2018 IEEE 26th International Conference on Network Protocols

JamCloak: Reactive Jamming Attack over Cross-Technology Communication Links

Gonglong Chen and Wei Dong*
College of Computer Science, Zhejiang University.
Email: {chengl, dongw}@emnets.org

***Abstract*—Recently, CTC (Cross-Technology Communication), 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 paper, 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 LoS (Line-of-Sight) and NLoS (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 attack 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%.**

I. 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 [1]. These IoT devices are envisioned to employ highly heterogeneous wireless technologies such as WiFi, ZigBee, Bluetooth, etc., causing difficulties in directly interconnecting these devices due to completely different PHY-layer 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 [2–14]. Previous studies have demonstrated the promise of CTC in many IoT applications. For example, using CTC, the Cross-Technology Interference (CTI) problem can be much more effectively solved by sharing the spectrum in a TDMA fashion [2]. Or achieving efficient concurrent transmissions for different wireless protocols [11]. Or enabling real-time patient monitoring by combining the ubiquitous deployed WiFi with the low power Bluetooth (BLE) [6]. CTC, like other wireless communications, could suffer from malicious attacks such as jamming [15] or sniffing [16]. 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 to 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 [17], 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 [17]. 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 [15]. In this paper, we are interested in the following two questions: *(1) Can we design a powerful and generic reactive jamming attack system over CTC links? (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 into CTC due to the totally different modulation schemes. For example, FreeBee [6] 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 [18]) 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 [12]

*Corresponding author. This work is supported by the National Science Foundation of China under Grant No. 61772465 and No. 61472360, Alibaba Zhejiang University Joint Institute of Frontier Technologies, Zhejiang Provincial Key Research and Development Program (No. 2017C02044), and the Fundamental Research Funds for the Central Universities (No. 2017FZA5013).

978-1-5386-6043-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICNP.2018.00014

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 [2]) due to the fixed energy level mapping relationship. FreeBee [6] 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 a 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 challenge to devise a generic reactive jamming system against most CTC protocols.

To address 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 the 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 which 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 [7, 12], packet length based protocol [2, 5] and packet reorder based protocol [6, 9–11]. Then based on the taxonomy, the features of each category are extracted according to the observation: the existing CTC protocols constructs the energy characteristics with a large difference in normal traffic, guaranteeing to ensure the CTC information can be demodulated efficiently. For example, Esense [2] 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 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. 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) can not 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 in extra overhead because we perform the mitigation approach only when jamming attacks are detected.

We summarize the contributions of this work as follows:

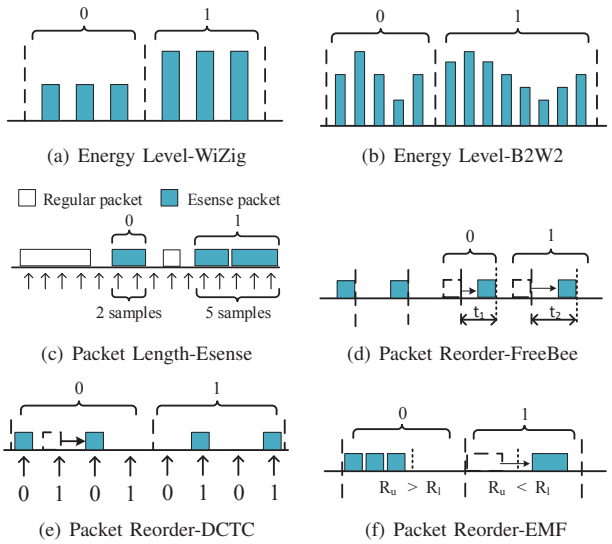


Fig. 1: Taxonomy of existing CTC protocols.

- 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 LoS (Line-of-Sight) and NLoS (Non-Line-of-Sight) 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 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%.

The rest of this paper is organized as follows. Section II presents the taxonomy of the existing CTC approach. Section III gives an overview of JamCloak. Section IV and Section V show the key component of JamCloak. Section VI presents the evaluation results. Section VII discusses a practical countermeasure against reactive jamming attack over CTC links . Section VIII introduces the related work and finally, Section IX concludes this paper.

