACM Trans. Netw.* 28, 1 (2020), 301–311.
[19] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. WiZig: Cross-technology energy communication over a noisy channel. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*.
[20] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau. 2010. Automatic modulation recognition using wavelet transform and neural networks in wireless systems. *EURASIP J. Adv. Signal Process* 2010, Article 42 (2010), 42:1–42:13 pages.
[21] J. Huang, G. Xing, G. Zhou, and R. Zhou. 2010. Beyond co-existence: Exploiting wifi white space for Zigbee performance assurance. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP'10)*.
[22] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua R. Smith. 2016. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the ACM Special Interest Group on Data Communication Conference (SIGCOMM'16)*.
[23] Heo Jeongyoon, Kim Jung-Jun, and etc. 2017. Dodge-Jam: Anti-jamming technique for low-power and lossy wireless networks. In *Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON'17)*.
[24] Wenchao Jiang, Zhimeng Yin, Song Mim Kim, and Tian He. 2017. Transparent cross-technology communication over data traffic. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*.
[25] Wenchao Jiang, Zhimeng Yin, Ruofeng Liu, Zhijun Li, Song Min Kim, and Tian He. 2017. BlueBee: a 10,000x faster cross-technology communication via PHY emulation. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys'17)*.
[26] Muhammad O. Khan, Lili Qiu, Apurv Bhartia, and Kate Ching-Ju Lin. 2015. Smart retransmission and rate adaptation in wifi. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP'15)*.
[27] Long Vu Klara Nahrstedt. 2020. CRAWDAD dataset uiuc/uim (v. 20120124). Retrieved from https://crawdad.org/uiuc/uim/20120124.
[28] Zhijun Li and Tian He. 2017. WEBee: Physical-layer cross-technology communication via emulation. In *Proceedings of the ACM Mobile Computing and Networking Conference (MOBICOM'17)*.
[29] Zhijun Li and Tian He. 2018. LongBee: Enabling long-range cross-technology communication. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*.
[30] Donggang Liu, Joshua Raymer, and Andy Fox. 2012. Efficient and timely jamming detection in wireless sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Smart Systems (MASS'12)*.
[31] Konstantinos Pelechrinis, M. Iliofotou, and Srikanth V. Krishnamurthy. 2011. Denial of service attacks in wireless networks: The case of jammers. *IEEE Comm. Surv. Tutor.* 13, 2 (2011), 245–257.
[32] Ettus Research. 2019. USRP N210 data sheet. Retrieved from https://www.ettus.com/all-products/un210-kit/.
[33] S. Michael, G. Domenico, L. Vincent, W. Matthias, and B. S. Jens 2014. Detection of reactive jamming in DSSS-based wireless communications. *IEEE Trans. Wireless Commun.* 13, 3 (2014), 1593–1603.
[34] Wireless Shark. 2020. Packet traces: IPv6 and 6LoWPAN over IEEE 802.15.4. Retrieved from https://wiki.wireshark.org/SampleCaptures.
[35] Min-Kim Song and He Tian. 2015. FreeBee: Cross-technology communication via free side-channel. In *Proceedings of the ACM Mobile Computing and Networking Conference (MOBICOM'15)*.
[36] J. P. van Brakel. 2020. Z-score: peak signal detection in realtime timeseries data. Retrieved from https://stackoverflow.com/ questions/22583391/peak-signal-detection-in-realtime-timeseriesdata/22640362.
[37] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of the ACM Annual Computer Security Applications Conference (ACSAC'14)*.
[38] Mathy Vanhoef and Frank Piessens. 2014. WiFiJam code. Retrieved from https://github.com/vanhoefm/modwifi.
[39] K. P. Vijayakumar, P. Ganeshkumar, and M. Anandaraj. 2015. A novel jamming detection technique for wireless sensor networks. *KSII Trans. Internet Inf. Syst.* 9, 10 (2015), 4223–4249.
[40] virtualabs. 2020. BtleJack: A new Bluetooth Low Energy swiss-army knife. Retrieved from https://github.com/virtualabs/btlejack.
[41] M. Wilhel, I. Martinovic, Jens B. Schmitt, and V. Lenders. 2011. Reactive jamming in wireless networks: How realistic is the threat?. In *Proceedings of the ACM Wireless Network Security*.
[42] A. D. Wood, J. A. Stankovic, and S. H. Son. 2003. JAM: A jammed-area mapping service for sensor networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS'03)*.
[43] W. Xu, W. Trappe, Y. Zhang, and T. Wood. 2005. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05)*.

[44] Z. Yin, W. Jiang, S. Min Kim, and T. He. 2017. C-Morse: Cross-technology communication with transparent morse coding. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*.

[45] Xinyu Zhang and Kang G. Shin. 2013. Gap Sense: Lightweight coordination of heterogeneous wireless devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13)*.

[46] Y. Zhang and Q. Li. 2013. HoWiES: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM'13)*.

[47] Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. 2014. ZiSense: Towards interference resilient duty cycling in wireless sensor networks. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'14)*.

[48] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. 2010. ZiFi: Wireless LAN discovery via zigbee interference signatures. In *Proceedings of the ACM Mobile Computing and Networking Conference (MOBICOM'10)*.

Received December 2019; revised August 2020; accepted August 2020