II. 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 [7, 12], packet length based [2, 5] and packet reorder based [6, 9–11].

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 [12] 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 Fig. 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 [7] encodes CTC bits by directly adjusting the adjacent packets' transmission power to form a sine wave as shown in Fig. 1(b). CTC bits "0" and "1" are distinguished by changing the frequency of the sine wave.

Packet length based protocol. According to the existing work [2], 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 bytes 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 [2] and HoWIES [5] 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 [19]), while following the current IEEE 802.11 standard [9].

Packet reorder based protocol. Under 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) 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 [6] modulates one CTC bit by shifting the timing of periodic beacon frames as shown in Fig. 1(d). CTC bits "0" and "1" are distinguished by changing the shifting time. DCTC [10] 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 Fig. 1(e). EMF [11] modulates CTC bits by shifting the packet order to form a unique pattern. Specifically, as shown in Fig. 1(f), within the two synchronized time window, the left part with larger packet occupancy ratio denotes CTC bit "0" and vice visa.

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

Recently, many high throughput CTC protocols have been proposed [13, 14, 20, 21]. 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. However, these high throughput CTC protocols just reuse the existing wireless protocols and therefore existing jamming techniques can effectively detect and jam CTC signals such as WiFiJam [18].

III. Challenges and System Overview

A. 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 [15], 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 without the knowledge of the victims. For example, the number of high-frequency packet length of the normal traffic is stable according to prior works [2], we can thus extract this feature to detect Esense [2].

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 [10], a possible way is to extract the periodic time points using the fast folding algorithm [6]. However, since EMF [11] 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.

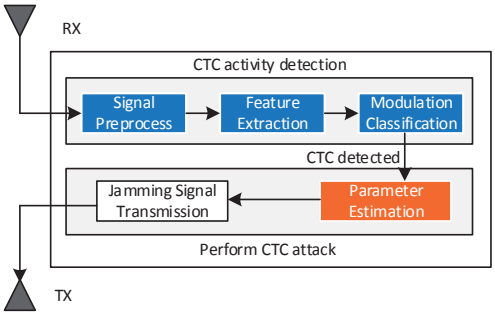


Fig. 2: Overview of JamCloak.

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.

B. 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 Fig. 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.

IV. CTC Activity Detection

To perform a reactive jamming attack over CTC links, determining whether the CTC is active 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.

A. Signal Sampling and Preprocessing

We utilize USRP N210 to sample RSSI series at high frequency about 1MHz [22]. 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:

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 10us and DIFS is 50us or 28us [19]), 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:

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

Where I_n denotes the noise floor. For a given transmission, the maximum amplitude within a packet is constant [23], we thus extract the maximum RSSI value within a packet to denote the packet level energy. Then the energy of i-th packet can be denoted as $max(I[B[i]], I[E[i]])$. The sets of WiFi packet level energy are:

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

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 50μs according to 802.11 standard [19].

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:

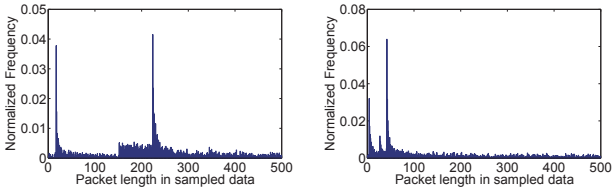
$$ST = \{E[i] - B[i-1] | 1 \leq i \leq len(B)\} \tag{4}$$

B. Feature Extraction

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

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 [24]. 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: $FE = \overline{SE}$.

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 will vary depending on the WiFi data rates as shown in Fig. 3. 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 1500 bytes packet changes from 220 samples to 40 samples



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

Fig. 3: Impact of WiFi data rates on packet length distribution.

Algorithm 1: Estimating the number of models

Input : Packet length sequence SL[], normalized histogram of packet length nh[].
Output: The number of models.

```
noiseSet = [];
for i=1: i < len(nh): i++ do
    if nh[i] ≤ noiseThd then
        noiseSet.append(i);
filter the packet length in noiseSet from SL;
SL = sort(SL);
find best k-means cluster set C using Elbow method;
find max number of elements MaxC in cluster set C;
find min number of elements MinC in cluster set C;
range = MaxC - MinC;
for i=0 ; i < len(C) ; i++ do
    isBorderC = len(C[i])/range < borderThd;
    drop the cluster C[i] from C if isBorderC is true;
return k=len(C);
```

when WiFi data rates changes from 54Mbps to 300Mpbs). The deviations from normal traffic are thus challenging to be used for distinguishing packet length based protocol.

We make the number of models in the packet length distribution as the feature to identify the packet length based protocols. 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 [2] considering different WiFi data rates. In other words, if the number of models is greater than two then we determine current traffic is modulated by packet length based protocols.

Based on observation, we utilize the clustering analysis based approach to estimate the number of models in the multi-modal distribution. The algorithm of estimating the number of models is shown in Algorithm 1. We thus let the number of estimated models to be the feature for distinguishing packet length based protocols: $FL = k$.

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 [25], which cannot be fitted by the modulated CTC traffic.

Then we use the K-S test (Kolmogorov-Smirnov 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: $FT = P/W$.

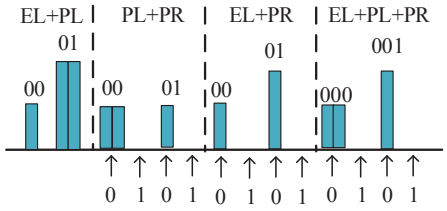


Fig. 4: New CTC protocols.

C. Classification Model Construction

After extracting all essential features for each CTC protocol, we then construct the classification model using C4.5 algorithm [26].

To validate the generality of our model, we also design four new CTC protocols based on the existing CTC protocols. As shown in Fig. 4, 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 [9]). (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 [26]. In Section VI, 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.

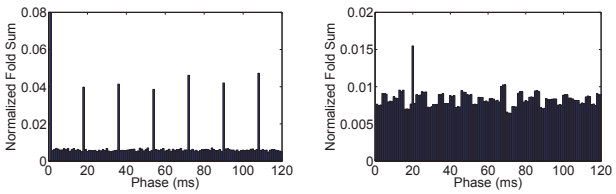
V. CTC Modulation Parameters Estimation

To maximize the jamming gain [15], 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.

A. Estimating Parameters of Energy Level Modulation

We assume that the possible modulated energy level patterns are known (e.g., square wave for WiZig [12] and sine wave for B2W2 [7]), 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



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

Fig. 5: Normalized folding sum histogram.

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

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

```
for i=1: i < len(phaseh): i++ do
    if phaseh[i]/sum(phaseh) ≤ noiseThd then
        phaseh[i] = 0; /*filter noise phase.*/
OneD = [];
for i=0; i < len(phaseh); i++ do
    /*generate 1 × phaseh[i] data for each phase.*/
    OneD.append(repmat(i, 1, phaseh[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 as
       Algorithm 1. Collect the number of rest elements into
       SumE.*/
return AveE = SumE/len(CT), CT;
```

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 minimize the jamming signal duration, we select the shortest segmented signal $S_g[min]$ from k sets and map the RSSI value in $S_g[min]$ to transmission power. The mapping function is obtained empirically.

B. Estimating Parameters of Packet Length Modulation

Based on the obtained cluster set C from Section IV-B2, 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 IV-A2. 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.

C. Estimating Parameters of Packet Reorder Modulation

We find that the folding peaks perform different distribution between DCTC [10] or FreeBee [6] and EMF [11]. As shown in Fig. 5, 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., 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 [11]. 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 $H = \{ps_1, ps_2, ..., ps_{len(CT)}\}$. 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.

D. 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.

VI. Evaluation

A. Experimental Methodology

We implement a prototype of JamCloak on the USRP N210/GNURadio platform [22] due to its wide range of transmission power and high RSSI sampling rate (e.g., 1MHz). 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 Fig. 6. We use the USRP platform to transmit WiFi packets following

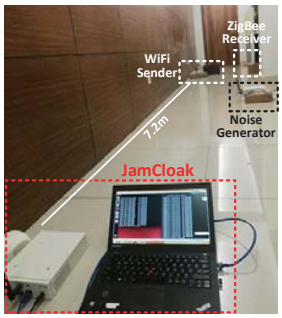


Fig. 6: Line-of-Sight

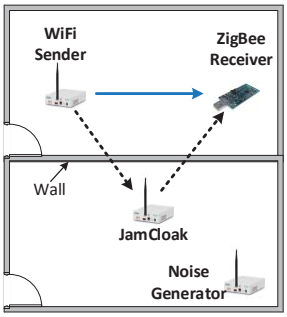


Fig. 7: None-Line-of-Sight.

IEEE 802.11 standards to the TelosB node. The WiFi data rate is set to 54Mbps 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., -2dB $\sim$ 10dB), we use another USRP [22] to generate the Gaussian noise with different power. Two scenarios are evaluated as follows:

Line-of-Sight (LoS): as shown in Fig. 6, the LoS scenario is a hallway which is 10.2m long and 1.2m wide. We fix the distance between JamCloak and the CTC transceiver pair to 7.2m, and the noise generator is placed between them.

Non-Line-of-Sight (NLoS): as shown in Fig. 7, in this scenario, the jammer and the CTC transceiver pair are deployed in two adjacent rooms which 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 [12], Esense [2] and EMF [11]) that represent three categories, and four new CTC protocols as described in Section IV. 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., 1500 bytes). Results are discussed in Section VI-B.

To validate the jamming effectiveness of JamCloak, we also apply JamCloak to attack above three existing CTC protocols (i.e., WiZig [12], Esense [2] and EMF [11]). 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 [4, 6]. 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 [18] (e.g., WiFiJam for short) in terms of the jamming gain [15]. The jamming gain is defined according to the existing work [15]: the inverse ratio of the time of jamming used to achieve a desired effect with the jammer under consideration

Table 1: Modulation parameter settings.

CTC Protocol	Energy Level	Packet Length	Critical Time Interval	Symbol Duration
WiZig	3dBm, 12dBm	normal	-	7ms
Esense	10dBm	2 pkts, 3 pkts	-	-
EMF	10dBm	normal	-	5ms
EL+PL	3dBm, 12dBm	2 pkts, 3 pkts	-	7ms
PL+PR	10dBm	2 pkts, 3 pkts	2ms	12ms
EL+PR	3dBm, 12dBm	normal	2ms	12ms
EL+PL+PR	3dBm, 12dBm	2 pkts, 3 pkts	2ms	12ms

Table 2: Classification accuracy in LoS scenario. Left:SNR= -2dB, Right:SNR=10dB

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

0.99	0	0	0	0	0	0	0
0	0.99	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0.99	0	0.01	0	0
0	0.01	0	0	0.98	0	0.01	0
0.01	0	0	0	0	0.99	0	0
0	0	0	0.01	0.02	0	0.99	0
0	0	0	0	0	0	0	1

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 $10\log_{10}(t^c_{jam}/t^r_{jam})$. We let WiFiJam transmits a short burst once it observes WiFi activities. Results are discussed in Section VI-C.

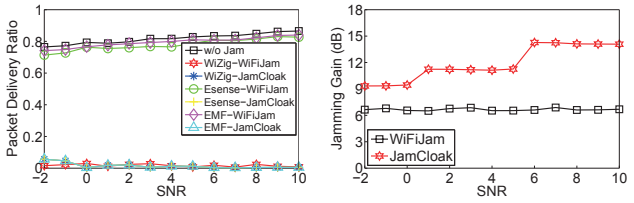
B. Classification Accuracy

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 Table 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 IV) for a wide SNR range (e.g., from -2dB to 10dB). We only show the confusion matrix in SNR with -2dB and 10dB due to space limitation. We note that when the SNR is low (e.g., -2dB), the FP (False Positives) 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 FN (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.

Table 3: Classification accuracy in NLoS scenario. Left:SNR= -2dB, Right:SNR=10dB

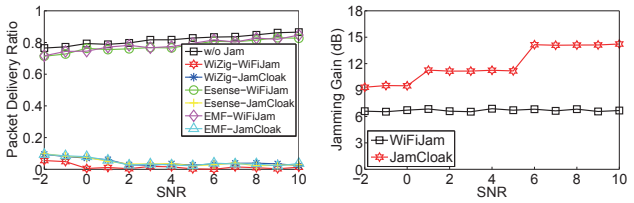
WiZig	0.91	0	0	0	0	0	0	0
Esense	0	0.9	0	0	0.01	0	0	0.05
EMF	0	0.01	0.9	0	0.03	0	0.01	0.03
EL+PL	0.01	0	0	0.89	0	0.11	0.04	0
PL+PR	0.01	0	0.05	0	0.88	0	0.01	0.02
EL+PR	0	0	0.02	0	0.06	0.88	0.04	0
EL+PL+PR	0	0.09	0.03	0.11	0	0.01	0.87	0
Normal	0.07	0	0	0	0.02	0	0.03	0.9

0.94	0	0	0	0	0	0	0.03
0	0.94	0	0	0	0	0	0
0	0	0.95	0	0	0	0	0.02
0	0	0	0.92	0.03	0.04	0.02	0.01
0	0.06	0	0	0.92	0	0.04	0.01
0.04	0	0.02	0	0	0.92	0.03	0.03
0	0	0.03	0.08	0.05	0.04	0.91	0
0.02	0	0	0	0	0	0.01	0.9



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

Fig. 8: Proof-of-concept for jamming attack in LoS.



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

Fig. 9: Proof-of-concept for jamming attack in NLoS.

C. Proof-of-concept for Reactive Jamming Attack

We compare JamCloak with the existing reactive jammer [18], WiFiJam, in terms of the effects of jamming attack over three existing CTC protocols (i.e., WiZig [12], Esense [2] and EMF [11]) in different scenarios. As shown in Fig. 8 and Fig. 9, 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.

D. 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}, 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: $t_{sample} + t_{detect} + t_{jam}^{min} \leq t_{packet}$.

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

E. 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 VI-D, 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 51.9ms, 522.8ms and 38.4ms, respectively, which is larger than the total jamming time of JamCloak (e.g, 28∼42ms, 125ms and 31∼41ms, 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 [2].

VII. Mitigation Approach

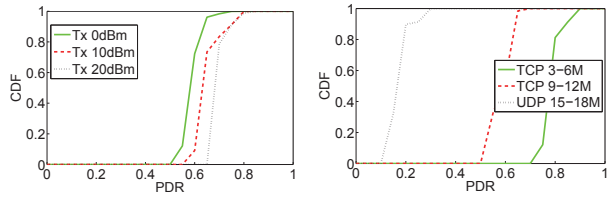
To counteract reactive jamming systems like JamCloak, we discuss a practical countermeasure that involves in jamming detection and mitigation. Our countermeasure will not introduce additional overhead when there is no jamming. Because our countermeasure switches to the anti-jamming mode only when jamming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

A. Reactive Jamming Detection

Existing detection approach can be classified into two types: the physical layer approach [27–29] and the MAC layer approach [15, 30]. The classification is based on the implementation layer of the reactive jamming as we detailed in Section VIII.

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.

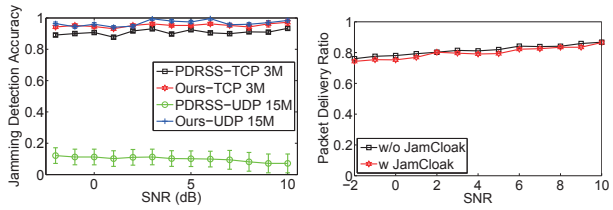
Consistency check detection approach [27–29] has been widely used to detect reactive jamming attacks. There are two type of consistency checks against reactive jamming: the signal strength consistency check and the location consistency check. The basic idea of above two checks is finding the inconsistency 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 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 Fig. 10. The distance



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

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

Fig. 10: Jamming detection metrics comparison.



(a) Jamming detection accuracy.

(b) Effectiveness of mitigation.

Fig. 11: Jamming mitigation approach evaluation.

of the WiFi transmitter and the ZigBee receiver is set to 5m. The background traffic is generated using the tool iperf [31].

We generate 3Mbps TCP traffic for obtaining the results in Fig. 10(a), and set the transmission power of WiFi to 20dBm for obtaining the results in Fig. 10(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 STR (Signal-to-Traffic-Ratio). 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) $STR < TH_{str}$. 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.

B. 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 [15]. 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.

C. 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 VI. We only shows the results of LoS scenario due to space limit. Fig. 11(a) shows the results of the reactive jamming detection accuracy between the existing signal strength consistency check based approach PDRSS [29] 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 Fig. 11(b). We can find that our proposed countermeasure can effectively decrease the reduction of PDR by 1.7%.

VIII. Related Works

Jamming attack. In general, there are two kind of elementary jamming attack: proactive and reactive [15]. Proactive jammer continuously sends random bits or electromagnetic energy on the channel (e.g., constant jammer [15]). To save energy, proactive jammer can also emit signals periodically (e.g., random jammer [15]). 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 [27]. 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) [28]. It is challenging to detect reactive jamming [15], because only limited interference with other nodes is experienced, which minimizes the risk of exposure [32]. However, achieving reactive jamming attack over CTC links is challenging due to the totally different modulation scheme. In this paper, 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.* [30] 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.* [27] proposed a consistency checks based approach to detect reactive jamming attacks, and it is further improved in [28, 29]. 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 paper, 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 [15, 29, 33–35]. Frequency-Hopping based approach [29, 33] hops to another channel when jamming attacks are detected. Coding based approach [15] improves the resilience of jamming by employing forward error correction code. mobile agent based approach [34] explores the unjammed area and then designs a new routing path to improve network level performance. Convert channel based approach [35] 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 can not be applied.

IX. Conclusion

This paper 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] Gartner, "Gartner news," in *http://www.gartner.com/newsroom/id/3598917*.

[2] K. Chebrolu and A. Dhekne, "Esense: Communication through Energy Sensing," in *Proc. of ACM MOBICOM*, 2009.

[3] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "ZiFi: Wireless LAN Discovery via ZigBee Interference Signatures," in *Proc. of ACM MOBICOM*, 2010.

[4] X. Zhang and K. G. Shin, "Gap Sense: Lightweight Coordination of Heterogeneous Wireless Devices," in *Proc. of IEEE INFOCOM*, 2013.

[5] Y. Zhang and Q. Li, "HoWiES: A Holistic Approach to ZigBee Assisted WiFi Energy Savings in Mobile Devices," in *Proc. of IEEE INFOCOM*, 2013.

[6] M.-K. Song and H. Tian, "FreeBee: Cross-technology Communication via Free Side-channel," in *Proc. of ACM MOBICOM*, 2015.

[7] Z. Chi, Y. Li, H. Sun, Y. Yao, and etc., "B2W2: N-Way Concurrent Communication for IoT Devices," in *Proc. of ACM SenSys*, 2016.

[8] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith, "Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices," in *Proc. of ACM SIGCOMM*, 2016.

[9] Z. Yin, W. Jiang, S. M. Kim, and T. He, "C-Morse: Cross-technology Communication with Transparent Morse Coding," in *Proc. of IEEE INFOCOM*, 2017.

[10] W. Jiang, Z. Yin, S. M. Kim, and T. He, "Transparent Cross-technology Communication over Data Traffic," in *Proc. of IEEE INFOCOM*, 2017.

[11] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, "EMF: Embedding Multiple Flows of Information in Existing Traffic for Concurrent Communication among Heterogeneous IoT Devices," in *Proc. of IEEE INFOCOM*, 2017.

[12] X. Guo, X. Zheng, and Y. He, "WiZig: Cross-Technology Energy Communication over a Noisy Channel," in *Proc. of IEEE INFOCOM*, 2017.

[13] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, , and T. He, "BlueBee: a 10,000x Faster Cross-Technology Communication via PHY Emulation," in *Proc. of ACM SenSys*, 2017.

[14] Z. Li and T. He, "WEBee: Physical-Layer Cross-Technology Communication via Emulation," in *Proc. of ACM MOBICOM*, 2017.

[15] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Comm. Surveys and Tutorials*, vol. 13, no. 2, pp. 245–257, 2011.

[16] W. A., J. Huang, and G. Xing, "Practical Bluetooth Traffic Sniffing: Systems and Privacy Implications," in *Proc. of ACM MobiSys*, 2016.

[17] K. G., A. L., and Q. Y., "Jamming and Anti-jamming Techniques in Wireless Networks: A Survey," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 17, no. 4, pp. 197–215, 2014.

[18] M. Vanhoef and F. Piessens, "Advanced Wi-Fi Attacks Using Commodity Hardware," in *Proc. of ACM ACSAC*, 2014.

[19] W. L. W. Group, "Ieee standard part 11: Wireless lan mac and phy specifications," in *IEEE Std 802.11-2012*, 2012.

[20] Y. Chen and Z. Li, "TwinBee: Reliable Physical-Layer Cross-Technology Communication with Symbol-Level Coding," in *Proc. of IEEE INFOCOM*, 2018.

[21] Z. Li and T. He, "LongBee: Enabling Long-Range Cross-Technology Communication," in *Proc. of IEEE INFOCOM*, 2018.

[22] E. Research, "USRP N210 data sheet," in *https://www.ettus.com/*.

[23] A. G., *Wireless communications*. Cambridge university press, 2005.

[24] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau, "Automatic Modulation Recognition Using Wavelet Transform and Neural Networks in Wireless Systems," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 42:1–42:13, 2010.

[25] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for Zigbee performance assurance," in *Proc. of IEEE ICNP*, 2010.

[26] R. D. and P. H. etc., *Pattern classification*. WileyInterscience, 2001.

[27] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. of ACM MOBIHOC*, 2005.

[28] K. P. Vijayakumar, P. Ganeshkumar, and M. Anandaraj, "A novel jamming detection technique for wireless sensor networks," *Ksii Transactions on Internet & Information Systems*, vol. 9, no. 10, pp. 4223–4249, 2015.

[29] D. Liu, J. Raymer, and A. Fox, "Efficient and Timely Jamming Detection in Wireless Sensor Networks," in *Proc. of IEEE MASS*, 2012.

[30] M. S., D. G., V. L., M. W., and J. B. S., "Detection of reactive jamming in dsss-based wireless communications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1593–1603, 2014.

[31] "iperf," https://iperf.fr/.

[32] M. Wilhel, I. Martinovic, J. B. Schmitt, and V. Lenders, "Reactive Jamming in Wireless NetworksHow Realistic is the Threat?" in *Proc. of ACM WiSec*, 2011.

[33] H. Jeongyoon, K. Jung-Jun, and etc., "Dodge-Jam: Anti-Jamming Technique for Low-power and Lossy Wireless Networks," in *Proc. of IEEE SECON*, 2017.

[34] A. Wood, J. Stankovic, and S. Son, "JAM: A Jammed-Area Mapping Service for Sensor Networks," in *Proc. of IEEE RTSS*, 2003.

[35] S. Doro, L. Galluccio, G. Morabito, and etc., "Defeating jamming with the power of silence: A game-theoretic analysis," *IEEE Transactions on Wireless Communications*, vol. 5, no. 14, pp. 2337–2352, 2015.